POLITECNICO
MILANO 1863

SCUOLA DI ARCHITETTURA, URBANISTICA
E INGEGNERIA DELLE COSTRUZIONI

Using Equilibrium Principles to Inform Urban Transport Planning: Insights from the Braess' Paradox

Tesi di Laurea Magistrale in
ARCHITECTURE AND URBAN DESIGN

Author: **Yuzhou Tong**

Student ID: 243240
Supervisor: Prof. Lorenzo Mussone
Co-supervisor: Prof. Roberto Notari
Academic Year: 2023-26

Abstract

Urban transport planning and street design often assume that adding capacity or new connections will reduce congestion. Yet urban mobility is an equilibrium phenomenon shaped by decentralized route choice under congestion, so "improvements" in connectivity can trigger re-routing and unintended network-wide deterioration. This thesis investigates how equilibrium traffic assignment can function as a design-relevant diagnostic: not only to reproduce the Braess' paradox on stylized networks, but to anticipate when street-network interventions may worsen performance and how design can reduce that risk while pursuing spatial-quality goals.

The study proposes a reproducible before–after workflow for equilibrium comparison across demand levels and cost specifications. Two behavioural models are examined: deterministic user equilibrium (DUE), representing fully rational route choice, and stochastic user equilibrium (SUE), representing dispersed perception and heterogeneous decisions. Experiments begin with the canonical diamond network as a benchmark for validation and interpretation, then scale to ladder-type networks that capture repeated urban structures with parallel corridors and potential shortcut links. Demand is scanned systematically and multiple congestion cost functions, including BPR-type forms, are tested. Numerical regime mapping is complemented by symbolic analysis in linear-cost settings to identify equilibrium-equivalence families, in which distinct cost configurations or intervention levers generate similar equilibrium outcomes.

Results show that paradoxical worsening is conditional rather than universal. Added connectivity can be beneficial at low demand yet detrimental over intermediate regimes, with regime boundaries shifting under the shape of the congestion function. Behavioural dispersion matters: compared with DUE, SUE smooths abrupt changes in link usage and can extend adverse effects through gradual shortcut adoption. In ladder networks, shortcut density and placement shape outcomes, indicating that repeated local "improvements" may interact and compound into system-level inefficiency when embedded within broader hierarchies.

For urban design, the central contribution is a translation of these findings into an equilibrium-informed intervention logic. Equilibrium is treated as a constraint on the design space, supporting scenario testing of connectivity changes, identification of demand ranges where new links are risky, and assembly of intervention bundles—geometric, operational, or pricing-like—that can be equivalent in equilibrium effect while enabling improvements in safety, public space, and multimodal performance.

Keywords: Braess' paradox; User equilibrium (DUE/SUE); Urban street-network connectivity; Equilibrium-informed urban design

Abstract in lingua italiana

La pianificazione dei trasporti e il progetto dello spazio stradale assumono spesso che l'aumento di capacità o l'aggiunta di nuove connessioni riducano la congestione. Tuttavia la mobilità urbana è un fenomeno di equilibrio, determinato da scelte di percorso decentralizzate in presenza di congestione: di conseguenza, "miglioramenti" della connettività possono innescare riassegnazioni dei flussi e peggioramenti inattesi a scala di rete. Questa tesi analizza come l'assegnazione del traffico in equilibrio possa operare come strumento diagnostico rilevante per il progetto: non solo per riprodurre il paradosso di Braess su reti stilizzate, ma per prevedere quando interventi sulla rete stradale possano peggiorare le prestazioni e come il progetto possa ridurre tale rischio perseguendo obiettivi di qualità spaziale.

Lo studio propone un flusso di lavoro riproducibile di confronto before–after in equilibrio, variando i livelli di domanda e le specificazioni dei costi. Sono considerati due modelli comportamentali: l'equilibrio deterministico degli utenti (DUE), che rappresenta scelte pienamente razionali, e l'equilibrio stocastico (SUE), che rappresenta percezioni disperse ed eterogeneità decisionale. Gli esperimenti partono dalla rete a diamante come benchmark per validazione e interpretazione, quindi si estendono a reti "a scala" (ladder) che rappresentano strutture urbane ripetute con corridoi paralleli e possibili collegamenti di scorciatoia. La domanda viene scandita in modo sistematico e si testano diverse funzioni di congestione, incluse formulazioni di tipo BPR. La mappatura numerica dei regimi è affiancata da analisi simboliche in contesti a costo lineare per identificare famiglie di equivalenza di equilibrio, nelle quali differenti configurazioni di costo o leve d'intervento generano esiti di equilibrio simili.

I risultati mostrano che il peggioramento paradossale è condizionato e non universale. L'aumento di connettività può essere vantaggioso a basse domande ma dannoso in regimi intermedi, con confini che variano in funzione della forma della funzione di congestione. La dispersione comportamentale è rilevante: rispetto al DUE, il SUE attenua cambiamenti bruschi nell'uso dei collegamenti e può prolungare gli effetti avversi tramite un'adozione graduale delle scorciatoie. Nelle reti ladder, densità e posizione delle scorciatoie influenzano gli esiti, indicando che "miglioramenti" locali ripetuti possono interagire e accumularsi in inefficienze a scala di sistema quando inseriti in gerarchie più ampie.

Per il progetto urbano, il contributo principale è la traduzione di tali evidenze in una logica di intervento informata dall'equilibrio. L'equilibrio viene trattato come vincolo sullo spazio di progetto, supportando la valutazione per scenari delle modifiche di connettività, l'identificazione di intervalli di domanda in cui i nuovi collegamenti sono rischiosi e l'assemblaggio di pacchetti di interventi—geometrici, gestionali o di tipo tariffario—equivalenti negli effetti di equilibrio, pur abilitando miglioramenti in sicurezza,

qualità dello spazio pubblico e prestazioni multimodali.

Parole chiave: Paradosso di Braess; Equilibrio degli utenti (DUE/SUE); Connettività della rete stradale urbana; Progetto urbano informato dall'equilibrio

Contents

Abstract **i**

Abstract in lingua italiana **iii**

Contents **v**

Notation and Units **ix**

1 Introduction **1**

2 Literature Review **15**
2.1 Classical Braess' Paradox . . . 15
2.2 Cost function generalizations . . . 19
2.3 Deterministic versus stochastic equilibrium . . . 22
2.4 From Diamond to Complex Networks . . . 25
2.5 Computational approaches . . . 31
2.6 The Braess' Paradox from the Perspective of Urban Design . . . 34
2.7 Critical assessment and research gaps . . . 39

3 Equilibrium Assignment of Demand **43**
3.1 Mathematical Formulation of Deterministic User Equilibrium (DUE) . . . 43
3.2 Stochastic User Equilibrium and Probabilistic Route Choice . . . 47
3.3 Numerical Solution Methods . . . 50
3.4 Verification on the Braess Network . . . 54

4 Methodology **57**
4.1 Reproduction of the classic Braess' paradox . . . 58
4.2 Cost function families and parameterization . . . 61
4.3 Deterministic versus stochastic equilibrium . . . 66
4.4 Network families, interventions, and path set construction . . . 68
4.5 Numerical computation and symbolic computation . . . 75
4.6 Converting numerical cases into urban design . . . 80
4.7 Summary of the experimental plan . . . 83

5 Results and Discussion **85**
5.1 Classical Braess' paradox . . . 85
5.2 Demand and Cost Sensitivity of the Braess' Paradox . . . 88
5.3 DUE versus SUE: Behavioural Dispersion and the Paradox Window . . . 93
5.4 Ladder networks: structural generalisation of the Braess' paradox . . . 96
5.5 Symbolic structure of Braess' paradox conditions and parameter maps . . . 103
5.6 Results evaluation and discussion . . . 110

6 From Equivalent Cost Families to Spatial Interpretation **115**
6.1 Recap of Equivalent Cost Families . . . 119
6.2 From Mathematical Equivalence to Design Freedom . . . 122
6.3 Network Equivalence and Spatial Non-Uniqueness . . . 126
6.4 Translating Cost Parameters into Street Design Elements . . . 129
6.5 Alternative Spatial Realizations of Equivalent Cost Families . . . 137
6.6 Empirical Correspondence with Urban Design Practice . . . 139
6.7 Implications for Connectivity-Oriented Urban Design . . . 145

7 Conclusions **149**
7.1 Summary of findings . . . 149
7.2 Methodological contributions . . . 151
7.3 Limitations . . . 152
7.4 Future work . . . 153

Bibliography **155**

Acknowledgements **161**

A Tables and Supplementary Results **163**
A.1 Supplementary table for Section 5.1 . . . 166
A.2 Supplementary table for Section 5.2 . . . 166
A.3 Supplementary table for Section 5.3 . . . 174
A.4 Supplementary table for Section 5.4 . . . 175

B Matlab code used in this thesis **199**
B.1 Package overview and verification logic . . . 199
B.2 Script 1: `goal2_main.m` . . . 199
B.2.1 What this script computes . . . 199
B.2.2 How to run and verify . . . 200
B.2.3 User-editable parameters (what to edit) . . . 200
B.2.4 Outputs (what is produced) . . . 201
B.2.5 Configuration used in the present submission . . . 201
B.2.6 Source code listing . . . 201
B.3 Script 2: `braessfigure.m` . . . 219
B.3.1 What this script computes . . . 219
B.3.2 Numerical method (solver and tie-handling) . . . 219

B.3.3 User-editable parameters (what to edit) . . . 219
B.3.4 Outputs (what the script generates) . . . 220
B.3.5 Current configuration in this submission (default values) . . . 220
B.3.6 Source code listing . . . 220
B.4 Script 3: `braess\_batch\_atlas.m` (symbolic atlas generator) . . . 230
B.4.1 What this script computes . . . 230
B.4.2 User-editable parameters (what to edit) . . . 230
B.4.3 How to run (reproducibility) . . . 230
B.4.4 Outputs . . . 230
B.4.5 Current configuration in this submission (default values) . . . 231
B.4.6 Source code listing . . . 231
B.5 Script 4: `plot\_paradox\_maps.m` (atlas visualisation) . . . 235
B.5.1 What this script computes . . . 235
B.5.2 Inputs and dependencies . . . 235
B.5.3 Outputs . . . 235
B.5.4 Source code listing . . . 235
B.6 Script 5: `sf\_equiv\_family\_link23.m` (symbolic equivalent-family generator) . . . 238
B.6.1 What this script computes . . . 238
B.6.2 Inputs (parameter interface) . . . 239
B.6.3 Outputs . . . 239
B.6.4 Source code listing . . . 239
B.7 Script 6: `goal2\_find\_global\_linear\_equiv.m` (numerical equivalence search/verification) . . . 243
B.7.1 What this script computes . . . 243
B.7.2 User-editable parameters (what to edit) . . . 243
B.7.3 Outputs . . . 244
B.7.4 Source code listing . . . 244

Notation and Units

Notation used throughout the thesis (ordered by first appearance)

Symbol	Meaning	Units
$t_{0,a}$	Free-flow travel time on link a	h
c_a	Practical (nominal) capacity of link a	veh/h
α	BPR calibration coefficient controlling congestion penalty magnitude	–
β	BPR exponent controlling curvature of congestion growth	–
θ^1	Theoretical dispersion parameter used in the literature review (conceptual SUE)	–
$\phi(\cdot)$	Delay factor function in Davidson/Akçelik-type formulations	–
a	Link (arc) index, $a \in A$	–
p, q	Path indices (generic feasible paths)	–
$G = (N, A)$	Directed transportation network represented as a graph	–
N	Set of nodes	–
A	Set of directed links	–
W	Set of origin–destination (OD) pairs	–
(r, s)	Origin and destination nodes	–
$\mathcal{P}_{rs}$	Set of feasible paths between r and s	–
$\lvert\mathcal{P}_{rs}\rvert$	Number of feasible paths between r and s	–
δ_{ap}	Path–link incidence indicator (1 if link a belongs to path p)	–
f_p	Path flow on path p	veh/h
x_a	Link flow on link a	veh/h
$\mathbf{f}$	Vector of path flows	veh/h
$\mathbf{x}$	Vector of link flows	veh/h
$\hat{\mathbf{x}}$	Auxiliary link-flow vector obtained from deterministic or probabilistic loading	veh/h
Q_{rs}	Travel demand between OD pair (r, s)	veh/h
Q	Total demand used in scanning experiments (single-OD benchmark)	veh/h
$t_a(x_a)$	Travel time (cost) function on link a	h
$C_p(\mathbf{x})$	Travel time (systematic cost) on path p	h

Symbol	Meaning	Units
$\min_{q \in \mathcal{P}_{rs}} C_q(\mathbf{x})$	Minimum path cost between (r, s) under flow $\mathbf{x}$	h
$\arg\min_{q \in \mathcal{P}_{rs}} C_q(\mathbf{x})$	Set of minimum-cost paths between (r, s) under flow $\mathbf{x}$	–
u	Dummy integration variable in the Beckmann integral formulation	–
ATT	Average travel time of the network	h
ΔATT	Difference in average travel time (After–Before)	h
$ATT^{(B)}$	Average travel time in the before network	h
$ATT^{(A)}$	Average travel time in the after network	h
$TSTT$	Total system travel time ($\sum_p f_p C_p$)	veh·h
T_p	Travel time of path p (reported results)	h
Φ_p	Path flow of path p in result tables	veh/h
U_p	Random perceived utility of path p	–
ε_p	Random perception error term	–
P_p	Probability of choosing path p	–
ΔC	Systematic cost difference between two alternatives (used in Logit odds ratio)	h
θ	Dispersion parameter used in numerical experiments and simulations	–
b_a	Intercept parameter in linear cost $t_a(x_a) = b_a + m_a x_a$	h
m_a	Slope parameter in linear cost $t_a(x_a) = b_a + m_a x_a$	h/(veh/h)
P	Fixed admissible path set used by the solver (union over OD pairs)	–
$P_{\min}$	Minimum-cost path set (used in deterministic all-or-nothing loading)	–
$\lvert P_{\min} \rvert$	Number of minimum-cost paths	–
$\mathbf{\Delta}$	Path–link incidence matrix mapping $\mathbf{x} = \mathbf{\Delta f}$ (entries consistent with δ_{ap})	–
$\mathbf{f}^{(k)}$	Path-flow vector at iteration k	veh/h
$\mathbf{x}^{(k)}$	Link-flow vector at iteration k	veh/h
$y_p^{(k)}$	Auxiliary path flow on path p at iteration k (loading output)	veh/h
$\mathbf{y}^{(k)}$	Auxiliary path-flow vector at iteration k	veh/h
$P_p^{(k)}$	Path choice probability for path p at iteration k (SUE loading)	–
$\mathbf{T}(\mathbf{x})$	Fixed-point flow update mapping operator	–
α_k	Step size in Method of Successive Averages (MSA) iteration	–
k	Iteration index	–
$K_{\max}$	Maximum iteration budget per assignment run	–
$\lVert \cdot \rVert$	Vector norm (default L_2 unless otherwise stated)	–
$\lVert \cdot \rVert_1$	L_1 norm (used in convergence / restart criteria)	–
ε	Convergence tolerance	–
$\varepsilon_{\text{zero}}$	Numerical zero threshold for treating $\lvert \Delta ATT \rvert$ as 0	–

Symbol	Meaning	Units
$\varepsilon_{\text{paradox}}$	Paradox classification threshold for ΔATT	–
it	Iteration counter used in restart monitoring	–
rel	Relative change metric used to detect stagnation/restart triggering	–
$\text{rel}_{\min}$	Best-so-far value of rel within a restart window	–
K_{restart}	Restart window size (max consecutive iterations without meaningful improvement)	–
τ_{restart}	Restart improvement threshold defining "meaningful improvement" in rel	–
$R_{\max}$	Maximum number of restarts allowed within a single assignment run	–
$P_{\max}$	Maximum number of enumerated simple paths (implementation safety cap)	–
u_i	Link label used in Braess-type benchmark networks ($i = 1, \ldots, 5$)	–
x_{u_i}	Flow on link u_i (used in result tables/figures)	veh/h
u_5	Added/shortcut link in the Braess diamond network	–
n	Ladder length parameter (number of segments / bypass positions)	–
`ConfigID`	Configuration index for a specific after-network bypass subset	–
`NumBypassPos`	Number of activated bypass positions in a configuration	–
`BypassPos`	Set/list of activated bypass locations in a configuration	–
$t_i(x)$	Affine link cost function $t_i(x) = a_i x + b_i$	h
a_i	Slope (congestion sensitivity) parameter of affine cost	h/(veh/h)
b_i	Intercept (baseline delay) parameter of affine cost	h
$t_5(x)$	Cost of added link 5, $t_5(x) = a_5 x + b_5$	h
a_5	Slope parameter of link 5	h/(veh/h)
b_5	Intercept parameter of link 5	h
Δ	Performance shift defined as $T_{\text{after}} - T_{\text{before}}$	h
$\Delta(a_5, b_5)$	Performance difference function induced by (a_5, b_5)	h
D	Demand symbol used in symbolic/batch appendices (same role as Q)	veh/h
a_U, b_U	Grouped linear parameters on upper links (e.g., $1 \to 2$ and $3 \to 4$)	a_U: h/(veh/h), b_U: h
a_M, b_M	Grouped linear parameters on middle links (e.g., $1 \to 3$ and $2 \to 4$)	a_M: h/(veh/h), b_M: h
dt	Target performance shift used in symbolic atlases (requested Δ)	h
Δt_{target}	Target shift notation used in appendix plots/scripts	h

1 | Introduction

Transportation networks are among the most enduring and capital-intensive components of urban systems. Roads, highways, railways, metro and tram lines do more than accommodate daily mobility: they structure accessibility, reshape land values and economic interactions, and influence urban development over decades. Because of long service lives and substantial sunk costs, transport infrastructures outlast political terms and shifts in planning paradigms, continuing to steer travel behaviour and spatial growth. As a result, network expansion, reconstruction, or removal is typically framed as a strategic investment expected to relieve congestion, improve accessibility, and enhance productivity over long horizons. Yet these expectations implicitly assume that network-scale interventions translate into better equilibrium outcomes—an assumption that warrants systematic theoretical scrutiny.

In engineering practice and transport planning, a core premise is that increasing capacity or connectivity improves overall performance. Classical planning frameworks forecast how flows redistribute after projects and treat reductions in congestion and travel time as primary success metrics [1, 2]. Within this supply-oriented view, congestion is interpreted mainly as insufficient capacity, so adding links, widening roads, or improving connectivity is presumed to yield a better system outcome. However, this logic tacitly presumes that supply-side modifications produce a uniquely improved equilibrium, overlooking two important possibilities: distinct infrastructures may generate equivalent equilibrium outcomes, and some expansions may even worsen equilibrium performance.

Throughout much of the twentieth century, this "additive logic" supported policies such as bypass construction, ring roads, and shortcut connections intended to divert flows and reduce travel distance. The underlying expectation is straightforward: more route options or shorter paths should lower travel times and increase efficiency. This belief is embedded in cost–benefit appraisal, where time savings typically dominate estimated benefits [1].

Real-world experience, however, often conflicts with this intuition. New expressways can induce additional demand that erodes capacity gains; bypasses may rapidly congest; and altered route choices can shift congestion onto corridors that were previously uncongested. Such outcomes suggest that the expansion–performance relationship is neither linear nor monotonic, complicating any simple mapping from "physical modification" to "system benefit." More broadly, they point to the possibility that multiple network configurations can produce similar equilibrium flow patterns.

A central reason is behavioural response. Transportation networks are not passive conduits: they form socio-technical systems shaped by infrastructure supply, congestion dy-

namics, and decentralized decision-making. Travellers adapt to interventions by changing routes, departure times, destinations, and—over longer horizons—residential and employment locations. These adjustments interact nonlinearly with congestion, so network interventions must be assessed in terms of how they reshape equilibrium rather than through link-level reasoning alone.

This systemic perspective is formalized by equilibrium traffic assignment theory. User equilibrium, grounded in Wardrop's first principle [3], requires that all routes actually used between a given OD pair have equal and minimal travel cost, while any unused route has a cost no lower; Wardrop's second principle defines the system optimum, in which flows are arranged to minimize total system travel cost. Equilibrium flows therefore arise from the interaction of many self-interested decisions rather than centralized optimization, forming the basis of most operational planning tools [4, 5]. At the same time, equilibrium theory also admits that different network structures or cost specifications may lead to the same equilibrium pattern, opening the possibility of equivalence among infrastructure interventions—and, in some cases, counterintuitive outcomes. The canonical diamond network and its extensions used in this thesis (generalized diamond and ladder networks) are illustrated in Figures 4.1, 4.4, 4.5, and 4.6.

Unintended consequences in transportation networks must be situated within the longer evolution of urban and transport planning theory. Transport systems are not designed in isolation: they are embedded in urban environments where infrastructure, land use, economic activity, and social behaviour co-evolve. As a result, how networks are conceived, evaluated, and implemented reflects prevailing paradigms and policy priorities, as well as implicit assumptions about how travellers respond to change. Within this broader context, the Braess paradox is not only a warning against naive expansion. It also provides a planning-relevant framework for reasoning about how network-wide modifications can, under certain conditions, reproduce comparable (or even equivalent) equilibrium flow patterns despite substantial differences in local street design.

For much of the twentieth century, transportation planning was dominated by a technocratic, engineering-oriented paradigm. Classical frameworks conceptualized mobility problems primarily as capacity shortages to be addressed through infrastructure provision and traffic engineering [1, 6]. In this tradition, planners forecast travel demand, assign it to networks using equilibrium or incremental assignment techniques, and identify expansions required to reach target levels of service. Congestion was treated as a temporary supply–demand imbalance, remediable through added capacity, improved connectivity, or better operational control. Evaluation procedures emphasized travel time savings, vehicle operating cost reductions, and increased throughput, thereby institutionalizing the idea that "improvement" is largely synonymous with increasing supply. Although equilibrium assignment was widely adopted for operational realism [4, 5], the prevailing narrative in practice remained broadly monotonic: more capacity and more connectivity were expected to translate into better network performance at equilibrium.

Over time, however, two lines of critique emerged that complicate this supply-oriented intuition. First, urban planning and urban design scholarship challenged the idea that

streets should be evaluated primarily as conduits for vehicular flow. Second, empirical transport research documented systematic gaps between capacity expansion and observed congestion outcomes. Together, these critiques motivated a shift from local, link-level reasoning toward system-wide accounts of behavioural response and network interaction.

A formative planning critique of automobile-oriented policy was articulated by Jane Jacobs [7], who argued that excessive road building and traffic segregation undermine urban vitality, walkability, and social interaction. While Jacobs' arguments were qualitative rather than model-based, they established a lasting intellectual foundation for understanding streets as civic spaces as well as movement channels. This reframing is crucial for contemporary interventions such as arterial "road diets" or lane narrowing, whose objectives are not to maximize vehicular throughput but to re-balance street functions in favour of safety, accessibility, and public life. In this perspective, congestion management is not simply a technical optimization problem; it is inseparable from questions of urban form, spatial quality, and the distribution of mobility benefits and burdens.

Urban morphology and network-based approaches further strengthened this shift by demonstrating that the structure of street networks shapes movement patterns even before congestion dynamics are considered. Space syntax theory showed that street connectivity and hierarchy influence pedestrian and vehicular flows through spatial configuration [8]. Research on street pattern typologies emphasized that cities evolve as structured networks balancing local permeability and global hierarchy [9]. More recently, network science approaches have treated urban street systems as spatial graphs whose topological properties condition efficiency, resilience, and accessibility [10–12]. These perspectives converge on a common implication: connectivity is not an unqualified good. What matters is how connectivity is organized into hierarchies and how local changes perturb global structure. This implication aligns closely with the central idea explored in this thesis: if topology and hierarchy shape how flows distribute, then coordinated, network-wide changes may preserve—or intentionally reproduce—equilibrium flow patterns even when local street conditions are substantially modified.

In parallel with these conceptual shifts, empirical transportation research began to document systematic discrepancies between capacity expansion and congestion relief. The phenomenon of induced travel demand, synthesized by Goodwin [13] and by Cervero and Hansen [14], showed that new capacity generates additional vehicle travel that erodes predicted time savings. Rather than yielding persistent congestion relief, capacity additions can trigger behavioural adjustments such as route changes, mode shifts, and trip generation, which partially or fully offset the expected benefits. This evidence undermined the foundational assumption of classical planning that supply expansion produces durable performance improvements, and it reinforced a broader lesson: performance impacts cannot be inferred from local changes alone because users adapt to network conditions.

Duranton and Turner [15] provided rigorous quantitative confirmation of this mechanism through the "fundamental law of road congestion", demonstrating that vehicle kilometres travelled scale proportionally with highway lane kilometres in U.S. metropolitan areas. Their analysis implied that highway expansion does not reduce congestion in the long run; instead, it accommodates and induces additional travel through multiple adjustment channels. For the present thesis, this line of evidence strengthens the motivation to

examine equilibrium mechanisms even in the short run: if expansion can be neutralized or reversed by adaptation, then the relevant analytical question becomes how interventions reshape equilibrium incentives, and whether alternative network configurations can yield comparable equilibrium outcomes under different local street designs.

Transport–land use interaction research extends this point by showing that infrastructure influences spatial development, residential location, and economic activity [16, 17]. From this perspective, road expansion is not merely a response to congestion; it also acts as a driver of urban sprawl, automobile dependence, and longer trip distances. Transportation planning is therefore inseparable from broader urban development trajectories, and congestion cannot be treated as a purely technical problem independent of land-use policy. This implication is especially relevant for design-oriented interventions such as systematic arterial lane narrowing paired with improved local accessibility: their effects arise from the joint response of network supply, urban form, and behavioural adaptation rather than from traffic operations alone.

A complementary empirical strand documented cases in which reducing or reallocating road capacity did not produce the severe congestion predicted by traditional intuition. Cairns et al. [18] reviewed international examples of "disappearing traffic," where road closures, pedestrianization, or capacity reductions resulted in stable or even improved network conditions. These cases suggest that travellers adapt through destination changes, mode shifts, trip suppression, and reorganization of route choice, leading to new equilibria that do not necessarily exhibit increased congestion. For contemporary urban design agendas, this evidence is particularly salient: it indicates that capacity reduction does not mechanically imply diversion onto local streets, and it motivates a more precise analytical question—under what network-wide conditions can reallocation interventions preserve acceptable equilibrium flows while improving local street environments?

Contemporary street design practice has institutionalized these insights through systematic road space reallocation policies. Guidelines increasingly advocate transforming car-dominated corridors into multimodal public spaces accommodating pedestrians, cyclists, and public transport [19]. Empirical accounts of implemented projects indicate that such interventions can enhance livability without causing unmanageable disruption [20]. These practices embody a shift away from capacity maximization toward balanced, human-centred mobility systems. Yet their evaluation often remains contested because standard engineering intuition predicts diversion and delay; what is needed is an equilibrium-based explanation of how coordinated network change can offset local capacity reductions and, in some cases, sustain broadly comparable equilibrium flow patterns.

Collectively, these developments have reshaped urban transportation planning in recent decades. Rather than treating congestion as a simple supply–demand imbalance, contemporary planning increasingly recognizes transportation networks as adaptive socio-technical systems. Infrastructure interventions trigger behavioural, spatial, and modal responses that can amplify, dampen, or even reverse intended effects. As a result, evaluating projects solely through static forecasts and link-level reasoning is no longer sufficient. What is increasingly required is a system-wide account of how interventions reconfigure incentives across the network and whether distinct configurations can be analytically comparable at equilibrium despite substantial differences in local street form.

Nevertheless, despite extensive empirical documentation of induced demand, disappearing traffic, and street reallocation outcomes, planning research has often lacked a unifying analytical language for explaining why such counterintuitive effects occur under equilibrium conditions. Many studies rely on case evidence, before–after observation, or econometric inference. By contrast, equilibrium traffic assignment theory provides mathematically explicit tools for analysing how decentralized route choice interacts with congestion and network structure [4, 5]. The challenge is therefore to bridge these domains: to relate physical interventions and design strategies to the equilibrium incentives they generate at the network scale, and to compare interventions not only by form but by whether they produce similar equilibrium outcomes under modified cost conditions.

This is precisely where the Braess paradox becomes central. The paradox provides a minimal theoretical demonstration that adding capacity or connectivity can worsen network performance under equilibrium routing [21, 22]. Although originally formulated in highly stylized networks, its logic resonates with planning observations of expansion that fails to relieve congestion and with design concerns about cut-through traffic and hierarchy disruption. In this sense, the Braess paradox offers a micro-foundation for empirical findings: it shows that, under rational route choice, performance degradation can occur even without long-run land-use adaptation. Equally important for this thesis, it motivates an inverse analytical perspective: by treating performance as an equilibrium outcome, one can examine how network-wide modifications of traffic conditions (for example, systematic changes in link cost parameters) may reproduce comparable or equivalent equilibrium flow patterns, thereby enabling spatial interventions that reshape local streets without necessarily inducing harmful diversion.

Urban network morphology research suggests that real street systems contain repeated structural motifs analogous to Braess-type subgraphs [9, 11]. This implies that paradox-generating mechanisms are not confined to toy examples but may be embedded within larger urban fabrics. If such motifs recur across arterial–local interfaces, then interventions such as coordinated arterial lane narrowing must be evaluated in terms of how they perturb (or preserve) equilibrium incentives across these embedded structures, and whether compensating network-wide modifications can maintain acceptable routing patterns.

At the same time, contemporary planning practice continues to rely heavily on equilibrium assignment principles [4, 5]. Deterministic and stochastic user equilibrium models remain workhorses of metropolitan planning, underpinning long-range investment decisions. If these models admit paradoxical outcomes under certain interventions, evaluation procedures must explicitly account for such possibilities rather than assuming monotonic improvement from added capacity. Conversely, if equilibrium models admit families of network conditions that reproduce similar equilibrium outcomes, they provide a rigorous basis for designing intervention bundles that change local street form while controlling system-wide diversion risk.

From a policy perspective, the implications are substantial. Urban transport policy increasingly aims to reduce automobile dependence, promote sustainable modes, and reallocate street space toward public and active transport [23–25]. Achieving these goals often requires deliberate reduction or reconfiguration of road capacity. Equilibrium reasoning clarifies why such strategies can succeed or fail: local geometry alone does not determine

outcomes; what matters is how system-wide incentives reshape equilibrium routing. The Braess paradox therefore reframes transportation planning from a purely capacity-driven enterprise into a strategic design problem involving topology, behavioural response, and congestion interaction. Within this reframing, the central analytical task pursued in this thesis is to identify when network-wide modifications can yield comparable or equivalent equilibrium flows, and what this implies for design interventions that seek to reclaim street space while avoiding unintended diversion.

Although the Braess paradox is often introduced through a stylized network, its deeper significance lies in the methodological challenges it poses for transportation modeling, equilibrium analysis, and infrastructure evaluation. The paradox is not merely a curiosity tied to a particular configuration; it exposes structural limits in how transportation systems are conceptualized and assessed in both research and planning practice. By showing that a local "improvement" can generate global deterioration under equilibrium routing, it compels a reconsideration of the assumptions embedded in traffic assignment, congestion representation, and network design. For this thesis, it also motivates a complementary question: whether network-wide modifications to traffic conditions can be structured to reproduce comparable or even equivalent equilibrium flow patterns, thereby enabling spatial interventions that alter local street form without mechanically inducing harmful diversion.

At its core, the paradox arises from the interaction of three standard modeling ingredients: network topology, congestion-dependent link costs, and decentralized route choice. Individually, each ingredient is conventional; jointly, they can yield outcomes that contradict intuitive engineering logic. This contradiction is methodological rather than anecdotal: it reveals that seemingly benign modeling choices may interact in nontrivial ways once embedded in an equilibrium framework. Accordingly, the Braess paradox can be read as a stress test for any model intended to support planning decisions: it requires the model to anticipate and explain counterintuitive equilibrium outcomes. More broadly, if equilibrium outcomes are governed by network-wide incentive structures rather than isolated link changes, then methodological attention must shift from local edits to the identification of intervention families whose equilibrium implications can be systematically characterized.

A first methodological dimension concerns how equilibrium is defined. Classical transportation planning relies heavily on deterministic user equilibrium (DUE) (Section 3.1), in which travellers are assumed to have perfect information and choose minimum-cost routes [3]. Under DUE, equilibrium flows satisfy Wardrop's first principle: no traveller can reduce travel cost by unilaterally switching routes. This equilibrium concept is widely used because it is analytically clear and computationally tractable [4, 5]. The classical Braess paradox is derived precisely under DUE [21, 22], showing that paradoxical outcomes are not exceptions to equilibrium modeling but possibilities internal to its most standard formulation. Methodologically, this implies that infrastructure interventions cannot be evaluated through monotonic "more is better" assumptions; they must be assessed by how they reshape equilibrium incentives across the network.

At the same time, the behavioural realism of perfect information and full rationality

has long been questioned. Travellers exhibit perception errors, incomplete information, habit, and bounded rationality. Stochastic user equilibrium (SUE) models address these issues by embedding random utility theory and representing route choice probabilistically rather than deterministically [26, 27]. A key methodological question is therefore whether paradoxical effects persist when equilibrium is defined probabilistically and whether equivalence in equilibrium outcomes remains robust to behavioural noise. Prior work indicates that Braess-type effects can indeed occur under SUE, although their magnitude and sensitivity to parameters may differ [28, 29]. This finding is important for planning applications: it implies that paradoxical degradation is not an artifact of idealized rationality, and that both DUE and SUE frameworks require explicit attention to system-level incentive shifts.

A second methodological dimension concerns the representation of congestion through link cost functions. Analytical treatments of the Braess paradox often adopt linear or affine travel time functions for tractability, whereas applied planning models typically employ nonlinear cost formulations, most notably BPR-type functions [5]. Such functions introduce strong convexity and capacity effects that influence equilibrium sensitivity and stability. Because this thesis links equilibrium outcomes to design-relevant modifications of traffic conditions, the functional form and parameterization of link costs become methodological choices: they condition when a local capacity reduction can be offset by network-wide adjustments and when it triggers large equilibrium redistributions.

Research has shown that the existence and severity of the Braess paradox depend on the shape of congestion functions [30, 31]. Highly convex costs can amplify the equilibrium congestion shift induced by new connectivity, while weaker convexity may shrink paradoxical regions without eliminating them. In practical planning contexts, cost functions are calibrated from speed–flow relationships and capacity estimates, but calibration uncertainty and link heterogeneity imply that equilibrium outcomes may vary significantly with parameter choices. The paradox thus highlights a broader methodological issue: relying on single point estimates can obscure regime changes driven by equilibrium feedback. For intervention design, this supports the need for systematic exploration of parameter space to identify regions where outcomes are robust and, in particular, where intervention equivalences persist across plausible cost specifications.

A third methodological dimension concerns network structure. The canonical Braess network is intentionally minimal, which has led some critics to treat the paradox as a peculiarity of toy examples. However, subsequent research indicates that Braess-like substructures can arise within larger, more realistic networks [32, 33]. Urban street systems commonly contain multiple overlapping routes between origin–destination pairs, cross-connections between parallel corridors, and hierarchical link functions. These features create precisely the conditions under which equilibrium flow redistribution can become counterintuitive. Methodologically, the paradox should therefore be treated as a local motif whose effects depend on how it is embedded in a broader topology, and equivalence between interventions should likewise be evaluated with respect to network embedding rather than isolated subgraphs.

Network science provides a complementary lens by showing how local connectivity changes can reconfigure global load and centrality patterns in complex networks [34, 35]. From

this perspective, the Braess paradox is a specific instance of a more general phenomenon: in systems governed by decentralized decisions, adding a link can shift load toward critical components and degrade overall performance. Analogous antagonistic effects have been documented beyond transportation, including electrical grids and communication networks [36, 37]. For planning and design, the methodological implication is that network performance is an emergent property of the whole topology. Local interventions such as lane narrowing or new connectors cannot be interpreted purely locally if they perturb global load distribution; the analysis must identify how network-wide incentives and capacities jointly determine equilibrium allocations.

A fourth methodological dimension concerns temporal representation. Classical formulations of the Braess paradox assume static equilibrium: demand and conditions are fixed over the analysis horizon. Real networks, however, evolve dynamically, with queue formation, spillback, departure-time choice, and adaptive routing. Dynamic traffic assignment models attempt to capture these processes [38, 39]. This raises two methodological questions: whether Braess-type effects persist under dynamic modeling, and whether equivalence identified under static equilibrium remains meaningful when adjustment paths generate transient diversion or instability.

Recent work suggests that dynamic Braess-type phenomena can occur, including time-dependent amplification and instability [40, 41]. This strengthens the interpretation of unintended consequences as intrinsic to decentralized network dynamics rather than artifacts of static simplification. For planning practice, it also implies that evaluation should consider not only equilibrium end states but whether the transition dynamics impose unacceptable temporary burdens. This concern is especially relevant for staged interventions implemented corridor by corridor, where short-run diversion dynamics may differ from long-run average conditions.

A fifth methodological dimension concerns behavioural richness and generalized cost. Traditional equilibrium models often emphasize travel time, but travellers may also respond to reliability, comfort, tolls, environmental factors, and habit. Extended equilibrium formulations incorporating generalized cost, bounded rationality, and learning have been proposed [42, 43]. Evidence suggests that behavioural enrichment can shift paradox regimes but does not eliminate the underlying possibility of performance degradation under added connectivity. For this thesis, the methodological implication is that equivalence-based reasoning should not rely on an excessively fragile behavioural idealization; it must be formulated to remain informative under reasonable departures from perfect rationality.

The paradox also raises questions about evaluation metrics. Planning appraisal commonly focuses on aggregate measures such as total travel time or average delay. The Braess paradox shows that such measures can worsen even when each traveller follows individually cost-minimizing behaviour. This implies that link-level indicators (for example, faster speeds on a project corridor) can be misleading if considered in isolation, because equilibrium redistribution may overwhelm local gains. For design-oriented strategies, the relevant concern is not simply whether a corridor slows, but whether the network-wide equilibrium reallocates flows onto sensitive local streets and whether such redistribution can be controlled through coordinated, network-wide adjustments.

Finally, the Braess paradox reframes network design as a strategic problem under equilibrium constraints. If adding a link can worsen performance, then design cannot be reduced to monotonically increasing capacity where congestion appears highest. Instead, network design requires anticipating equilibrium response and, in some cases, removing or downgrading links to improve global performance. This insight motivates research on network design with user equilibrium constraints [31, 44]. In the perspective adopted here, a further implication follows: network design can also be posed as the search for intervention families that alter local street form while preserving desired equilibrium flow patterns at the network scale. This framing links equilibrium analysis directly to planning and urban design objectives, including systematic arterial lane narrowing strategies that aim to reclaim street space without inducing uncontrolled diversion across the wider network.

Building on the planning and urban design background above, the methodological questions raised by the Braess' paradox are not confined to abstract modeling choices; they correspond directly to concrete spatial decisions concerning street layout, permeability, hierarchy, and the allocation of public space. In urban design practice, interventions that "add a link" or "increase connectivity" are materialized as new connectors, shortcut streets, underpasses, flyovers, mid-block passages, or the opening of previously restricted links. Conversely, "removing a link" or "reducing capacity" corresponds to pedestrianization, road diets, modal filters, one-way conversions, or selective closures. The Braess' paradox becomes design-relevant precisely because it formalizes a mechanism by which such localized spatial changes can yield network-wide outcomes that are counterintuitive under purely local reasoning, and because it suggests that spatial reclamation strategies must be evaluated as system-wide modifications of equilibrium incentives rather than as corridor-level geometry alone.

A first bridge between Braess theory and urban design lies in the dual identity of streets as both movement networks and public spaces. Urban design critiques of car-oriented planning emphasize that streets should not be evaluated solely by vehicular throughput but by their contribution to urban life and spatial quality [7, 20]. Yet even when restricting attention to mobility performance, the discussion above showed that real cities often exhibit adaptive responses such as "disappearing traffic" [18] and that capacity expansion can be undermined by behavioral feedbacks such as induced demand [13–15]. The Braess' paradox provides a short-run, equilibrium-based micro-foundation for these observations: it demonstrates that individual users, by rationally exploiting an apparently beneficial shortcut, may collectively shift flows in a way that increases system cost [21, 22]. In urban design terms, the new connector can function as a magnet that reorganizes route choice, redistributes congestion, and alters the spatial hierarchy of movement; conversely, coordinated reductions in capacity can sometimes function as a filter that stabilizes hierarchy and discourages inefficient shortcut-chasing without necessarily triggering diversion onto local streets.

A second bridge concerns spatial configuration and hierarchy. Urban morphology and network-based urban analysis have shown that street systems operate as structured hierarchies rather than uniform grids; movement concentrates along a limited set of highly integrated corridors, while local streets support finer-grained access [8, 9, 11]. From this

viewpoint, the addition of a seemingly minor cross-connection can reconfigure global centrality patterns and redirect substantial through-movement into spaces not designed to accommodate it. In equilibrium terms, the new link changes the set of feasible routes and the perceived generalized costs, potentially concentrating flow onto corridors that become over-saturated. In design terms, this is the familiar problem of "rat-running" or cut-through traffic: local permeability introduced for accessibility may inadvertently attract regional through-flows, degrading both mobility and urban quality. The Braess' paradox formalizes a precise condition under which such an outcome is not accidental but structurally induced by equilibrium incentives, and it motivates an inverse design reading in which interventions such as systematic arterial lane narrowing are assessed by whether they preserve the intended hierarchy of flows rather than simply displacing them.

A third bridge is provided by the notion of topological side effects in complex networks. Urban networks can be conceptualized as spatial graphs in which changes in connectivity alter load distribution in non-linear ways [10, 12]. Network science further suggests that adding links may increase vulnerability or overload critical components by shifting flows toward high-betweenness paths [34]. In transportation terms, this aligns with the Braess mechanism: a shortcut that appears individually attractive can cause the equilibrium to collapse onto a smaller subset of links, thereby increasing congestion externalities and total travel time. In urban design terms, this means that connectivity interventions cannot be judged solely by geometric distance reduction or local accessibility improvements; they must be evaluated by how they reshape the global flow structure and whether they strengthen or destabilize the intended movement hierarchy. The same logic applies to capacity-reduction strategies: narrowing an arterial is not merely a local loss of throughput, but a topological perturbation that can either redirect flow uncontrollably or, if aligned with network-wide incentive shifts, dampen inefficient routing patterns and protect local streets from becoming default bypasses.

Importantly, this translation allows the Braess' paradox to be interpreted as a design question about where connectivity should be placed, rather than whether connectivity is desirable in principle. Urban design practice frequently aims for permeability to support walkability and local access, but also relies on selective filtering or hierarchical control to prevent excessive through-traffic on local streets. Contemporary street design guidance explicitly endorses interventions such as road diets and multimodal reallocation [19], while case evidence suggests that cities can adapt to reduced road capacity without unacceptable congestion [18]. The Braess' paradox complements this practice by demonstrating that under equilibrium routing, reducing certain connections or capacities can in some cases improve system performance. Thus, what appears in planning as an empirical phenomenon (traffic evaporation) and in design as a place-making strategy (street calming) acquires a rigorous equilibrium-based explanation, and it becomes plausible to treat arterial lane narrowing not as a purely local "road diet" but as part of a network strategy that can preserve equilibrium flow patterns while reclaiming public space.

The methodological implication is that equilibrium analysis can be used as a design diagnostic to identify where a new connector is likely to trigger paradoxical rerouting, and conversely, where selective removal or filtering may improve both network performance and urban quality. While classical planning evaluations may focus on link-level bene-

fits, the Braess framework insists on system-level assessment: the relevant question is not whether a connector shortens a particular route, but whether it changes equilibrium incentives such that the network shifts toward a less efficient flow pattern. This resonates with the broader sustainable mobility policy shift away from throughput maximization toward accessibility, livability, and demand management [23–25]. Framed in this way, interventions such as systematic arterial lane narrowing can be evaluated by whether they modify incentives at the network scale—raising generalized costs of opportunistic shortcut routes while preserving the functional role of arterials—so that local streets are reclaimed without becoming involuntary spillover corridors.

Finally, relating Braess to urban design clarifies why the paradox should be treated as a methodological problem rather than a rare anomaly. Urban street networks routinely include the ingredients of Braess-type conditions: multiple near-substitutable routes, cross-links between parallel corridors, and heterogeneous link functions (arterials versus locals). Moreover, urban design interventions frequently target precisely those elements (opening or closing links, changing lane allocation, introducing filters) that can move the system across different equilibrium regimes. Therefore, the methodological agenda implied by the Braess' paradox—sensitivity to topology, cost representation, and behavioral assumptions—becomes directly relevant to evaluating urban design strategies. In later chapters, the thesis operationalizes this agenda by systematically scanning network configurations and cost families under deterministic and stochastic equilibria, thereby producing planning-relevant "paradox atlases" that can inform the design of connectivity and street hierarchy interventions, including the identification of network-wide modifications capable of yielding equivalent equilibrium flows despite substantial changes in local street allocation.

In summary, the Braess' paradox is best understood not merely as a counterintuitive result in a toy network, but as a methodological challenge that cuts across equilibrium definition, congestion representation, network topology, behavioral modeling, dynamic processes, and evaluation metrics. It reveals that the standard modeling toolkit of transportation planning contains within it the seeds of paradoxical outcomes. This recognition does not invalidate equilibrium modeling; rather, it underscores the necessity of using equilibrium tools critically, systematically, and with explicit awareness of their systemic implications, especially when the design objective is not new capacity but spatial redistribution of street space under controlled network-wide traffic outcomes.

By exposing the limits of local reasoning in network evaluation, the Braess' paradox compels transportation researchers, planners, and urban designers to adopt holistic, system-oriented analytical frameworks. It calls for methodological approaches capable of exploring parameter sensitivity, topological variation, and behavioral uncertainty rather than relying on single deterministic forecasts. The computational and symbolic frameworks developed in this thesis respond directly to this methodological need, providing tools to systematically map where and why paradoxical outcomes arise. Viewed in this light, the Braess' paradox becomes more than a theoretical curiosity: it becomes a diagnostic instrument for stress-testing transportation models, a conceptual bridge linking network science and planning practice, and a foundation for more robust infrastructure evaluation and urban design methodologies, including the analytical justification of interventions

that reclaim local streets while avoiding diversion through the identification of equivalent or stable equilibrium flow regimes.

The structure of this thesis is designed to reflect the progressive development of the research problem, moving from conceptual foundations and planning context to methodological formulation, systematic analysis, and interpretation of results. Each chapter builds upon the previous one, forming a coherent narrative that links equilibrium-based network theory with contemporary challenges in urban and transportation planning.

Chapter 2 provides a comprehensive literature review. It situates the Braess' paradox within the broader theoretical landscape of traffic assignment and equilibrium modeling, reviews extensions of the paradox across network topologies and cost function families, and summarizes developments in deterministic and stochastic equilibrium theory. The chapter also consolidates empirical findings from transportation planning and urban design research, including induced demand, disappearing traffic, and street network morphology. By integrating these strands, Chapter 2 establishes the theoretical and empirical foundations upon which the subsequent methodological developments are built.

Chapter 3 introduces the equilibrium assignment models that form the analytical backbone of the thesis. It presents the mathematical formulation of deterministic and stochastic user equilibrium, discusses their behavioral interpretations, and describes the numerical algorithms employed to solve them. Particular attention is given to convergence properties, computational challenges, and verification procedures. This chapter ensures that the modeling framework used in later analyses is transparent, reproducible, and theoretically grounded.

Chapter 4 presents the methodological framework developed for the systematic investigation of Braess-type paradoxes. It formalizes the network representations, demand settings, and cost function families adopted in the research. The chapter introduces the unified simulation environment that integrates network construction, equilibrium assignment, parameter scanning, and paradox detection. This framework enables systematic exploration of how equilibrium outcomes change under variations in topology, congestion representation, and behavioral assumptions.

Chapter 5 contains the main analytical and numerical results. It reports the outcomes of extensive equilibrium simulations, identifies regions of paradoxical behavior across parameter spaces, and examines how paradox severity depends on demand levels, cost functions, and equilibrium type. The chapter also presents the symbolic derivation of equilibrium conditions for linear-cost networks and introduces the inverse formulation that characterizes equivalence classes of paradox-generating configurations. Through these results, Chapter 5 transforms the Braess' paradox from an isolated example into a structured and generalizable phenomenon.

Chapter 6 interprets the analytical findings from a planning and urban design perspective. It translates abstract equilibrium results into spatial and policy-relevant insights, discussing how connectivity interventions, street hierarchy design, and capacity reallocation may influence network-wide performance. The chapter outlines implications for

transportation planning evaluation, sustainable mobility strategies, and urban street design practice. It also discusses limitations of the present study and identifies directions for future research, including multi-origin–destination networks, dynamic assignment, and integration with land-use modeling.

Together, these chapters advance from theoretical context to methodological development and applied interpretation. The thesis as a whole demonstrates how equilibrium-based network analysis can serve not only as a predictive modeling tool but also as a diagnostic and exploratory instrument for understanding and guiding transportation and urban design interventions.

2 | Literature Review

2.1. Classical Braess' Paradox

The Braess' paradox is foundational in transportation network theory because it exposes a tension between infrastructure-based intuition and equilibrium-based behavioral response under congestion [4, 21]. The phenomenon was introduced by Braess in 1968 and subsequently disseminated through canonical examples that clarified its relevance for traffic assignment [21, 45]. Early analytical clarification was provided by Murchland, while later work established that paradoxical worsening can arise in broader network classes that contain similar substructures [22, 30, 32].

A central planning implication is methodological rather than anecdotal: adding capacity or constructing a new link does not guarantee improved performance once users adapt routes at equilibrium [5, 46]. In the canonical four-node construction, adding a connecting link between intermediate nodes can increase equilibrium travel time relative to the pre-expansion equilibrium, despite the added link being locally attractive under fixed flows [21, 22, 47]. The mechanism is not imperfect information but decentralized optimization with congestion externalities, where each user internalizes private travel time but not delays imposed on others [4, 48].

From a game-theoretic perspective, the Braess' paradox is a concrete instance of inefficiency in selfish routing, where expanding the strategy set can worsen equilibrium welfare [31, 49, 50]. Price of Anarchy results formalize the gap between equilibrium and system-optimal outcomes and provide bounds showing that topology changes can increase worst-case latency under selfish behavior [48, 51, 52]. For transportation planning, the strength of this strand is its conceptual clarity and worst-case guarantees; its limitation is that worst-case bounds do not directly identify intervention families that preserve equilibrium flows under realistic demand and cost scaling [44, 53].

Classical transportation treatments also shaped evaluation practice by motivating equilibrium-based assignment as a baseline tool to capture the coupled loop between supply, congestion, and endogenous route choice [4, 5]. However, many classical discussions remain centered on whether a specific add/remove operation yields worsening, rather than characterizing families of coordinated network-wide modifications that can produce comparable equilibrium outcomes [47, 54]. This distinction is crucial for urban interventions implemented as coordinated cost and capacity changes across many links, where the key design risk is diversion onto sensitive local streets rather than only aggregate delay [18, 19].

Accordingly, the present thesis treats the classical Braess setting as the methodological

foundation: it provides a clean benchmark for verifying equilibrium solvers, reporting conventions, and the core mechanism of path-set expansion under congestion coupling [4, 45]. Building on this foundation, the thesis shifts the focus from isolated paradox identification to a parameter-space approach that maps paradox and non-paradox regimes and then searches for intervention families that preserve equilibrium outcomes across specifications [5, 54].

The classical Braess' paradox is grounded in user equilibrium (UE), whose behavioral interpretation originates from Wardrop's first principle [3]. Wardrop equilibrium states that all routes used between an OD pair have equal and minimal cost, and no traveler can reduce cost by unilaterally switching routes [3, 55].user equilibrium (UE) admits an equivalent interpretation as a Nash equilibrium (NE) of a nonatomic congestion (routing) game, in which an infinitesimal traveler cannot affect link costs individually [48, 49, 52]. This interpretation clarifies why equilibrium route choices need not be socially optimal: travelers minimize private generalized costs while ignoring congestion externalities imposed on others. The resulting efficiency loss can be formalized and bounded using the Price of Anarchy framework [48, 50].

In deterministic user equilibrium (DUE), costs are deterministic and identically perceived, which isolates the network mechanism from perception error and taste heterogeneity [2, 4]. This baseline is analytically important because paradoxical worsening arises even under perfect information and homogeneous perception, so it cannot be dismissed as a behavioral artifact [21, 22]. It is also mathematically tractable: DUE admits convex optimization and variational inequality formulations under standard monotone separable link costs, enabling existence, stability, and sensitivity reasoning [54, 56]. These properties make DUE a suitable starting point for systematic scanning and for defining equivalence targets that are comparable across scenarios [5].

SUE formulations introduce stochastic perception or random utility, generating probabilistic path splits even under identical physical flows [26, 57]. Behavioral relaxations such as bounded rationality or mixed equilibrium behaviors can shift paradox prevalence and magnitude, which matters for robustness checks [29, 42]. In this thesis, these strands are treated as extensions of the DUE benchmark: DUE establishes the mechanism and a controlled baseline, while SUE and mixed behaviors define an additional axis for testing whether identified regimes and equivalences persist under richer behavior [28, 58].

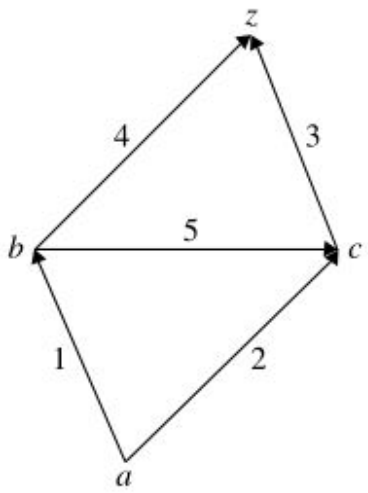


Figure 2.1: Braess network [21]. Node a is the origin and node z is the destination; the intervention adds the connecting link $u_5 : b \to c$. A standard cost specification under which the Braess paradox occurs (with link indices as in the figure) is $t_1(\phi) = 10\phi$, $t_2(\phi) = 50 + \phi$, $t_3(\phi) = 10\phi$, $t_4(\phi) = 50 + \phi$, and $t_5(\phi) = 10 + \phi$, with total demand $\Phi = 6$, where ϕ denotes link flow.

In the canonical construction, two parallel OD routes are composed of links with increasing travel time functions, often represented as affine costs for transparency [21, 47]. Before adding the central link, equilibrium splits across the two routes (under symmetry) so that used paths satisfy equal travel time [55]. Adding the connecting link introduces a composite path that appears locally shorter under fixed flows and attracts travelers from both original routes [45].

As users switch, congestion rises on critical segments and the new equilibrium can have higher travel time than the pre-addition equilibrium, producing the paradox regime [31]. The key point is structural: the intervention changes the feasible path set and therefore the equilibrium structure itself, rather than merely adjusting parameters on a fixed set of alternatives [5]. Experimental studies suggest that Braess-type outcomes can be reproduced under controlled incentive settings, supporting its interpretation as a behavioral-equilibrium phenomenon rather than a mathematical curiosity [33].

A consistent finding is that Braess' paradox does not occur for all demand levels but emerges within critical intervals [30, 47]. At low demand, congestion feedback is weak and the added link tends to improve or leave unchanged equilibrium performance; paradoxical worsening is most likely at intermediate demand where marginal congestion coupling is strongest [54]. This demand dependence motivates parameter-space scanning, because paradox is best characterized as a region in demand–cost space rather than a binary property of topology alone [5].

The planning relevance is aligned with induced-demand evidence, which documents that

capacity expansion can generate additional travel that erodes expected congestion relief [13, 15]. Conversely, capacity reduction syntheses show that traffic may reroute, reschedule, shift modes, or be suppressed, implying that network-wide reallocations can be absorbed without proportional congestion increase in some contexts [18, 19]. These strands jointly support an equilibrium-feedback view: interventions must be assessed under behavioral response rather than static assignment [1].

Robustness studies show that Braess-type substructures can be prevalent in larger networks, implying that paradox risk is not confined to a single four-node example [30, 32]. Extensions also examine persistence under stochastic, mixed, or dynamic settings, where boundaries and magnitudes can shift but Braess-like effects may remain [28, 40].

Braess-like paradoxes have also been reported in mechanical, electrical, biological, and oscillator networks, indicating that a redistribution-under-constraints mechanism can recur across domains [59, 60]. In the present thesis, these cross-domain results are cited only to support the generality of the mechanism and to situate Braess as a broader network phenomenon; they are not studied in depth because they are weakly connected to the thesis main line, which is equilibrium traffic assignment and design-relevant interventions in urban street networks [36]. This scope choice is deliberate: deep engagement with cross-domain physics or circuit analogies would increase breadth without improving the thesis ability to identify traffic-equilibrium equivalences and intervention families in planning contexts [4, 5].

At the urban design scale, network morphology and connectivity shape feasible path sets and thus condition equilibrium redistribution after interventions [8, 11]. Tools for street-network analysis facilitate empirical diagnosis of real topologies and make topology-aware evaluation operational [12]. However, descriptive morphology alone does not determine equilibrium outcomes; translation requires a parameter-sensitive mapping between cost changes and equilibrium redistribution [54]. This motivates the thesis emphasis on computational scanning and equivalence identification as a bridge from equilibrium theory to intervention logic [5].

In summary, the classical Braess' paradox establishes that under DUE, link additions can worsen equilibrium performance within critical demand regimes and that outcomes depend jointly on demand, cost structure, and topology [21, 47]. The literature also indicates where the main opportunities for extension lie: broadening cost specifications beyond the canonical affine forms, comparing DUE and SUE as behavioral regimes, and generalizing from the diamond network to richer topologies where shortcut density and placement matter [28, 61, 62]. Yet existing strands are often fragmented: some focus on theoretical bounds, others on isolated examples, and many do not provide a systematic parameter-space tool to identify intervention families that preserve equilibrium outcomes under coordinated cost and capacity adjustments [44, 54].

Building on this gap, the present thesis develops an atlas-based computational framework to scan demand, cost, and topology spaces to map paradox regimes and to identify equivalence regimes in which distinct network-wide modifications yield comparable equilibrium outcomes under specified behavioral assumptions [5, 54]. This choice is also design-motivated: interventions such as arterial lane narrowing are implemented as coor-

dinated changes, and the relevant success criterion is whether equilibrium flows can be stabilized sufficiently to avoid undesirable diversion while enabling public-space reclamation [19, 63].

2.2. Cost function generalizations

The specification of link cost functions is a central modelling choice in equilibrium analysis, because it determines how congestion is translated into path costs and therefore into network-wide performance under endogenous rerouting [3–5]. In Braess-type settings, cost functions play a dual role: they represent physical/operational constraints (e.g., capacity and delay growth) and they shape the analytical and numerical conditions under which paradoxical regime shifts can occur [54, 55]. Accordingly, this thesis treats cost specification as part of the research design rather than as a purely technical detail: linear costs provide the main computational backbone (including symbolic derivations and large parameter scans), while BPR costs are adopted as the primary nonlinear benchmark for testing whether the identified mechanisms and equivalence patterns remain valid under capacity-normalised convex congestion [1, 2, 21].

A practical implication is that different function families can change

(i) the sensitivity of equilibrium to topology modifications,

(ii) the width of critical-demand windows, and

(iii) the severity of paradoxical outcomes [28, 31, 54].

For this reason, the subsection is organised from the linear family used throughout most of the thesis workflow to increasingly nonlinear families used for robustness checks and selected validations, rather than aiming to exhaust all possible performance functions in the literature.

Classical analytical demonstrations of the Braess' paradox typically rely on linear link costs because they preserve separability and monotonicity while enabling closed-form equilibrium comparisons in small networks [21, 22, 45]. In this thesis, linear costs are not only a pedagogical baseline: they constitute the primary cost family for the main computational pipeline, including symbolic calculations of equilibrium conditions and equivalence patterns, as well as large-scale numerical scanning [4, 54, 56]. The adopted specification is

$$t_a(x_a) = \alpha_a + \beta_a\, x_a, \qquad a \in A, \tag{2.1}$$

which matches the canonical linear constructions used in the Braess literature and in equilibrium assignment textbooks [4, 5, 21].

Here $t_a(x_a)$ denotes the travel time (generalised cost) on link a as a function of link flow x_a (veh/h), consistent with standard static assignment notation [3, 4, 64]. The intercept α_a (time units) represents free-flow or baseline time, capturing fixed components such as length/speed and average control delay, while β_a (time per unit flow) represents marginal congestion sensitivity that governs how strongly rerouting feeds back into increased costs

[1, 2, 5]. Because Braess-type effects are triggered by the interaction of path-set expansion with congestion coupling, β_a is a direct driver of whether a topology change produces only a small redistribution or a regime shift in the used-path set [30, 31].

Linear costs have three main advantages for the thesis objectives. First, they support transparent mechanism interpretation: paradoxical worsening can occur even without strong nonlinearity, which helps isolate the structural equilibrium mechanism from calibration artefacts [21, 22, 45]. Second, under separable monotone linear costs, deterministic user equilibrium can be written as a convex Beckmann problem, which enables robust computation and sensitivity reasoning that are essential for atlas-style mapping and equivalence identification [54–56].Here, the reference to a "Beckmann problem" means that the deterministic user equilibrium can be represented via a convex potential formulation under standard assumptions (continuous, separable, monotone link costs). The Beckmann objective is not introduced as a behavioural welfare criterion; rather, it is a mathematical device whose minimiser is equivalent to Wardrop's equilibrium conditions, and it should not be conflated with the system-optimal objective that minimises total travel time. For the explicit objective form and its link–path constraints, see Equation 3.9.

Third, linearity is compatible with symbolic manipulation at scale, which is central to deriving and verifying equivalence conditions across interventions in later chapters [4, 54].

The main limitation is representational realism: proportional delay growth can understate acceleration near capacity and thus may under-represent how sharply equilibrium incentives change in heavily loaded regimes [2, 65]. For this reason, nonlinear families are used as targeted robustness extensions, with BPR as the principal nonlinear benchmark because it is both widely used in practice and naturally parameterised by capacity [1, 5].

A common intermediate step between linear theory and capacity-based practice is to introduce strictly convex polynomial costs (often quadratic) to capture accelerating marginal delay as flow increases [2, 65]. A representative form is

$$t_a(x_a) = \alpha_a + \beta_a\, x_a + \gamma_a\, x_a^2, \qquad a \in A, \tag{2.2}$$

which preserves separability while making marginal delay increasing in x_a through the curvature parameter γ_a [4, 5].

The key benefit of convex polynomial costs is diagnostic: they allow the thesis to test whether equivalence and paradox boundaries identified under linear costs are stable when congestion curvature amplifies the penalty of flow concentration on critical links [54, 66]. They can also improve numerical stability in some scanning tasks because strong convexity regularises the equilibrium mapping [55, 64]. However, quadratic and polynomial functions remain stylised: parameters are not directly tied to capacity in the way practitioners interpret level of service, and cross-study comparability is often weaker than for capacity-normalised families [1, 2]. For these reasons, polynomial costs are treated as secondary validations in selected experiments, whereas the main nonlinear extension is implemented with BPR-type functions [5, 54].

For practical static assignment and planning evaluation, the Bureau of Public Roads (BPR) function is a widely adopted link performance model because it is smooth, monotone, and explicitly normalised by capacity [1, 2, 5]. The thesis adopts the standard form

$$t_a(x_a) = t_a^0 \left[1 + \alpha \left(\frac{x_a}{C_a}\right)^{\beta}\right], \qquad a \in A, \tag{2.3}$$

as the principal nonlinear benchmark for extending the main findings beyond the linear setting [1, 2].

In (2.3), t_a^0 is the free-flow travel time, C_a (veh/h) is nominal capacity, and (α, β) are dimensionless shape parameters controlling the magnitude and convexity of delay growth [2, 5]. Compared with linear costs, BPR has two key advantages for the thesis goals. First, it captures the empirical pattern that delay grows slowly at low utilisation but escalates rapidly near capacity, making it an appropriate stress-test for whether paradoxical and equivalence regimes persist under stronger nonlinear feedback [65, 66]. Second, because costs are expressed through the ratio x_a/C_a, interventions such as lane narrowing or capacity reallocation can be represented directly and comparably across links and network states, which aligns with the thesis focus on network-wide modifications rather than a single added-edge toy change [5, 44].

At the same time, nonlinear congestion does not mechanically guarantee more paradox: stronger convexity can either amplify or suppress paradoxical regimes depending on how it penalises over-utilisation of the newly attractive options, which is why the thesis treats paradox absence under some BPR settings as informative boundary evidence rather than as a modelling failure [28, 31, 58]. This is also the reason BPR is prioritised over many alternative nonlinear forms: it offers a widely recognised, capacity-based reference model that is sufficiently representative for robustness claims, while keeping the parameter space interpretable and comparable across scans [1, 2, 5].

Beyond BPR, planning practice includes alternative performance functions designed to capture specific regimes, including Davidson-type and Akçelik-type formulations [2, 61]. These functions can shift curvature and therefore move paradox boundaries under the same topology and demand, which makes them useful as occasional cross-checks [54]. However, they are not used as the main nonlinear family in this thesis for two reasons. First, the thesis requires a unified nonlinear benchmark across many experiments; BPR provides the most standardised, capacity-normalised choice and is more directly comparable to standard assignment practice in the planning literature [1, 5]. Second, expanding all nonlinear families systematically would dilute the main contribution (equivalence-oriented scanning across topology and behavioural regimes) by multiplying calibration degrees of freedom without proportionate gain for the design-relevant questions addressed later [44, 54].

In summary, the cost-function strategy is intentionally selective. Linear costs are the primary vehicle for mechanism explanation, symbolic derivation, and large-scale atlas computation, while BPR costs provide the principal nonlinear extension for robustness and planning relevance; other nonlinear forms appear only in targeted validation steps

where they help confirm that observed patterns are not artefacts of a single calibration family [5, 45, 54].

2.3. Deterministic versus stochastic equilibrium

The occurrence, severity, and interpretation of Braess' paradox depend not only on network topology and cost functions, but also on the behavioural assumptions embedded in the equilibrium concept that closes the feedback loop between route choice and congestion [4, 5]. In the classical traffic assignment tradition, the dominant paradigm is Deterministic User Equilibrium (DUE), which operationalises Wardrop's first principle under the idealisation that travellers perceive costs accurately and choose minimum-cost routes in a best-response manner [3, 55]. A second, increasingly influential, paradigm is stochastic user equilibrium (SUE), which relaxes perfect information by introducing perception error and heterogeneity in route choice, thereby producing probabilistic path splitting even when cost differences exist [26, 67]. More recent behavioural research further argues that neither perfectly rational best-response nor purely random error is sufficient for describing real route choice; instead, travellers may exhibit bounded rationality, inertia, and satisficing (i.e., choosing an alternative that is "good enough" rather than strictly optimal), motivating equilibrium concepts that remain Wardrop-like in spirit but differ in switching thresholds and stability properties [5, 42].

This section reviews these equilibrium notions from a literature perspective and clarifies why the DUE–SUE distinction is particularly consequential for Braess-type phenomena when the thesis question concerns network-wide modifications that may yield equivalent equilibrium flows rather than simple monotone improvement [4, 54]. Importantly, the purpose here is conceptual rather than algorithmic: numerical solution methods and implementation details are deferred to Chapter 3, so that the present discussion can focus on how behavioural postulates shape paradox mechanisms, robustness across regimes, and the interpretation of design and policy interventions [5, 64]. Within this thesis workflow, DUE provides the primary analytical platform (including symbolic derivations), whereas SUE is used selectively as a robustness extension to test whether DUE-based findings persist under dispersed route choice [28, 29].

DUE formalises Wardrop's first principle by asserting that, at equilibrium, no traveller can unilaterally reduce their travel cost by switching routes [3]. In practical terms, for each origin–destination (OD) pair, all used paths have equal and minimal cost, while all unused paths have cost no less than this minimum [4, 55]. This equilibrium concept has become the canonical behavioural baseline in traffic assignment, both because of its analytical tractability and because it corresponds to a clear normative interpretation: equilibrium reflects selfish route choice under full information and instantaneous adjustment [5, 64].

Theoretical work clarifies when such an equilibrium exists and whether it is unique, which is essential if paradox statements are to be interpreted as well-defined comparative statics across network modifications [54, 64]. Under standard assumptions—separable link costs that are continuous and monotone increasing in link flow—Wardrop equilibrium exists,

and under stronger conditions (e.g. strict monotonicity) the equilibrium is unique in link flows [5, 55]. These properties matter in paradox analysis because they delimit when a network change (such as adding a link) can be interpreted as shifting the equilibrium to a different but well-defined outcome rather than creating multiple ambiguous steady states whose selection could confound paradox diagnosis [5, 54].

Within the Braess literature, DUE is historically dominant and provides the cleanest benchmark for isolating the structural mechanism of equilibrium deterioration after a topology change [22, 45]. Braess' original paradox construction shows that adding a seemingly beneficial link can increase equilibrium travel time because individual best-responses collectively induce a less efficient flow pattern [21, 22]. Subsequent analytical and computational studies confirmed that the paradox is not an idiosyncrasy of a single example but a structural possibility of selfish routing in congested networks, including cases where embedded substructures trigger system-level worsening [30, 31]. From this viewpoint, Braess' paradox can be understood as an efficiency loss generated by decentralised route choice, closely related to price-of-anarchy reasoning in congestion games and its transportation interpretations [31, 68].

Yet, as many authors emphasise, DUE is behaviourally stringent and can overstate the sharpness of behavioural response in real networks [5, 67]. Because travellers are assumed to respond deterministically to arbitrarily small cost differences, DUE can predict unrealistically abrupt switching and corner solutions where virtually all flow concentrates on a small subset of routes [27, 67]. In paradox settings, this determinism can amplify the mechanism: once a new link creates a slightly cheaper-looking composite route, DUE allows a large mass of travellers to move to it, potentially pushing the network into a high-congestion regime that makes everyone worse off [30, 31]. These concerns motivate extensions that embed imperfect perception and heterogeneity, which is the core role of SUE in the subsequent literature and a key axis along which DUE-based "equivalent" network-wide interventions must be tested for robustness [26, 58].

SUE generalises Wardrop's idea by recognising that travellers do not observe or process costs perfectly and that unobserved attributes produce heterogeneous perceived utilities [26, 67]. Instead of assuming that each traveller deterministically selects the minimum-cost path, SUE posits that perceived utility (or perceived cost) contains a random component, reflecting perception error, taste heterogeneity, and unobserved attributes [5, 67]. As a result, route choice is described by probabilities, and equilibrium requires that the resulting expected path flows be consistent with the congestion-dependent costs produced by those flows [5, 27].

A classical operational form of SUE uses a Logit route choice model, which yields smooth probabilistic splitting and avoids the knife-edge switching implied by DUE [26, 67]. For an OD pair (r, s) and a feasible path set P^{rs}, the probability of choosing path $p \in P^{rs}$ is commonly written as

$$\Pr(p \mid r, s) = \frac{\exp(-\theta^1 C_p(\mathbf{x}))}{\sum_{q \in P^{rs}} \exp(-\theta^1 C_q(\mathbf{x}))}, \qquad p \in \mathcal{P}_{rs}[5,\ 26]. \tag{2.4}$$

In (2.4), $C_p(\mathbf{x})$ denotes the generalised cost of path p computed from the link-flow vector $\mathbf{x}$, and θ is a dispersion parameter (with units inverse to cost) that controls travellers' sensitivity to cost differences [67, 69]. As $\theta \to \infty$, choice concentrates on minimum-cost paths and SUE approaches the DUE outcome, whereas $\theta \to 0$ yields nearly uniform splitting, representing extreme dispersion [5].

For Braess-type analysis, the key benefit of SUE is that it provides a behavioural robustness dimension. Probabilistic splitting can smooth the reallocation of flow after a network modification and may damp abrupt concentration effects that are accentuated under DUE [67]. At the same time, SUE does not automatically "remove" paradoxes: a substantial body of work shows that paradoxical deterioration can persist under SUE-type models, although it may become smoother and more parameter-dependent than in deterministic benchmarks [28, 29]. The interpretation therefore shifts from a single sharp equilibrium switch to a behaviourally parameterised phenomenon whose magnitude and even sign can vary with dispersion and information structure [67, 70].

The literature contains richer behavioural extensions, including bounded rationality, inertia, and satisficing equilibria, as well as mixed populations in which different user groups follow different decision rules [29, 42, 43]. These approaches are important for interpreting real-world route choice, but they also introduce additional behavioural thresholds, learning dynamics, and heterogeneity structures that substantially expand the model space and reduce comparability across the parameter scans central to this thesis [5, 39]. Because the thesis objective is to construct a regime-based atlas and to identify equivalence classes of network-wide interventions under consistent equilibrium definitions, these richer behavioural models are treated as contextual motivation for robustness rather than as primary modelling layers [4, 67]. Accordingly, the remainder of the thesis focuses on DUE as the main analytical baseline and uses SUE as the principal robustness extension, which already captures the key contrast between deterministic best-response and dispersed probabilistic choice [28, 58].

The DUE–SUE distinction is not only a modelling preference but also a methodological constraint imposed by the thesis research design. A core contribution of the thesis is to identify equivalence patterns across network-wide interventions and to derive regime boundaries through symbolic and semi-analytic reasoning. For this purpose, DUE is the primary platform because its equilibrium conditions can be expressed with used-path equal-cost relations that remain algebraic under the linear cost specifications employed in the symbolic parts of the thesis [54, 55]. By contrast, standard SUE formulations combine exponential choice terms with congestion-dependent costs, typically producing transcendental fixed-point relations that are not amenable to closed-form symbolic regime derivation in the same way [5, 67]. This analytical limitation is the main reason why SUE is not used as the default assumption in the symbolic equivalence workflow.

Nevertheless, using only DUE would risk over-interpreting equivalence and paradox patterns as behaviourally universal. For this reason, SUE is introduced as a targeted extension used to test robustness: the thesis varies the dispersion parameter θ in (2.4) to span regimes from near-deterministic routing to highly dispersed choice and checks whether

the DUE-derived conclusions remain qualitatively stable under probabilistic splitting [28, 29, 58]. This positioning keeps the main analytical pipeline consistent and interpretable while still meeting the behavioural credibility requirement needed for planning and design translation in later chapters [4, 5, 64].

2.4. From Diamond to Complex Networks

While Braess' original example relies on a minimal four-node diamond network, subsequent research suggests that paradoxical deterioration is not confined to this specific configuration but is connected to broader structural properties of the underlying graph [21, 22, 30]. In particular, topology shapes how many competing routes exist, how they overlap on shared links, and how marginal congestion costs propagate through the path set after a perturbation such as adding a link [30, 32, 54]. Extending the analysis from the canonical diamond to more complex structures has therefore become a recurring theme in the literature, both to test generality and to clarify structural conditions under which paradoxical equilibria may be more or less likely [30–32].

A first reason topology matters is that Braess-type deterioration is an equilibrium phenomenon: adding a link changes the feasible path set and thereby changes the behavioural best-response mapping that defines equilibrium [4, 45, 48]. A second reason is that topological redundancy does not automatically imply system-wide efficiency, because redundancy can enable individually attractive but collectively harmful flow patterns, especially when routes share bottleneck links in different combinations [22, 31, 68]. Accordingly, a substantial literature has moved from "the Braess network" as a single example toward Braess motifs and classes of networks where paradoxes may or may not occur, often with an emphasis on comparability across families rather than on any single canonical instance [30, 32, 53].

Topology-oriented research spans

(i) graph-theoretic characterisations of when paradoxes are possible or impossible, and

(ii) structured intermediate families used to scale the classical mechanism beyond one toy network.

The present thesis cites both strands to establish generality, but it adopts the second as its main empirical laboratory. In particular, the computational and symbolic workflow in later chapters remains anchored in the classical diamond as the primary baseline, and then extends to ladder networks and multi-bypass variants as the main topology generalisation. More abstract results on broad graph classes are therefore reviewed only insofar as they motivate why topology matters, but they are not reproduced or exhaustively tested here, because the thesis objective is not to prove new impossibility theorems for arbitrary graphs but to build design-relevant diagnostics on controlled, enumerable families under consistent equilibrium solvers [4, 5, 54].

Early extensions focused on perturbations of the classical diamond network because it provides a simple topology that contains two disjoint alternatives plus a cross-connection

whose addition can change equilibrium structure [21, 22, 45]. Reformulations of the example often emphasised planning interpretation rather than algebraic novelty: the core lesson is that project appraisal must anticipate endogenous re-routing, not merely compare free-flow or partial-equilibrium travel times [1, 4, 22]. Within this family of near-diamond networks, authors explored asymmetry in link costs, alternative bypass placements, and additional parallel links to indicate that paradoxical outcomes can persist beyond the perfectly symmetric textbook case [30, 47].

The diamond remains central in this thesis for two methodological reasons. First, it is the minimal topology in which the Braess mechanism can be stated transparently as a change in the used-path set induced by an added connector, making it the natural reference point for interpreting later topological generalisations [4, 45]. Second, much of the thesis verification and equivalence analysis relies on symbolic derivations under linear costs, where the diamond provides an analytically tractable benchmark against which numerical solvers and regime classification can be validated before scaling to larger families [54, 55]. For these reasons, even when the thesis explores larger topologies, the classical diamond is retained as the principal "unit test" network used repeatedly for both numerical and symbolic checking rather than being treated only as a historical illustration [47, 54].

A useful conceptual development in the diamond literature is the recognition that paradox tends to arise when a perturbation creates a new route that is locally attractive for many users but globally increases congestion on shared components [31, 48]. From the congestion-game viewpoint, adding an edge modifies the strategy space and can increase equilibrium latency even when the social optimum would benefit, illustrating a gap between selfish and system-optimal routing [53, 68]. This interpretation makes explicit why topology is central: whether an added edge can create such globally harmful strategy shifts depends on the combinatorial arrangement of overlaps among routes, not on any single link attribute alone [30, 32].

The discussion by Pas and Principio [47] helped consolidate the idea that paradox is often best understood through demand regimes and equilibrium path-usage patterns rather than as a binary yes/no property of a single configuration [47, 54]. In particular, the same topology can be paradoxical in intermediate demand ranges but non-paradoxical at very low or very high demand, implying that topology and cost scaling jointly shape critical thresholds [47, 66]. This regime perspective motivates later work that scans parameters and network variants to map paradox regions in a topology–parameter space rather than validating only isolated examples [35, 54].

An influential contribution in the topology literature is the prevalence analysis of Steinberg and Zangwill, who showed that paradoxical deterioration can arise across broad classes of separable, monotone increasing link cost functions, suggesting that the phenomenon is not restricted to a single numeric calibration [30, 55]. Their argument shifts attention from specific coefficients to the existence of interacting path patterns whose costs respond differently to flow redistribution, which is fundamentally a graph-structural condition [30, 32]. Methodologically, this move is important because it reframes Braess analysis as

a question about network classes and subgraph motifs, aligning it with the idea that local structures can generate global vulnerability [30, 35].

Motif-based interpretations also connect naturally to planning intuition: a new connection can be risky when it creates a shortcut that attracts flow through a bottleneck [1, 4]. In equilibrium terms, this corresponds to an added edge increasing the overlap of many routes on a congestible component, pushing equilibrium toward a more concentrated allocation [31, 68]. Because overlap patterns are topological, two networks with similar numbers of links can behave differently depending on whether their cycles and cross-connections create diamond-like competitive structures [9, 32].

This thesis uses prevalence and motif insights primarily as justification for systematic scanning on controlled families. It does not attempt to enumerate motif occurrence in arbitrary graphs or to validate prevalence theorems computationally across large random networks, because such work would shift the thesis away from intervention equivalence and toward network-science style statistical characterisation [34, 35]. Instead, the thesis adopts structured families (diamond generalisations and ladder templates) where motifs are built-in by design and where topology parameters can be enumerated transparently, allowing the resulting regime maps to be interpreted as intervention-relevant diagnostics rather than as graph-theoretic taxonomy [30, 54].

A substantial theoretical development came with the identification of network classes in which Braess' paradox cannot occur in a strong sense, most notably the class of series–parallel graphs [32, 48, 68]. Milchtaich showed that, for series–parallel networks, equilibrium is Pareto efficient with respect to link additions: adding a link cannot make all users worse off under standard assumptions [32, 55]. This result is consequential because it connects Braess' paradox to graph structure: paradox requires the presence of a non-series–parallel substructure in the classical deterministic setting [30, 32].

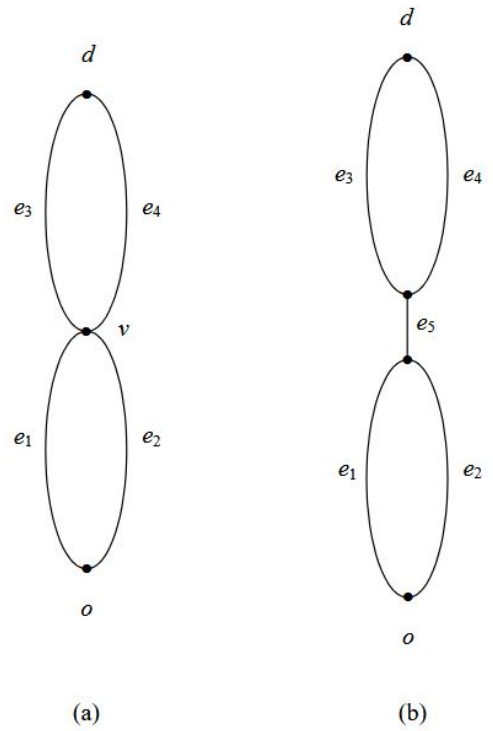


Figure 2.2: Series–parallel configurations used by Milchtaich [32] to illustrate an impossibility result for Braess-type worsening. Panel (a) is a two-terminal series–parallel network obtained by a series composition of two parallel subnetworks: two parallel edges (e_1, e_2) connect the origin o to the intermediate node v, and two parallel edges (e_3, e_4) connect v to the destination d. Panel (b) modifies panel (a) by inserting an additional series edge e_5 between the two parallel subnetworks.

In the present thesis, series–parallel results serve as a conceptual boundary condition rather than a target for reproduction. Two reasons motivate this choice. First, urban street networks are rarely series–parallel; they contain cycles, grids, and multi-scale cross-connections, so impossibility classes are difficult to apply directly as design rules without substantial additional modelling of turning movements, control delay, and multimodal constraints [9, 11]. Second, the thesis emphasis is not on proving whether a given real network is Braess-proof, but on constructing equivalence-oriented intervention families and testing them on controlled topologies where mechanisms can be diagnosed and compared across cost and behavioural assumptions [5, 54]. For this reason, the series–parallel literature is retained for interpretive context but is not pursued as a dedicated research axis in later chapters [31, 32].

Between the minimal diamond and a full-scale urban network lies a useful class of structured graphs that allow controlled scaling of path interactions while keeping interpretation relatively transparent [9, 32]. A prominent construct in this category is the ladder-like

network, which can be understood as two parallel rails connected by repeated transversal links, effectively concatenating multiple diamond-like interaction zones along the OD direction. This construction is particularly compatible with the thesis aims for three reasons.

First, ladder networks preserve a clear structural similarity to the classical diamond. Each rung or bypass creates a local cross-connection that can generate the same type of locally attractive shortcut and shared-link congestion coupling that drives the original Braess mechanism, but now replicated across multiple stages. This makes ladders a natural next step after the diamond: they generalise the same equilibrium logic rather than introducing an unrelated topological pattern [30, 32]. Second, ladders offer a compact parameterisation of topology. One can scale network size (number of stages) and vary bypass placement and density while keeping the overall template fixed, which is essential for producing comparable atlas-style regime maps under consistent solvers [35, 54]. Third, ladders remain computationally manageable for systematic scanning and for controlled comparisons between DUE and SUE, whereas more general topologies quickly lead to a combinatorial explosion in path structure and intervention configurations that would dilute interpretability [5, 54].

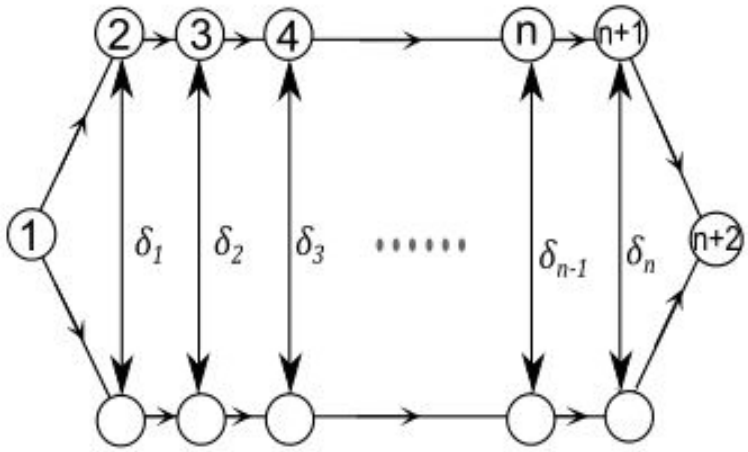


Figure 2.3: ladder Network[62]

A key reference aligned with this choice is Toyota and Ogura [62], which formulates Braess-like paradoxes on ladder networks as an intermediary between the canonical diamond setting and more complex networks [62]. Their results support the idea that when topology offers multiple bypass positions, paradoxical degradation is often better interpreted as a regime phenomenon shaped by bypass placement, bypass multiplicity, and the interaction between bypass costs and rail costs [54, 62]. This regime-based framing aligns directly with the thesis goal of constructing parameter-sensitive maps and identifying intervention families that preserve (or approximately preserve) equilibrium outcomes.

The decision not to adopt other network generalisations follows from the same interpretability constraint. General modular graphs, random networks, or large empirical

street graphs are valuable for demonstrating prevalence, vulnerability, or statistical regularities, but they do not provide a clean intermediate step for diagnosing equivalence and for attributing changes in outcomes to a small set of controlled topological parameters [34, 35]. Similarly, multi-OD and multi-commodity networks are central to road network design, but they introduce interaction effects across OD pairs that would require a different experimental design and would obscure the thesis focus on mechanism explanation, symbolic validation, and controlled equivalence classes [31, 44]. Accordingly, these literatures are cited as context for future scaling, but the thesis concentrates on ladder families and multi-bypass variants as the main topology expansions that remain structurally close to the diamond and compatible with systematic scanning [54, 62].

Beyond single-OD stylisations, later work examines more general networks and indicates that paradox severity can change when multiple OD pairs interact through shared infrastructure [31, 68]. In multi-commodity settings, selfish routing can create congestion spillovers across OD pairs, raising the stakes of network design and making robustness a central concern [35, 53]. The thesis does not pursue these settings directly because its main contributions require repeated, comparable regime scans and symbolic verification steps, which are substantially more difficult to maintain in multi-OD configurations where equilibrium structure depends on cross-OD interactions and larger path sets [5, 54].

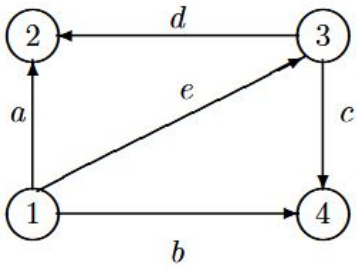


Figure 2.4: A multi-OD example network illustrating multi-commodity interactions under shared infrastructure. Nagurney and Qiang [35].

Similarly, theoretical and algorithmic studies explore how to avoid or mitigate Braess' paradoxes through constraints, incentives, or topology-aware controls [48, 71]. These approaches are relevant to policy design, but the present thesis focuses instead on equivalence-oriented intervention families (e.g. coordinated capacity adjustments) and on diagnosing when equilibrium patterns can be preserved while reallocating street space, which requires a controlled laboratory where intervention parameters can be scanned systematically [44, 54].

Cross-domain Braess-like effects in mechanical, electrical, and oscillator networks support the claim that the phenomenon is rooted in general properties of networked systems under decentralised or locally optimal flow allocation [36, 59, 60]. In this dissertation, such evidence is used only to motivate the generality of the mechanism; these domains are not analysed further because their modelling primitives, performance measures, and

control variables differ from the transport equilibrium and design-intervention setting that the thesis targets [31, 35].

A further strand links Braess-relevant topology concepts to urban street-network representation and morphology. Urban networks have additional geometric, hierarchical, and configurational properties that shape feasible routes and overlap patterns beyond what is captured in minimal graph models [8, 9, 11]. Tools such as OSMnx enable systematic acquisition and analysis of large-scale urban street graphs, making it possible to operationalise motif thinking at scale [12]. These approaches are relevant for future application and empirical correspondence, but the present thesis prioritises controlled topology families to keep mechanisms diagnosable and to maintain comparability across cost and behavioural regimes in atlas-style scans [5, 54].

The progression from diamond networks to structured and generalised networks yields three implications for Braess research. First, it suggests that paradox is not restricted to minimal examples but can arise in networks with structural complexity, especially when non-series–parallel motifs and overlapping route patterns are present [30, 32]. Second, it highlights the importance of identifying repeating motifs—diamonds, multi-stage cross-links, and ladder-like constructions—that can act as local generators of equilibrium inefficiency when embedded within larger graphs [31, 51]. Third, it motivates systematic methods capable of analysing families of networks rather than isolated cases, since paradox outcomes depend on the interaction between topology, cost curvature, and behavioural dispersion [28, 54, 66].

These considerations motivate the methodological approach adopted in this dissertation. The classical diamond is retained as the principal baseline for verification, symbolic derivation, and regime interpretation because it isolates the mechanism and supports transparent comparability across model variants [47, 54]. Ladder networks and multi-bypass variants are then adopted as the main topology extension because they preserve diamond-like interaction logic while enabling controlled scaling and enumeration, which is essential for atlas-style scanning and for identifying equivalence classes of interventions across topology settings [54, 62]. More abstract theoretical generalisations and broader network classes remain important context, but they are intentionally treated as background because the thesis contribution is directed toward design-relevant, parameter-sensitive diagnostics on controlled families rather than toward comprehensive graph-theoretic classification of all paradox-capable networks [31, 32].

2.5. Computational approaches

As Braess-type paradoxes are extended beyond canonical small networks toward larger topologies, nonlinear congestion, and heterogeneous behavioral assumptions, analytical tractability diminishes and computational approaches become central [45, 54]. In this literature, computation serves two recurring roles: validating theoretical insights on controlled motifs and exploring equilibrium regimes that are difficult to access through closed-form derivations [4, 5]. The resulting body of work increasingly combines algorithmic

solvers with parameter-space exploration to identify where paradoxical degradation may occur and how sensitive it is to demand, cost curvature, and behavioral dispersion [35, 58].

A common computational strategy is systematic scanning of multidimensional parameter spaces, which allows paradox to be represented as a regime phenomenon rather than a single counterexample [54, 66]. A second strategy embeds minimal motifs such as the diamond and ladder structures within broader topology families, using repeated equilibrium computation to test whether paradox persists, amplifies, or disappears under structural generalization [30, 62]. These trends support a shift in interpretation, from asking whether paradox exists to asking which network families and behavioral assumptions are most paradox-prone [31].

The Method of Successive Averages (MSA) is among the most widely used computational procedures for user equilibrium because it is simple to implement and scales to large networks [4]. MSA alternates between auxiliary assignments under current costs and averaging updates of flows, approximating a fixed-point iteration of the equilibrium mapping [64, 72]. Its practical robustness explains why it is frequently adopted as the workhorse solver in studies that require repeated equilibrium computation across many scenarios [5].

In the Braess literature, iterative fixed-point schemes have been used to reproduce paradox effects under increasing levels of complexity, including nonlinear congestion and stochastic perception [47, 58]. They are also suited to regime mapping because the same computational routine can be executed thousands of times under controlled variations of demand, capacities, and cost parameters [54, 62]. At the same time, authors repeatedly note that MSA may converge slowly near equilibrium and that paradox boundaries can be sensitive to numerical tolerances, which complicates cross-study comparison when convergence controls are not reported consistently [4, 54]. A further limitation is interpretive: MSA yields equilibrium flows but does not itself explain why a topology-parameter configuration produces paradoxical deterioration, motivating complementary analytical or structural arguments [31, 47].

A foundational result for deterministic user equilibrium is the Beckmann transformation, which shows that under separable and increasing link cost functions, equilibrium corresponds to the minimizer of a convex potential function [4, 56]. This equivalence underpins the use of mathematical programming methods and supports convergence arguments for classical algorithms such as Frank–Wolfe in structured settings [72, 73]. In the paradox literature, optimization-based viewpoints are valued because they help distinguish genuine equilibrium shifts induced by network modification from numerical artifacts, especially when multiple solvers or calibrations are compared [45, 54]. They also connect naturally to sensitivity analysis, enabling researchers to study how equilibrium solutions vary continuously with demand and cost parameters and to identify structural transitions between regimes [54].

For stochastic user equilibrium, a parallel optimization interpretation arises through entropy-regularized formulations in which expected travel cost is balanced against an en-

tropy term governed by a dispersion parameter [5, 26]. This framework provides a smooth bridge between highly random route choice and near-deterministic behavior as dispersion changes, which has been used to interpret how paradox severity depends on behavioral noise [28, 29]. Comparative studies use this interpretation to explain why some paradox regimes persist under stochasticity while others weaken or shift in parameter space [58].

Variational inequality (VI) formulations provide a unifying representation for deterministic, stochastic, multi-class, and constrained equilibria, generalizing equilibrium conditions beyond cases with a readily available convex objective [74, 75]. From a Braess perspective, VI is useful because link additions or removals alter both the feasible set and the cost operator, clarifying that paradox arises from modifying the equilibrium problem itself rather than from perturbing a fixed solution [31, 54]. Extensions such as quasi-variational inequalities and complementarity formulations support more complex settings, including dynamic interactions and control-dependent constraints [39, 76].

Beyond static equilibria, dynamic traffic assignment and simulation-based models incorporate temporal congestion propagation, queue spillback, and adaptive route choice [38, 40]. Recent work indicates that Braess-like effects can persist under dynamic conditions, suggesting that paradox is not solely a static equilibrium artifact and can interact with temporal dynamics [40]. Behavioral extensions, including bounded rationality and perturbed-utility models, relax perfect information assumptions and provide additional lenses on when paradoxical inefficiency may emerge under learning and imperfect knowledge [42, 69]. At the same time, simulation-based studies are often sensitive to calibration choices and stochastic variation, which can limit the extent of systematic topology-parameter regime mapping compared with structured scanning studies [5, 54].

For small structured networks, symbolic or semi-analytical computation can sometimes derive closed-form equilibrium solutions, demand thresholds, or parameter relations that directly explain when paradox occurs [45, 47]. Such derivations provide mechanism-level interpretation and are frequently used as anchors for validating numerical results on canonical motifs [55]. Hybrid approaches combine motif-level analytical reasoning with numerical scanning on generalized families such as ladders and modular bypass configurations, supporting phase-diagram representations in topology-parameter space [30, 62]. This regime-mapping perspective has become prominent because it reframes paradox as a comparative diagnostic across model families rather than a single illustrative example [31, 35].

Overall, the computational literature suggests that no single paradigm dominates across all settings. Iterative solvers enable scalable scanning, optimization frameworks support validation and sensitivity analysis, VI generalizes equilibrium structure, and dynamic or hybrid methods extend external realism and interpretability in complementary ways [5, 54]. Taken together, these approaches provide the methodological foundation for contemporary Braess research that emphasizes regime characterization across topology, cost specification, and behavioral assumptions [58, 62].

Across the computational literature reviewed above, a recurring methodological goal is to move from isolated paradox demonstrations toward comparable regime characterizations across topology, cost specification, and behavioral assumptions [54, 62]. In this context, the present study positions its empirical contribution at the level of structured regime outputs rather than single-case outcomes [31, 35].

The primary outputs reported in this thesis are paradox regime maps and phase-diagram style representations that classify link interventions into degradation, neutral, and improvement regimes across parameter space [54, 66]. A second output is a set of topology-indexed comparisons across controlled network families, which makes it possible to read paradox propensity as a property that varies systematically with structural generalization rather than as an exception tied to one calibration [30, 62]. A third output is a comparative DUE versus SUE regime assessment, interpreted through the established connection between stochastic choice models and dispersion-controlled smoothing of equilibrium structure [26, 28].

This positioning reflects a practical compromise documented in the literature. Iterative equilibrium computation enables high-coverage scanning and comparable regime plots, while optimization and VI viewpoints provide theoretical framing for interpretation and robustness claims [4, 5, 74]. Dynamic simulation and richer behavioural models (e.g. day-to-day learning, fully dynamic traffic assignment, or heterogeneous populations with inertia) are important for external realism, but they are not required to obtain the central results targeted here. The core contribution of this thesis is a regime-based characterization of paradox occurrence and intervention equivalence across topology and cost specifications, for which static DUE/SUE formulations already capture the essential mechanism: equilibrium reconfiguration induced by path-set expansion under congestion coupling. Accordingly, these richer models are treated as follow-on directions for robustness and empirical calibration, rather than as necessary components of the present methodological pipeline [38, 40].

By explicitly structuring results as regime maps, topology-family comparisons, and DUE/-SUE contrasts, the study aligns with the regime-mapping trend in recent Braess research and prepares the ground for later discussion on how such outputs can inform network design interpretation [31, 58].

2.6. The Braess' Paradox from the Perspective of Urban Design

Although the Braess' paradox was originally formulated as a counterintuitive property of equilibrium traffic assignment on abstract networks, its implications can be read in ways that intersect with urban design and spatial planning [31, 45]. Urban design concerns the physical and spatial configuration of streets, blocks, intersections, and public spaces that structure everyday mobility and accessibility patterns [7, 9]. Because these configurations shape feasible route sets, degrees of overlap, and connectivity hierarchies, they also shape the structural conditions under which Braess-type paradoxes may arise under equilibrium response [30]. Recent urban network analysis provides concrete representational tools for

linking street morphology to graph structure [11, 12]. A concrete mapping helps clarify the bridge between the two domains: adding a connector or opening a restricted street corresponds to an edge addition; closures and modal filters correspond to edge removals or constraints; and interventions such as road diets or lane reallocations correspond to capacity and free-flow-time changes in link performance functions, which jointly modify the feasible path set and the cost mapping that determine equilibrium outcomes [4, 5].

From a planning and design perspective, the paradox complicates the long-standing assumption that increasing connectivity and vehicular capacity necessarily improves network performance [1, 23]. Instead, it indicates that under equilibrium behavioural adaptation, additional links can induce route-choice reorganizations that may increase congestion and travel times in some regimes [45, 48]. This insight resonates with urban design arguments for selective connectivity, modal filtering, and street hierarchies rather than unrestricted vehicular permeability [19, 20]. In interpreting this connection, it is important to keep the scope explicit: the classical Braess mechanism concerns equilibrium re-routing of vehicular traffic under a given demand level, rather than broader multimodal accessibility, land-use feedbacks, or long-run demand changes.

Consequently, the Braess' paradox has gradually been used not only as a traffic-engineering curiosity but also as one theoretical lens for evaluating spatial network interventions in cities [35]. It offers an equilibrium-based explanation for why some connectivity enhancements may underperform expectations, and why controlled disconnection or capacity reduction can, in certain contexts, improve overall performance [18, 32]. This section reviews the emerging dialogue between Braess-type network theory and urban design research, and it outlines the conceptual bridge that motivates the present dissertation.

A recurring limitation in the interdisciplinary literature is that design debates on "connectivity" and "permeability" often remain descriptive, while formal Braess research frequently remains at the level of abstract graphs and inefficiency bounds, making it nontrivial to translate results into actionable spatial diagnostics [9, 31]. Moreover, many applied discussions rely on a small set of emblematic cases without systematically separating topology effects from cost curvature and behavioural dispersion, which are all known to influence paradox regimes [28, 54]. The present thesis contributes by operationalizing a structured set of network templates (diamond, ladder, modular bypass families) and by constructing comparable "paradox maps" across DUE/SUE and cost specifications, thereby providing a controlled bridge from abstract paradox mechanisms to design-relevant interpretations about when added vehicular connectivity may be more or less risky [58, 62].

Urban street systems can be represented as spatially embedded graphs whose nodes correspond to intersections and whose edges correspond to street segments [11, 12]. This representation supports network-theoretic analyses of connectivity, redundancy, and path overlap in real urban fabrics [34]. Configurational theories of space syntax further emphasize that topological depth and choice structures, rather than Euclidean distance alone, are important determinants of observed movement patterns [8].

From the perspective of Braess-type analysis, these representations help clarify that real

street networks can contain diamond-like and ladder-like subgraphs embedded within larger grids and hierarchies [30, 62]. Local design interventions—such as adding a connector road, opening a gated street, inserting a flyover, or reconfiguring intersection priorities—modify these subgraphs in ways analogous to adding links in abstract Braess networks [31]. Thus, urban design decisions can be understood as perturbations of equilibrium routing environments, even when no formal traffic model is explicitly applied [1]. This correspondence is conceptually useful in literature review terms because it specifies what, exactly, is being transferred from the paradox framework into spatial practice: not the full complexity of urban mobility, but the link between local connectivity edits and network-wide equilibrium redistribution under congestion [4].

Empirical analyses of city street networks also suggest strong morphological heterogeneity, with coexisting grid-like, tree-like, and irregular fabrics generating varied topological vulnerabilities [9]. This heterogeneity implies that Braess-prone motifs may occur locally even when the overall network appears robust, which motivates diagnostic approaches that combine equilibrium reasoning with spatial network analysis [35].

Classical urban design theory since Jacobs has celebrated permeability and fine-grained connectivity as foundations of urban vitality and accessibility [7]. In transportation planning, similar logic informed postwar investments in arterial grids and urban expressways intended to increase vehicular throughput [1, 4]. However, the Braess' paradox suggests that under selfish routing, expanding connectivity can sometimes reduce effective accessibility by enabling inefficient equilibrium flow concentration [45, 48].

Empirical evidence on induced demand is consistent with a broad equilibrium sensibility in planning debates, but it should be distinguished from Braess-type effects in mechanism and interpretation [13]. Induced demand concerns demand response and long-run adjustment when capacity expands, whereas the Braess' paradox is defined under a fixed demand level and arises through re-routing and flow redistribution at equilibrium. The two can coexist in real systems, but they are analytically distinct: induced demand changes the amount of travel, while Braess changes the routing pattern of a given amount of travel [4]. Conversely, cases of "disappearing traffic" indicate that removing or downgrading road links can, in some contexts, improve network-wide performance [18]. Taken together, these empirical regularities suggest that behavioural feedback mechanisms matter in real urban systems, lending plausibility to paradox-based cautions about naive connectivity expansion, while also indicating the need to separate short-run equilibrium redistribution from longer-run demand adaptation in literature synthesis [15].

Urban design practice has increasingly explored this lesson through strategies such as modal filtering, superblocks, low-traffic neighborhoods, and transit-priority corridors [19, 25]. Such interventions reduce vehicular connectivity in selected areas to reshape route-choice patterns and discourage inefficient cut-through movements. From the perspective of Braess theory, these measures can be interpreted as constraining paradox-prone subgraphs embedded in the urban street network [32]. At the same time, restricting vehicular permeability can impose access costs on some users or trips, so the design problem is not equivalent to minimizing travel time, but to balancing efficiency with safety, public-space

quality, and distributional objectives [24].

Urban network resilience research investigates how street systems respond to disruptions, closures, or structural interventions [34, 35]. Within this literature, vulnerability often refers to configurations where small perturbations produce disproportionately large performance changes. The Braess' paradox can be understood as a specific case of such vulnerability, where adding a link triggers a reorganization of equilibrium flows that reduces overall efficiency [31].

Network-science methods have been applied to identify critical connectors, redundant corridors, and motif distributions in real urban networks [12]. These analyses provide empirical tools for detecting diamond-like and ladder-like structures that may be susceptible to Braess-type effects under equilibrium routing [62]. Thus, paradox analysis and resilience diagnostics converge on a shared concern: identifying connectivity patterns that generate systemic risk under adaptive flow redistribution [32, 35].

In urban design, this insight supports a focus on selective connectivity: networks may remain highly connected for walking and cycling while restricting vehicular through-movement in sensitive areas [20]. Such differentiated connectivity modifies the feasible strategy space of travelers and can reduce the likelihood of harmful equilibrium routing patterns. In this sense, Braess-type reasoning can be read as one formal justification for design doctrines that balance permeability and control rather than maximizing raw vehicular connectivity [9].

Sustainable mobility paradigms emphasize reducing car dependence, promoting modal shift, and reshaping street space allocation toward public and active transport [23, 25]. Within this paradigm, reducing road capacity, reallocating carriageways, and constraining vehicular through-traffic are increasingly treated as legitimate design interventions [19]. Braess-type paradoxes provide theoretical support for the idea that capacity-reduction strategies do not inevitably degrade network performance and may, under certain equilibrium conditions, improve it [31, 66].

Empirical congestion research similarly indicates that expanding supply rarely resolves congestion because induced demand counteracts capacity increases [15]. Braess theory formalizes a related message at the level of equilibrium routing: transport systems are adaptive, and supply-side expansions should be evaluated through behavioural feedback rather than static capacity logic [4]. By embedding equilibrium thinking into urban design, planners can better anticipate unintended consequences of connectivity interventions and design for adaptive system performance.

While the Braess' paradox was developed in abstract directed networks, contemporary computational tools allow mapping of these ideas onto city street graphs [12]. By extracting real-world networks from open geographic data and applying equilibrium assignment models, it becomes possible to identify zones where new links or capacity upgrades may trigger adverse equilibrium shifts [54]. This integration supports the development of di-

agnostic instruments for equilibrium-aware urban network design [35].

Such diagnostics can transform the paradox from a general warning into a risk-assessment frame for connectivity interventions. They also complement spatial-morphological approaches by adding behavioural feedback as an evaluation dimension [8, 9]. As a result, one plausible direction for urban design practice is to rely more explicitly on equilibrium-based computational testing to compare alternative connectivity layouts before implementation [5].

Within this evolving interdisciplinary field, the present dissertation positions the Braess' paradox as a conceptual and computational bridge between transportation equilibrium theory and urban design decision-making. By constructing structured network families (diamond, ladder, modular bypasses) and systematically mapping equilibrium response under congestion functions and behavioural dispersion, the study develops transferable knowledge about how connectivity interventions propagate through adaptive routing behavior [62]. To make this positioning concrete for the literature review, the study is organized around outputs that are comparable across model variants: regime maps that classify when added links improve or degrade equilibrium performance, topology-sensitive scans that identify where paradox boundaries occur in parameter space, and cross-model comparisons that isolate behavioural dispersion effects under DUE/SUE [28, 58].

In contrast to much existing work that either stays abstract (graph-theoretic or inefficiency bounds) or stays case-specific (single networks and calibrations), the contribution here is to provide a comparable experimental scaffold across

(i) topology families,

(ii) cost curvature assumptions, and

(iii) DUE/SUE behaviour,

so that design-oriented interpretation can rest on regime comparisons rather than isolated examples [31]. In doing so, the research supports an emerging notion of equilibrium-aware urban design, where spatial connectivity is evaluated not only geometrically but through predicted behavioural and systemic performance [25]. This choice is methodologically necessary because the same spatial intervention can be benign under one behavioural assumption and harmful under another, and a consistent scaffold allows these differences to be attributed to structure and behavior rather than to incompatible setups.

The following chapters operationalize this bridge by developing computational experiments on structured network templates, deriving paradox regime maps, and discussing how such results can inform evidence-based urban connectivity design. In this way, the Braess' paradox is treated less as a singular anomaly and more as a structured lens for discussing when urban connectivity interventions are more likely to be beneficial, neutral, or potentially counterproductive under equilibrium adaptation.

2.7. Critical assessment and research gaps

The preceding sections indicate that the Braess' paradox has expanded from a single counterintuitive construction into a research area spanning traffic equilibrium theory, network topology, computational methods, and increasingly explicit connections to urban design interpretation. At the same time, the literature remains somewhat fragmented across these strands, and several limitations continue to shape what can be concluded and what can be transferred into planning and design reasoning. This section offers a cautious critical assessment of the state of knowledge and identifies the specific research gaps that motivate the present dissertation.

Early theoretical work established existence, uniqueness, and prevalence conditions for Braess-type effects under deterministic user equilibrium and separable monotone cost functions [30, 55]. Subsequent advances extended these ideas to broader congestion games and provided bounds on paradox severity [31, 51, 68]. However, many analytical insights remain tied either to small stylized networks or to abstract bounds that are not straightforward to operationalize for intermediate-scale network families [31, 68]. Here, intermediate-scale refers to networks larger than minimal motifs (such as a single diamond) but still structured enough to be treated as families with repeated substructures and design meaning, for instance ladder-like corridor systems, repeated block-scale bypass opportunities, or multi-stage connectors that reflect urban block and corridor compositions rather than a full metropolitan network [9, 62]. Closed-form equilibrium characterizations tend to become difficult once realistic cost functions, asymmetric parameters, or multi-stage topologies are introduced [47, 54].

As a consequence, the theoretical landscape is often split between

(i) elegant but highly idealized analytical proofs, and

(ii) numerical demonstrations whose generality is not always clear [30, 54].

A systematic bridge between symbolic tractability and scalable computational exploration is still relatively underdeveloped. This limits the ability to convert mathematical results into design-relevant statements about when connectivity interventions are more likely to be benign versus risky under equilibrium response. In this thesis, risky is defined in a measurable way as the probability or frequency of a travel-time deterioration event under a controlled intervention, operationalized as a positive change in equilibrium total system travel time or average equilibrium travel time after adding a link or increasing capacity within a specified parameter range [45, 53].

Topology has been recognized as a key determinant of paradox occurrence, particularly through series–parallel impossibility results and motif-based interpretations [30–32]. Ladder networks and modular multi-stage constructions have been proposed as controlled extensions [62]. Yet, existing studies often examine only a small number of configurations or rely on a limited set of illustrative variants, which makes it difficult to infer how paradox risk changes as topology is scaled or recomposed [32, 62].

There is therefore limited availability of unified frameworks that systematically enumerate, parameterize, and scan whole families of structured variants [54]. As a result, the literature contains relatively few comprehensive "paradox maps" that relate network size, bypass placement, and overlap structure to equilibrium performance regimes [35, 62]. This absence of structured topological exploration makes it harder to assess how paradox risk scales from minimal motifs to the kinds of meso-scale patterns that resemble repeated urban blocks, corridors, and parallel-route systems. Here, meso-scale denotes a level between a single motif and a citywide network, where a repeated pattern can be interpreted both as a graph family and as a plausible urban composition (for example, a sequence of parallel streets connected by periodic cross-links, or a corridor with repeated bypass opportunities) [9, 11].

Evidence suggests that Braess-like deterioration can persist under stochastic perception and behavioural heterogeneity [28, 58]. Nevertheless, deterministic and stochastic equilibria are frequently studied in partially separated literatures, often with different modeling choices, cost specifications, and computational settings. This means that comparisons between DUE and SUE can sometimes confound behavioural differences with differences in network representation, parameterization, or solver conventions. Typical confounds include changing the cost function family between the DUE and SUE experiments, altering demand normalization or capacity scaling, using different stopping criteria and gap measures, or switching between link-based and path-based loading conventions, each of which can shift the apparent presence or severity of paradox regimes even if behaviour is held constant [4, 5].

A coherent comparative scaffold that evaluates identical network families under both DUE and SUE while systematically varying behavioural dispersion remains relatively rare [28, 54]. This gap matters for interpretation: without controlled comparison, it is difficult to assess whether paradox outcomes are mainly artifacts of perfect rationality or whether they remain plausible structural risks under realistic behavioural uncertainty. For design and planning purposes, this distinction affects whether paradox reasoning is best treated as a worst-case caution or as a robust diagnostic [43].

Many foundational studies employ linear or affine cost functions for analytical convenience [21, 30]. Some numerical investigations adopt convex or polynomial forms, and planning practice often relies on capacity-normalized functions such as BPR and related families [61]. However, systematic regime mapping under empirically grounded nonlinear cost functions remains relatively limited when compared to the volume of work under affine assumptions [66].

Because BPR-type functions introduce nonlinear sensitivity to flow–capacity ratios, equilibrium re-routing responses may deviate from linear-theory intuition, particularly near capacity. Without systematic mapping across cost curvature and calibration ranges, it remains uncertain how far theoretical paradox conditions derived under affine costs transfer to practical planning representations [54, 61]. This motivates an explicit emphasis on realism–tractability tradeoffs and on computational scanning under standardized nonlin-

ear cost specifications [5].

Computational studies widely employ MSA and related numerical solvers to detect paradoxical outcomes across parameter sweeps [4, 54]. Separately, symbolic analyses have derived explicit equilibrium conditions for small networks and have clarified critical thresholds in analytically tractable settings [47]. Yet, hybrid numerical–symbolic workflows that use symbolic reasoning to interpret large-scale numerical scans remain relatively uncommon.

Without such integration, computational studies can become collections of case-specific outcomes whose structural explanation is hard to communicate, while analytical studies remain constrained to small motifs. A methodological gap therefore persists in linking scalable computation to interpretable equilibrium structure in a way that supports regime-level understanding (for example, phase-diagram reasoning) rather than isolated examples. For literature review purposes, this gap is important because it explains why many papers either report broad computational findings without a transparent mechanism, or provide sharp mechanisms that do not scale beyond a few links, leaving limited guidance on how to generalize structure-to-outcome explanations across families [47, 54].

Recent work has begun to connect Braess-type reasoning to urban street-network morphology and sustainable mobility paradigms [9, 11]. However, many urban design discussions remain conceptual or normative and rarely include explicit equilibrium modeling of structured street-network interventions. In particular, there is limited work translating paradox regime analysis into operational diagnostics that can inform the evaluation of connectivity interventions in design practice [19].

A further reason for this weak translation is that urban design interventions are typically evaluated under multiple criteria, including safety, public-space quality, liveability, emissions, and equity, rather than travel time alone. As a result, even when equilibrium travel-time deterioration is detected, the design implication is not simply to avoid connectivity, but to articulate trade-offs and to identify conditions under which selective connectivity can improve some objectives while degrading others. This multi-criteria context motivates a diagnostic framing where paradox-style results help identify where vehicular permeability is likely to induce inefficient through-movement or bottleneck concentration, which can then be weighed against non-traffic objectives and mitigation strategies [20, 25]. Equilibrium-aware spatial design frameworks therefore remain at an early stage, with only partial integration between

(i) graph-based urban morphology descriptions and

(ii) congestion equilibrium computation under behavioral uncertainty.

This motivates research that uses structured network templates as a bridge: sufficiently stylized to remain interpretable, yet sufficiently general to suggest transferable diagnostics about connectivity, hierarchy, and selective permeability [5, 9].

Synthesizing the above assessment, five interrelated gaps can be stated as follows:

(i) Analytical–computational gap: limited availability of hybrid symbolic–numerical frameworks that connect scalable equilibrium computation with interpretable analytical structure.

(ii) Topological exploration gap: lack of systematic scanning of structured network families that bridge minimal motifs and larger modular networks [62].

(iii) Behavioral robustness gap: relatively few controlled comparisons of DUE and SUE across identical network and cost settings while varying dispersion [28].

(iv) Realistic congestion gap: insufficient mapping of paradox regimes under empirically grounded nonlinear cost functions (such as BPR-type forms) across calibration ranges [61, 66].

(v) Design translation gap: limited operational integration of equilibrium paradox analysis into urban network and street-design evaluation [19, 25].

The present dissertation is designed to address these gaps through a unified methodological framework. First, structured network families (diamond, ladder, and generalized bypass networks) are parameterized to enable systematic topological enumeration rather than isolated case selection [62]. Second, deterministic and stochastic equilibria are computed under both linear and BPR-type cost functions using consistent MSA-based solvers, supporting controlled comparisons across behavioural regimes and congestion specifications [4, 5]. Third, symbolic equilibrium derivations are developed for representative configurations to interpret numerical regime maps and to clarify which structural mechanisms explain observed phase boundaries [47]. Finally, results are reframed in terms of connectivity-intervention diagnostics that can be discussed in urban network and street-design terms (for example, when adding a connector is more likely to concentrate flow onto bottlenecks versus when selective filtering can reduce systemic risk) [9, 19].

By integrating topology, behavior, cost realism, and hybrid computation, the dissertation aims to move Braess' paradox research from isolated counterexamples toward a more transferable analytical–computational toolkit. This positioning establishes the rationale for the methodological development in Chapter 3 and the systematic experiments presented in Chapters 4 and 5, and it also provides a clearer basis for the design-oriented interpretation advanced in Chapter 6 [35].

3 | Equilibrium Assignment of Demand

3.1. Mathematical Formulation of Deterministic User Equilibrium (DUE)

The concept of Deterministic User Equilibrium(DUE) constitutes the theoretical foundation of classical traffic assignment theory. In this framework, travelers are assumed to have perfect information about network conditions and to behave in a fully rational manner, choosing routes that minimize their individual travel times. The DUE formulation adopted in this thesis is grounded in Wardrop's first principle and represents a benchmark equilibrium concept for the analysis of congestion effects and network performance[61, 70].

Let the transportation network be represented as a directed graph

$$G = (N, A) \tag{3.1}$$

where N denotes the set of nodes and A the set of directed links. A set W of origin–destination (OD) pairs (r, s) is defined, each associated with a fixed and inelastic travel demand $Q_{rs} > 0$. In this graph representation, nodes correspond to decision points (e.g., intersections, ramps, or terminals), while directed links represent movement possibilities with an associated direction of travel. The directed structure is essential for modeling asymmetric travel conditions, such as one-way streets or direction-dependent capacities. Throughout the thesis, a single network realization is assumed to be fixed during an assignment run, while congestion effects enter through flow-dependent link cost functions introduced below.

The OD set W enumerates all travel markets considered in the assignment problem. Each pair $(r, s) \in W$ is associated with an inelastic demand level Q_{rs}, meaning that total demand is treated as given and does not respond endogenously to changes in travel time within the equilibrium computation. This assumption is consistent with the focus of the thesis on route choice and congestion redistribution under fixed demand.

For each OD pair (r, s), let $\mathcal{P}_{rs}$ denote the set of all feasible paths connecting origin r to

destination s. Path flows are denoted by

$$f_p \geq 0, \quad \forall p \in \mathcal{P}_{rs}, \ \forall (r,s) \in W \tag{3.2}$$

and must satisfy the flow conservation constraints

$$\sum_{p \in \mathcal{P}_{rs}} f_p = Q_{rs}, \quad \forall (r,s) \in W \tag{3.3}$$

The non-negativity constraint ensures that path flows represent physically meaningful traffic volumes. Together with the conservation constraints in (3.3), it defines the feasible set of path-flow vectors for each OD pair. In other words, equilibrium is sought over the simplex of path splits that exactly exhaust the OD demand without allowing negative flow allocations.

Equation (3.3) guarantees that all OD demand is fully assigned to the feasible path set. This condition is also crucial for the definition of aggregate performance indicators such as average travel time, because it ensures that the total assigned flow equals the total demand in the denominator of (3.8).

The relationship between path flows and link flows is expressed through the path–link incidence indicator δ_{ap}, which equals 1 if link $a \in A$ belongs to path p, and 0 otherwise. The flow on link a is then given by

$$x_a = \sum_{(r,s) \in W} \sum_{p \in \mathcal{P}_{rs}} \delta_{ap} f_p \tag{3.4}$$

The indicator δ_{ap} defines the structural mapping from path flows to link flows. Stacking δ_{ap} over all paths and links yields the path–link incidence matrix, which is the core data structure used in the numerical implementation: given a row vector of path flows, link flows are obtained by a single matrix multiplication. This mapping makes it possible to compute congestion effects at the link level while maintaining route choice behavior at the path level.

This representation allows for a clear correspondence between individual path choices and congestion effects occurring at the link level.

Each link $a \in A$ is associated with a travel time function $t_a(x_a)$, which is assumed to be continuous and non-decreasing with respect to the link flow x_a. These assumptions reflect congestion phenomena and ensure the existence of an equilibrium solution.

Continuity and monotonicity are standard regularity assumptions in traffic assignment because they exclude pathological behavior (e.g., decreasing travel time with increasing flow) and support existence results for equilibrium. They also underpin the convexity of the equivalent Beckmann formulation introduced later, which provides both theoretical insight and a foundation for numerical solution techniques.

Given a vector of link flows $\mathbf{x} = \{x_a\}$, the travel time of a path p is defined as the sum of the travel times of the links composing that path:

$$C_p(\mathbf{x}) = \sum_{a \in p} t_a(x_a) \tag{3.5}$$

Within the deterministic framework, all users are assumed to perceive these costs perfectly and identically, with no randomness or perception errors affecting route choice.

Wardrop's first principle provides the behavioral definition of deterministic user equilibrium and can be stated as follows: for each OD pair, the travel times on all routes actually used are equal and less than or equal to those on any unused route.

Formally, a path flow vector $\mathbf{f}^*$ satisfies the DUE conditions if, for every OD pair (r, s),

$$\begin{cases} f_p^* > 0 \;\Rightarrow\; C_p(\mathbf{x}^*) = \min\limits_{q \in \mathcal{P}_{rs}} C_q(\mathbf{x}^*), \\ f_p^* = 0 \;\Rightarrow\; C_p(\mathbf{x}^*) \geq \min\limits_{q \in \mathcal{P}_{rs}} C_q(\mathbf{x}^*), \end{cases} \tag{3.6}$$

Where, $\mathbf{f}^*$ denotes an equilibrium path-flow pattern and $\mathbf{x}^*$ is the corresponding equilibrium link-flow pattern obtained from (3.4). The quantity $\min\limits_{q \in \mathcal{P}_{rs}} C_q(\mathbf{x}^*)$ represents the minimum travel cost available between OD pair (r, s) under the prevailing congestion conditions. The Wardrop conditions therefore state that all used paths for a given OD pair share the same minimum cost, while any unused path cannot offer a strictly lower cost at equilibrium[3].

This condition implies that no traveler can unilaterally reduce their travel time by switching to an alternative path, given the prevailing network conditions.

Although Wardrop's principle is defined at the path level, equilibrium outcomes are often evaluated using aggregate performance indicators. In this thesis, particular emphasis is placed on the average travel time (ATT).

Under equilibrium assignment, all origin–destination demand is fully distributed over the feasible path set. Hence, the path flows satisfy the flow conservation condition

$$\sum_{p \in \mathcal{P}_{rs}} f_p^* = Q_{rs}, \qquad \forall (r, s) \in W. \tag{3.7}$$

Given this condition, the average travel time is defined as the demand-weighted mean travel cost:

$$\mathrm{ATT} = \frac{\sum_{(r,s) \in W} \sum_{p \in \mathcal{P}_{rs}} f_p^* C_p(\mathbf{x}^*)}{\sum_{(r,s) \in W} Q_{rs}} \tag{3.8}$$

The definition in (3.8) can be interpreted as a demand-weighted average over all users in the network. The numerator is the total experienced travel time aggregated over all OD pairs and all paths, which is often referred to as the total system travel time (TSTT). Dividing by the total demand yields a single scalar indicator that is comparable across network configurations, demand levels, and equilibrium assumptions.

In the numerical experiments of Chapters 4–6, ATT is computed in a way that is consistent with (3.8): the total travel time is obtained by summing path flows multiplied by their associated equilibrium path costs, and the result is normalized by total demand. This normalization ensures that ATT remains meaningful even when demand is scanned across a wide range, allowing paradoxical changes to be interpreted as changes in average user experience rather than changes in the scale of the system.

The ATT represents the mean travel time experienced by users in the network and serves as a key indicator for comparing different network configurations. It is especially relevant in the analysis of Braess-type paradoxes, where changes in network topology may lead to counterintuitive variations in overall performance.

It is important to note that a DUE does not minimize the average travel time. Instead, it emerges from decentralized, self-interested route choices rather than from a system-optimal objective.

Under the assumptions of separable and monotone link cost functions, the DUE problem admits an equivalent mathematical programming formulation. Specifically, the equilibrium link flow pattern $\mathbf{x}^*$ solves the following convex optimization problem:

$$\min_{x} \sum_{a \in A} \int_0^{x_a} t_a(u)\, du \tag{3.9}$$

In (3.9), the integral $\int_0^{x_a} t_a(u)\, du$ accumulates marginal travel time over increasing flow levels on each link, where u is a dummy variable of integration. The objective is expressed in terms of link flows, but feasibility is enforced through the existence of a path-flow decomposition satisfying (3.3)–(3.4). Under the stated assumptions, the optimality conditions of this convex program are equivalent to Wardrop's equilibrium conditions, providing a rigorous bridge between behavioral equilibrium and a variational principle[54, 56].

Although the present thesis focuses on fixed-point solution procedures, the Beckmann formulation provides a useful theoretical reference for understanding why equilibrium exists and why monotonicity assumptions are important for numerical stability.

subject to the feasibility constraints imposed by flow conservation and non-negativity of path flows.

This formulation establishes a direct connection between the behavioral equilibrium concept and numerical solution methods commonly employed in traffic assignment.

3.2. Stochastic User Equilibrium and Probabilistic Route Choice

The deterministic user equilibrium framework relies on the assumption that travelers possess perfect information and perceive travel costs identically. In practice, however, route choice behavior is subject to perception errors, incomplete information, and unobserved heterogeneity across individuals. These considerations motivate the introduction of Stochastic User Equilibrium (SUE), in which route choice is modeled probabilistically rather than deterministically [4, 5, 26]. Following the Logit-based SUE tradition in the traffic assignment literature [4, 26], the probabilistic choice rule in (3.11) and the consistency condition in (3.12) provide the mathematical basis of the formulation adopted below.

Let the perceived utility associated with path $p \in \mathcal{P}_{rs}$ for an OD pair (r, s) be defined as

$$U_p = -C_p(\mathbf{x}) + \varepsilon_p \tag{3.10}$$

where $C_p(\mathbf{x})$ is the systematic component of travel cost and ε_p is a random error term capturing perception errors and unobserved behavioral factors. The negative sign reflects the assumption that travelers prefer lower travel costs.

If the random terms ε_p are assumed to be independently and identically distributed according to a Gumbel distribution, the resulting route choice probabilities take the well-known Logit form:

$$P_p = \frac{\exp\left(-\dfrac{C_p(\mathbf{x})}{\theta}\right)}{\displaystyle\sum_{q \in \mathcal{P}_{rs}} \exp\left(-\dfrac{C_q(\mathbf{x})}{\theta}\right)}, \quad \forall p \in \mathcal{P}_{rs} \tag{3.11}$$

The parameter $\theta > 0$ is the dispersion (or sensitivity) parameter and plays a central role in determining the degree of randomness in route choice behavior.

Although the Logit probability expression adopted here is formally consistent with the classical stochastic user equilibrium model presented in the literature, its role in the present study differs from the purely theoretical formulation introduced in Chapter 2. In the literature review, the dispersion parameter (denoted by θ_1) is used only to describe the general behavioral sensitivity of travelers in a conceptual random-utility framework. By contrast, the parameter θ employed in the subsequent computational procedures is introduced from an implementation perspective and is treated as a calibrated numerical control variable.

This distinction motivates a slight reformulation of notation and interpretation. The theoretical model emphasizes behavioral interpretation, whereas the implemented formulation emphasizes numerical stability, parameter scanning, and reproducibility in simulation. In particular, expressing the Logit model in the temperature form adopted here facilitates

direct control of randomness, smoother convergence of the Method of Successive Averages, and systematic comparison with the deterministic user equilibrium limit ($\theta \to 0$). These advantages are especially important for the large-scale parametric experiments conducted in Chapters 4–6.

Importantly, this modification does not alter the underlying equilibrium concept: both formulations are mathematically equivalent representations of the same Logit-based stochastic user equilibrium. The change therefore serves purely computational and methodological clarity rather than introducing a different behavioral model.

From an implementation standpoint, the temperature form $P_p \propto \exp(-C_p/\theta)$ provides a direct and interpretable scale linking cost differences to probabilistic choice: for a fixed cost gap ΔC, the ratio of probabilities satisfies $P_i/P_j = \exp(-\Delta C/\theta)$. This makes θ a convenient knob for controlling randomness in a consistent manner across different demand levels and cost-function specifications. In large parameter sweeps, such a formulation also supports reproducibility, because the same numerical value of θ has the same qualitative effect on choice dispersion.

In addition, the adopted expression is numerically stable when combined with standard normalization practices in computation. In the code implementation, path utilities are shifted by subtracting the maximum value before exponentiation (i.e., a "log-sum-exp" stabilization), which prevents underflow/overflow when costs are large or when θ is small. This stability is essential for obtaining robust SUE solutions across the wide range of scenarios investigated in Chapters 4–6.

Under the SUE concept, equilibrium is achieved when the path flows implied by probabilistic route choice are consistent with the travel costs generated by those same flows. For each OD pair (r, s), the equilibrium path flows satisfy

$$f_p = Q_{rs}\, P_p, \quad \forall p \in \mathcal{P}_{rs} \tag{3.12}$$

Equation (3.12) shows that SUE can be viewed as a fixed-point consistency condition: costs determine choice probabilities, probabilities determine path flows, and path flows determine link flows and thus costs again. An equilibrium is reached when this circular dependence closes consistently, meaning that the realized flows reproduce the same costs that generated them through the Logit choice model.

Link flows are then obtained through the path–link incidence relationship, and travel times are updated accordingly. An SUE is thus defined as a fixed point in which perceived costs, choice probabilities, and resulting flows are mutually consistent.

Unlike the deterministic case, all feasible paths generally carry positive flow in SUE, although paths with higher costs receive exponentially smaller shares of demand.

The parameter θ governs the degree of sensitivity of travelers to differences in travel cost and has a clear behavioral interpretation in the temperature-based Logit formulation

adopted in this thesis.

When $\theta \to 0$, route choice becomes increasingly concentrated on the minimum-cost paths, and stochastic user equilibrium converges to deterministic user equilibrium. Formally,

$$\theta \to 0 \;\Rightarrow\; P_p \to \begin{cases} 1, & \text{if } p \in \arg\min\limits_{q \in \mathcal{P}_{rs}} C_q(\mathbf{x}), \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

where $\arg\min\limits_{q \in \mathcal{P}_{rs}} C_q(\mathbf{x})$ is the set of minimizers, such as the set of least-cost paths (possibly non-unique under ties).

When multiple paths tie for the minimum cost, the limiting statement should be interpreted set-wise: probability mass concentrates on the set of minimum-cost paths, while the allocation among tied minimizers is not uniquely determined by the limit alone. In practice, numerical procedures may distribute flow across tied paths depending on initialization and tolerance thresholds, which is one reason why SUE computations with small but positive θ can be advantageous for stability.

In this limit, even small cost differences are strongly amplified, and travelers behave as perfectly rational agents selecting only minimum-cost routes. Therefore, the SUE converges to the DUE solution as $\theta \to 0$.

Conversely, when $\theta \to \infty$, cost differences are attenuated and route choice becomes increasingly random. In the limit,

$$\theta \to \infty \;\Rightarrow\; P_p \to \frac{1}{|\mathcal{P}_{rs}|} \tag{3.14}$$

In this limit, travelers are effectively insensitive to travel cost differences, and route choice approaches a uniform distribution over all feasible paths. This corresponds to a context of extreme uncertainty, lack of information, or highly heterogeneous preferences.

It is important to emphasize that this convergence should be interpreted behaviorally rather than numerically. The SUE is not merely a smoothed approximation of DUE, but represents a distinct equilibrium concept rooted in probabilistic decision-making.

As in the deterministic case, the performance of the network under stochastic equilibrium is evaluated using the average travel time:

$$ATT = \frac{1}{\sum\limits_{(r,s) \in W} Q_{rs}} \sum_{(r,s) \in W} \sum_{p \in \mathcal{P}_{rs}} f_p \, C_p(\mathbf{x}) \tag{3.15}$$

The ATT definition in (3.15) uses the same demand-weighted structure as in the deterministic case, enabling direct comparison between DUE and SUE under identical network configurations and demand levels. Differences in ATT therefore reflect not a change in

the performance metric, but the behavioral mechanism by which stochastic choice spreads demand across multiple alternatives, including suboptimal paths, thereby altering congestion patterns and aggregate outcomes.

However, under SUE, ATT reflects not only congestion effects but also the distribution of flows across suboptimal paths induced by probabilistic choice. As a result, SUE outcomes may exhibit systematically different performance characteristics compared to DUE, even under identical network conditions.

This distinction plays a crucial role in the analysis of paradoxical phenomena, as the introduction of stochastic behavior can attenuate, amplify, or even eliminate paradox effects observed under deterministic assumptions.

The distinction between deterministic and stochastic equilibrium extends beyond mathematical formulation and reflects fundamentally different behavioral assumptions.

In DUE, equilibrium arises from perfectly rational users selecting minimum-cost paths, leading to sharp switching behavior and exact cost equalization across used routes. In contrast, SUE allows for persistent use of higher-cost paths, reflecting bounded rationality and imperfect information.

From a methodological perspective, SUE provides a more behaviorally realistic framework, while DUE serves as a limiting benchmark. Both concepts are therefore retained in this thesis, enabling a systematic comparison of equilibrium outcomes and their implications for network performance and paradox emergence. A broader discussion of DUE and SUE scenarios is provided in Section 2.3.

3.3. Numerical Solution Methods

The equilibrium conditions defined in the previous sections generally do not admit closed-form solutions, except in highly simplified network configurations. As a result, both deterministic and stochastic user equilibria must be obtained through numerical procedures. In this thesis, equilibrium solutions are computed using iterative fixed-point methods, with particular emphasis on the Method of Successive Averages (MSA)[4, 54].

Rather than being treated as a mere algorithmic choice, the numerical solution procedure is regarded as an integral component of the methodological framework, as it determines how the theoretical equilibrium concepts are approximated in practice.

Both DUE and SUE can be formulated as fixed-point problems. Given a vector of link flows $\mathbf{x}^{(k)}$ at iteration k, link travel times and path costs are computed, which in turn define a new set of path flows through either deterministic or probabilistic route choice rules. These updated flows induce a new vector of link flows $\hat{\mathbf{x}}^{(k)}$.

An equilibrium is reached when the flow update mapping satisfies

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} = \hat{\mathbf{x}}^{(k)} \tag{3.16}$$

The fixed-point notation highlights that equilibrium computation is fundamentally an iterative consistency problem rather than a direct closed-form evaluation. The mapping $\mathbf{T}(\cdot)$ represents the composite operation of

(i) computing link costs from current flows,

(ii) computing path costs,

(iii) generating an auxiliary assignment based on the equilibrium concept (all-or-nothing for DUE or Logit probabilities for SUE), and

(iv) translating the resulting path flows back to link flows.

At equilibrium, applying this mapping leaves the flow vector unchanged.

The numerical challenge lies in iteratively approaching this fixed point while maintaining stability and convergence, particularly in congested networks or in the presence of multiple competing paths.

The Method of Successive Averages is a widely adopted iterative scheme for solving traffic assignment problems[4]. Starting from an initial feasible flow pattern $\mathbf{x}^{(0)}$, the method updates flows according to the convex combination:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \left(\hat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\right) \tag{3.17}$$

The MSA update in (3.17) forms a convex combination between the current iterate and an auxiliary flow pattern implied by the current costs. The step size α_k controls the relaxation intensity: large values lead to faster but potentially unstable adjustments, while small values damp oscillations but may slow convergence. The diminishing sequence $\alpha_k = 1/(k+1)$ is widely used because it gradually stabilizes the iterates and has convergence guarantees under standard regularity conditions.

where $\hat{\mathbf{x}}^{(k)}$ is the auxiliary flow obtained from an all-or-nothing (for DUE) or probabilistic (for SUE) assignment, and α_k is the step size at iteration k.

A standard choice is

$$\alpha_k = \frac{1}{k+1} \tag{3.18}$$

which guarantees diminishing step sizes and ensures convergence under suitable regularity conditions[4, 54].

Importantly, MSA should be interpreted not as an optimization algorithm, but as a relaxation-based procedure for approximating the fixed point defined by the equilibrium conditions.

The structure of the auxiliary flow $\hat{\mathbf{x}}^{(k)}$ differs fundamentally between DUE and SUE.

In the DUE case, the auxiliary solution is obtained by assigning all demand to minimum-cost paths, resulting in an all-or-nothing update. This leads to sharp adjustments in path flows and can induce oscillatory behavior when multiple paths have nearly equal costs.

In contrast, the SUE auxiliary solution distributes flows across all feasible paths according to probabilistic choice. As a result, the update mapping is smoother, and successive iterations tend to exhibit more stable behavior.

This distinction has important numerical implications: while MSA is theoretically applicable to both equilibrium concepts, its practical convergence properties differ significantly between DUE and SUE.

Conceptually, the key difference is the sensitivity of the auxiliary assignment to marginal cost differences. In DUE, an arbitrarily small cost advantage can shift the entire demand to a different path, producing discontinuous changes in the auxiliary solution. In SUE, the Logit rule transforms cost differences into gradual probability shifts, so the auxiliary solution changes smoothly as costs evolve, which typically yields more stable fixed-point iterations in practice.

introbreak

In practice, convergence of MSA is often slow, particularly for deterministic user equilibrium. In networks with symmetric or nearly symmetric paths—such as those considered in Braess-type configurations—small fluctuations in travel times may cause large redistributions of flow between alternative routes.

This phenomenon manifests as persistent oscillations in link flows and path costs, even after a large number of iterations. Although the diminishing step size eventually damps these oscillations, the rate of convergence can be prohibitively slow for large-scale parameter sweeps.

From a methodological standpoint, these convergence issues are not merely numerical inconveniences, but reflect the inherent sensitivity of deterministic equilibrium to marginal cost differences.

Such oscillations are therefore not merely numerical artifacts; they reflect the intrinsic discontinuity of deterministic choice when multiple near-equivalent alternatives exist. This observation motivates the inclusion of acceleration and stabilization techniques, such as the restart strategy introduced next, which aim to improve practical efficiency while preserving the equilibrium definition.

The Method of Successive Averages (MSA) may converge slowly in congested networks, especially when successive auxiliary loadings induce small but persistent oscillations. In the implementation used in this thesis, the MSA step size is

$$\alpha_\ell = \frac{1}{\ell},$$

where ℓ is a local iteration counter. A restart mechanism is therefore introduced to reset ℓ whenever stagnation is detected, while preserving the accumulated information in the

current iterate.

Let $\mathbf{x}^{(it)}$ denote the path-flow vector at global iteration it, and let

$$\mathbf{X}^{(it)} = \mathbf{x}^{(it)}\mathbf{B}$$

be the corresponding link-flow vector (with $\mathbf{B}$ the path–link incidence matrix). Stagnation is monitored through the normalized ℓ_1-change in link flows,

$$\mathrm{rel}(it) = \frac{\left\|\mathbf{X}^{(it+1)} - \mathbf{X}^{(it)}\right\|_1}{\max(Q, 1)}, \qquad \|\mathbf{v}\|_1 = \sum_e |v_e|, \tag{3.19}$$

where Q is the OD demand used only as a scale normalizer. The smallest value observed so far is stored as $\mathrm{rel}_{\min}$, and an iteration is deemed to yield a meaningful improvement only if

$$\mathrm{rel}(it) + \tau_{\mathrm{restart}} < \mathrm{rel}_{\min}.$$

If no such improvement is observed for K_{restart} consecutive global iterations, a restart is triggered.

When a restart occurs, the current auxiliary loading $\mathbf{y}$ (all-or-nothing for DUE, Logit-based probabilistic loading for SUE) is adopted as a warm start,

$$\mathbf{x} \leftarrow \mathbf{y}, \qquad \mathbf{X} \leftarrow \mathbf{Y} = \mathbf{y}\mathbf{B},$$

and the local counter is reset to $\ell = 0$ so that subsequent MSA steps regain larger step sizes. To avoid excessive intervention, the total number of restarts within one assignment run is capped at $R_{\max}$. In the present implementation, $K_{\mathrm{restart}} = 200$, $\tau_{\mathrm{restart}} = 10^{-6}$, and $R_{\max} = 5$. The restart mechanism does not alter the equilibrium definition; it is solely a numerical acceleration safeguard for difficult cases. The specific settings of the experiment can be found in Section 4.5.

Convergence is assessed using the same normalized ℓ_1-change in link flows as in (3.19). The algorithm declares convergence when

$$\mathrm{rel}(it) = \frac{\left\|\mathbf{X}^{(it+1)} - \mathbf{X}^{(it)}\right\|_1}{\max(Q, 1)} < \varepsilon, \tag{3.20}$$

where ε is the flow-change tolerance. This ℓ_1-based criterion provides a scale-free stopping rule that remains comparable across different demand levels.

The numerical solution procedure plays a crucial role in shaping observed equilibrium outcomes. Differences between DUE and SUE are reflected not only in theoretical definitions, but also in convergence behavior, stability, and sensitivity to initialization.

By explicitly addressing convergence issues and adopting restart strategies, the numerical framework employed in this thesis ensures that observed paradoxical effects are attributable to equilibrium structure rather than numerical artifacts. This distinction is

essential for the validity of the comparative analyses presented in the following sections.

3.4. Verification on the Braess Network

To validate the equilibrium formulations and numerical procedures introduced in the previous sections, a verification exercise is conducted on the classical Braess network. Rather than serving as a case study in its own right, this network is employed as a benchmark to assess whether the deterministic and stochastic equilibrium concepts are correctly reproduced by the proposed methodological framework.

The Braess network is particularly suitable for this purpose, as it combines analytical tractability with the capacity to generate counterintuitive outcomes under equilibrium conditions.

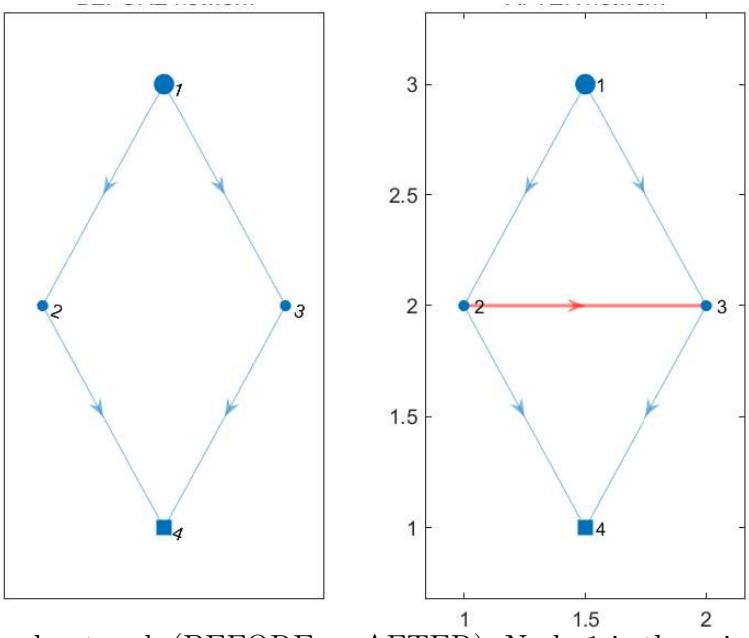


Figure 3.1: Diamond network (BEFORE vs AFTER). Node 1 is the origin, node 4 is the destination. The added connecting link is $u_5 : 2 \to 3$.

The network we consider in Figure 5.1 consists of a single origin–destination pair connected by two parallel routes, with an additional intermediate link introduced to form the Braess configuration. Travel demand is fixed and identical across all experiments, and link cost functions are specified so as to ensure monotonicity and consistency with the assumptions underlying both DUE and SUE.

Two network configurations are analyzed: a baseline network without the additional link, and an augmented network in which the link is added. For each configuration, equilibrium flows are computed under both deterministic and stochastic user equilibrium assumptions using the numerical methods described in Section 3.3.

The objective of this verification is not to explore parameter sensitivity, but to confirm

that the numerical solutions satisfy the defining properties of the respective equilibrium concepts.

The verification focuses on two complementary aspects. First, it checks behavioral consistency: the DUE solution should satisfy Wardrop indifference among used paths, while the SUE solution should satisfy probabilistic consistency between costs and flows through (3.11)–(3.12). Second, it checks performance reporting consistency: aggregate indicators such as ATT should be computed identically across configurations so that observed differences can be attributed to equilibrium structure rather than measurement choices.

Under deterministic user equilibrium, the computed solution exhibits strict compliance with Wardrop's first principle. In the augmented network, all paths carrying positive flow have equal travel times, while unused paths have travel times that are greater than or equal to this common value.

The equilibrium flow pattern reflects sharp route switching behavior: small differences in link travel times lead to complete reallocation of demand between alternative paths. This behavior is consistent with the assumption of perfectly rational users with identical cost perceptions.

Importantly, the average travel time in the augmented network is found to be higher than that of the baseline configuration, despite the increase in network capacity. This outcome confirms the presence of a Braess-type paradox under deterministic equilibrium assumptions and serves as a verification that the numerical framework correctly reproduces the theoretical phenomenon.

Operationally, compliance with Wardrop's first principle is assessed by verifying that all paths with positive equilibrium flow attain the minimum OD travel cost within a prescribed numerical tolerance, while any unused path has a cost no lower than this minimum. This provides a direct diagnostic that the computed fixed point corresponds to the intended deterministic equilibrium concept. The specific calculation results of DUE are reported in Figure 5.2 and Section A.1

When stochastic user equilibrium is considered, the resulting flow pattern differs markedly from the deterministic case. Demand is distributed across all feasible paths, including those with higher travel costs, reflecting the probabilistic nature of route choice.

In this setting, equilibrium is achieved not through exact cost equalization, but through consistency between perceived costs and choice probabilities. Paths with lower travel times attract a larger share of demand, yet no path is entirely excluded from use.

The resulting average travel time in the augmented network differs from that observed under DUE, both in magnitude and in sensitivity to network modification. Depending on the dispersion parameter, the paradoxical increase in average travel time may be attenuated or partially mitigated.

Varying θ allows the verification to interpolate continuously between deterministic-like behavior (small θ) and highly dispersed behavior (large θ). This sensitivity analysis

also serves as a numerical check: as θ decreases, the SUE solution should increasingly concentrate flow on the least-cost alternatives and approach the DUE pattern in both route usage and aggregate performance.

This behavior confirms that SUE does not merely smooth the deterministic solution numerically, but represents a fundamentally different equilibrium concept grounded in behavioral heterogeneity. The specific calculation results of DUE are reported in Figure 5.5 and Section A.3.

The comparison between deterministic and stochastic equilibria on the Braess network highlights the critical role of behavioral assumptions in shaping equilibrium outcomes.

Under DUE, the equilibrium is characterized by discontinuous adjustments and exact indifference among used paths, making the system highly sensitive to network changes. This sensitivity underlies the emergence of paradoxical effects when additional links are introduced.

Under SUE, by contrast, probabilistic route choice introduces inertia into the system. Users continue to utilize suboptimal paths, reducing the likelihood of extreme congestion concentration and moderating paradoxical outcomes.

From a methodological perspective, this verification exercise demonstrates that the proposed framework is capable of:

(i) reproducing classical deterministic equilibrium results;

(ii) capturing the behavioral richness of stochastic equilibria;

(iii) distinguishing structural paradox effects from numerical artifacts.

These properties are essential for the systematic analysis conducted in subsequent chapters, where network topology, demand levels, and cost functions are varied to investigate the conditions under which Braess-type paradoxes arise.

The results obtained on the Braess network confirm the internal consistency of the equilibrium formulations and numerical procedures adopted in this thesis. Both DUE and SUE solutions satisfy their respective defining conditions, and observed differences in network performance can be traced directly to differences in behavioral assumptions rather than implementation details.

As such, the Braess network serves as a methodological validation tool, providing confidence that the framework can be reliably extended to more complex network structures and parameter configurations in the following analyses.

4 | Methodology

This chapter operationalises the equilibrium concepts introduced in Chapter 3 by specifying a computational protocol for systematic, reproducible, and interpretable Braess-type experiments. Rather than relying on a single illustrative example, the chapter defines an experimental architecture that enables controlled comparisons across:

(i) alternative network structures, including scalable families;

(ii) multiple families of link cost functions;

(iii) both deterministic and stochastic equilibrium models implemented under consistent numerical conventions.

The framework further supports parameter scanning over demand and model coefficients, structured reporting of paradox indicators, and—where relevant—comparison with the symbolic conditions developed later in the thesis. Finally, the computational outputs are organised to support interpretation in terms of design-relevant network interventions.

The workflow is organised in progressively more general stages. It begins by reproducing the canonical Braess (diamond) network to validate the implementation against established results in the literature. The analysis is then extended by varying demand ranges and link cost specifications, and by introducing stochastic route choice through logit-based SUE in order to examine how behavioural dispersion affects the location, persistence, and boundaries of paradox regimes. The scope is subsequently expanded from the diamond benchmark to a ladder network family, allowing paradox mechanisms to be tested under repeated motifs and multiple interacting cycles.

This staged organisation provides a controlled experimental design as well as a narrative structure. At each stage, one modelling dimension is varied while the remaining components are held fixed, so that changes in paradox outcomes can be attributed more transparently to topology, cost-function curvature, or behavioural dispersion. Such control is particularly important for parameter scans and the synthesis of paradox maps, where uncontrolled variation can obscure the interpretation of boundary shifts.

Accordingly, this chapter fulfils two complementary functions. First, it documents the computational objects, parameterisations, and solver conventions that underpin the numerical results in subsequent chapters. Second, it clarifies the experimental comparisons that structure the remainder of the thesis, from benchmark reproduction to cost-function and behavioural generalisations, and finally to scalable network topologies. The following sections specify the network generators and cost-function configurations, and describe the

implementation choices—including reproducibility and reporting conventions—that make large-scale parameter exploration feasible.

4.1. Reproduction of the classic Braess' paradox

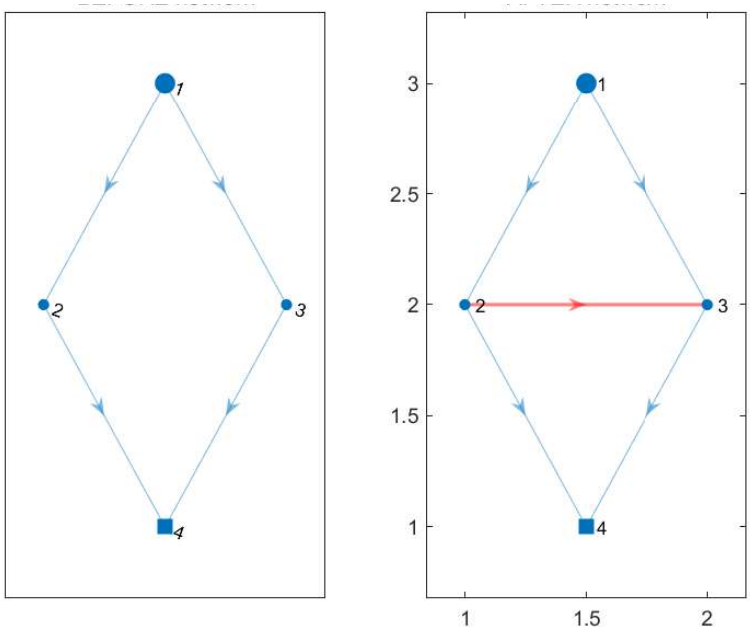


Figure 4.1: Diamond network (BEFORE vs AFTER). Node 1 is the origin, node 4 is the destination. The added connecting link is $2 \to 3$.

This section documents the exact computational configuration used to reproduce the classical Braess' paradox benchmark. The purpose is methodological: all modelling inputs (network, link ordering, cost functions, demand grid, solver settings, and paradox decision thresholds) are stated explicitly so that the results can be regenerated without relying on implementation-specific defaults.

All experiments in this thesis are built around a common set of computational objects that remain consistent across network types and equilibrium concepts. A network instance is defined by

(i) a directed link list (ordered node pairs),

(ii) a designated OD pair (r, s), and

(iii) a link ordering (legend) that fixes how link-flow vectors are interpreted.

For the classic diamond benchmark, the OD pair is $(r, s) = (1, 4)$ with intermediate nodes $\{2, 3\}$. The before network contains four directed links:

$$(1 \to 2),\ (2 \to 4),\ (1 \to 3),\ (3 \to 4),$$

and the after network adds the connecting link $(2 \to 3)$, yielding five directed links in total. To remove ambiguity in replication, the solver uses the following link legend in the after network:

$$a = 1 : (1 \to 2), \quad a = 2 : (2 \to 4), \quad a = 3 : (1 \to 3), \quad a = 4 : (3 \to 4), \quad a = 5 : (2 \to 3).$$

The link-flow vector $\mathbf{x} \in \mathbb{R}^{|A|}$ is always interpreted in this legend order.

For any given network state (before or after), the path set $\mathcal{P}_{rs}$ is defined as the set of all simple directed r–s paths (no repeated nodes), enumerated deterministically by a depth-first search. A safety cap $P_{\max} = 20000$ is imposed to prevent uncontrolled growth in larger networks; for the diamond benchmark this cap is non-binding (2 admissible paths in the before network and 3 in the after network, including 1–2–3–4).

Instead of introducing additional symbols, the implementation follows the incidence indicator δ_{ap} defined in Chapter 3. Given path flows f_p on $p \in \mathcal{P}_{rs}$, link flows are obtained as

$$x_a = \sum_{p \in \mathcal{P}_{rs}} \delta_{ap} \, f_p, \tag{4.1}$$

which is implemented as a matrix multiplication by stacking δ_{ap} over all links and paths.

To reproduce the standard benchmark, all five links follow linear travel-time functions of the form

$$t_a(x_a) = b_a + m_a x_a,$$

with parameters fixed as in Table 4.1. This benchmark corresponds to the classical linear specification used in the Braess literature and adopted in the thesis results chapter for calibration.

Table 4.1: Linear cost functions used for reproducing the classic Braess' paradox (reported in the solver's link legend order).

Link index a	Arc	Cost function $t_a(x_a)$	(b_a, m_a)
1	$1 \to 2$	$10x_1$	$(0, 10)$
2	$2 \to 4$	$50 + x_2$	$(50, 1)$
3	$1 \to 3$	$50 + x_3$	$(50, 1)$
4	$3 \to 4$	$10x_4$	$(0, 10)$
5	$2 \to 3$ (added)	$10 + x_5$	$(10, 1)$

The framework treats the cost family as a per-link attribute (rather than a global switch), so the same solver routines apply unchanged when later sections introduce nonlinear families (e.g. BPR) or heterogeneous mixtures.

The primary solver output for each demand value includes the equilibrium path-flow vector $\mathbf{f}^*$, the induced link-flow vector $\mathbf{x}^*$, link costs $\mathbf{t}(\mathbf{x}^*)$, and the resulting performance measures (total and average travel time). To support verification and interpretation, the implementation also records the before/after link-flow vectors under the same legend,

ensuring that paradox detection is accompanied by the equilibrium flow patterns that generate it.

This benchmark varies only the total OD demand because the classic paradox is demand-dependent and a demand scan provides a direct way to identify paradox intervals. Since there is a single OD pair (r, s), the scanned demand is written consistently with Chapter 3 as Q_{rs}, and we set $Q \equiv Q_{rs}$ for brevity. In the reproduction experiments, the demand grid is from 1 to 20 with step size 1. All other modelling inputs are held fixed: the node set, OD pair $(1, 4)$, the affine cost parameters in Table 4.1, and the numerical solver settings stated below.

Across the scan, the before–after comparison follows a strict controlled-comparison standard. OD pair and demand are identical between the two network states; link cost functions for links present in both networks are identical; and the only structural difference is the presence of the added link $(2 \to 3)$ in the after network. The cost function of the added link is also fixed (Table 4.1) in this reproduction step, so any performance difference can be attributed to the topology change under equilibrium response, rather than to re-parameterisation elsewhere in the network.

For each demand value in the scan, equilibrium is computed for both network states under deterministic user equilibrium (DUE), and the paradox indicator is evaluated using the difference in average travel time:

$$\Delta ATT(Q_{rs}) = ATT_{\text{after}}(Q_{rs}) - ATT_{\text{before}}(Q_{rs}). \tag{4.2}$$

Average travel time is computed consistently from the equilibrium solution as

$$ATT = \frac{TSTT}{Q_{rs}}, \qquad TSTT = \sum_{p \in \mathcal{P}_{rs}} f_p^* \, C_p(\mathbf{x}^*),$$

where $C_p(\mathbf{x}^*)$ is the path cost obtained by summing link costs along path p under the equilibrium link flows.

To prevent numerical noise from being misclassified as paradox, the decision rule uses two explicit thresholds: very small differences are treated as numerical zeros if

$$|\Delta ATT| \leq \varepsilon_{\text{zero}},$$

and otherwise a paradox flag is set if

$$\Delta ATT > \varepsilon_{\text{paradox}}.$$

In this thesis, the reproduction scans use $\varepsilon_{\text{zero}} = 10^{-2}$ and $\varepsilon_{\text{paradox}} = 10^{-3}$, and the same rule is retained in later large-scale scans for comparability across figures and parameter grids.

The output of a demand scan is therefore the function $Q_{rs} \mapsto \Delta ATT(Q_{rs})$, reported as

(i) a discrete curve over demand,

(ii) inferred demand intervals where $\Delta ATT > 0$ under the above rule, and

(iii) representative equilibrium link-flow diagnostics to support mechanism interpretation.

Reproducibility is treated as a first-class constraint: each scan is fully determined by an explicit configuration record comprising

(i) the network generator settings (diamond, one-way, $(r, s) = (1, 4)$, before/after link lists and the link legend above),

(ii) the per-link affine cost parameters (Table 4.1),

(iii) the demand grid Q_{rs} is from 1 to 20, (iv) the equilibrium model is DUE, and

(iv) numerical solver options.

In the current implementation, DUE is solved with a Method of Successive Averages (MSA) scheme using step size $a_k = 1/k$, with a maximum iteration budget of 5×10^4 and convergence tolerance 10^{-8} under a relative change criterion on link flows. At each MSA iteration, an auxiliary assignment is performed within the enumerated set $\mathcal{P}_{rs}$ by loading all demand onto the currently minimum-cost path(s); when multiple paths share the same minimum cost, the auxiliary flow is split uniformly across them. A restart safeguard is applied to mitigate stagnation: the algorithm monitors progress over a window of 200 iterations and triggers at most 5 restarts when the improvement falls below 10^{-6}.

Path enumeration is deterministic and, for this benchmark, all simple r–s paths are included with the explicit cap $P_{\max} = 20000$. Since the pipeline contains no random components for this benchmark, rerunning the scan under the stated configuration reproduces identical outputs.

4.2. Cost function families and parameterization

The specification of link cost functions is a central modelling choice in equilibrium analysis, because it determines how flow-dependent congestion effects are translated into perceived route costs and, consequently, into network-wide performance. In Braess-type studies, cost functions play a dual role. They provide an operational representation of infrastructure performance (e.g. baseline travel time, capacity constraints, and congestion sensitivity), while simultaneously shaping the analytical and numerical conditions under which paradoxical reallocation may arise. This section documents the cost-function families adopted in the thesis, the per-link parameterization strategy, and the conventions used to preserve comparability between before and after network states.

Classical presentations of the Braess' paradox typically adopt linear link costs, as they facilitate analytical derivations and admit closed-form equilibria in small benchmark networks [21, 56]. However, empirical congestion processes are rarely linear in flow, and planning practice routinely employs nonlinear performance functions to reflect rapidly

increasing marginal delay as utilisation approaches capacity. Restricting paradox analysis to a single linear specification therefore risks conflating structural effects with artefacts of a particular functional form.

For this reason, the thesis considers multiple cost-function families within a single computational interface. Linear functions are retained as a baseline because they support direct comparison with canonical results and with the symbolic derivations developed later in the thesis. In addition, Bureau of Public Roads (BPR) type functions are included to represent nonlinear congestion growth, consistent with standard transport-modelling references [2]. Finally, to probe robustness beyond these two widely used families while keeping the parameter interface minimal, the framework also supports simple polynomial variants (constant and quadratic), enabling controlled convexity changes without introducing additional model structure.

All cost functions in this thesis are parameterized per link and encoded in arrays aligned with a fixed link ordering (the after-network link legend). This convention is essential for reproducibility because solver inputs and outputs are defined on link-indexed vectors, and any ambiguity in index-to-arc mapping propagates directly into cost evaluation and equilibrium reporting.

For each experiment, the after-network legend is treated as the authoritative ordering. Cost parameters are specified in exactly that order, and the corresponding before-network cost object is obtained by removing entries associated with links that exist only in the after state, while keeping identical parameter values for links shared between the two states. This neutrality condition ensures that before–after comparisons isolate the effect of added connectivity (and, where explicitly studied, the parameterization assigned to added links), rather than reflecting hidden recalibration of baseline links.

For the classic diamond benchmark (Section 4.1), the after-network legend is fixed as $a = 1 : (1 \to 2)$, $a = 2 : (2 \to 4)$, $a = 3 : (1 \to 3)$, $a = 4 : (3 \to 4)$, $a = 5 : (2 \to 3)$. For ladder networks, the legend depends on the ladder size and direction convention as well as on the specified set of added links; in such cases, the reported legend defines the required ordering for all parameter arrays.

The linear family is represented by link travel-time functions of the form

$$t_a(x_a) = b_a + m_a x_a, \tag{4.3}$$

where x_a denotes the link flow and $t_a(\cdot)$ the resulting link travel time. The parameter b_a represents the baseline (free-flow) travel time on link a, while m_a controls the sensitivity of travel time to flow. This family provides the baseline reference case used throughout the thesis.

Costs are parameterised per link and encoded as arrays aligned with the after-network link legend (Section 4.2). In this representation, constant-cost links require no additional functional type: they are implemented as the special case $m_a = 0$, yielding $t_a(x_a) = b_a$.

For the classical reproduction benchmark on the diamond network, the link legend is fixed

as $a = 1 : (1 \to 2)$, $a = 2 : (2 \to 4)$, $a = 3 : (1 \to 3)$, $a = 4 : (3 \to 4)$, $a = 5 : (2 \to 3)$. Given this legend, the benchmark is fully determined by the parameter arrays (b_a) and (m_a) together with the demand scan $Q_{rs} = 1{:}20$ for the single OD pair $(r, s) = (1, 4)$ (Section 4.1).

To probe robustness beyond the linear and BPR specifications, a complementary polynomial test suite is considered in which link costs are prescribed as simple explicit functions of flow. Throughout this suite, link costs are restricted to three transparent functional forms: a linear term $10x$, a quadratic term $10x^2$, and a constant term 100. In what follows, x denotes the flow on the corresponding link in each table entry.

The polynomial suite is evaluated on the same diamond before/after topology as the classical benchmark. The before network contains four links (1−2), (2−4), (1−3), (3−4), and the after network adds the connector (2−3). To remove ambiguity when specifying per-link functions, the polynomial cases adopt the following fixed ordering for the after network:

$$(1{-}2),\ (2{-}4),\ (1{-}3),\ (3{-}4),\ (2{-}3),$$

and the before network uses the same order with the added link removed:

$$(1{-}2),\ (2{-}4),\ (1{-}3),\ (3{-}4).$$

All case definitions below follow this ordering.

Six reproducible combinations are defined by assigning one of the above functional forms to each link. Table 4.2 reports the complete mapping from links to cost functions. For each case, the before network inherits the same assignments on shared links, and the connector column is omitted.

Table 4.2: Polynomial robustness suite: per-link cost assignments for the diamond after-network (the before-network uses the same assignments on shared links and excludes the connector (2−3)).

Case	(1−2)	(2−4)	(1−3)	(3−4)	(2−3)
1 (linear + constant)	$10x$	100	100	$10x$	$10x$
2 (quadratic + constant; linear connector)	$10x^2$	100	100	$10x^2$	$10x$
3 (all quadratic)	$10x^2$	$10x^2$	$10x^2$	$10x^2$	$10x^2$
4 (linear–quadratic mix; constant connector)	$10x$	$10x$	$10x^2$	$10x^2$	100
5 (constant + linear + quadratic)	100	$10x^2$	$10x$	100	$10x$
6 (quadratic main links; constant upper branch)	$10x^2$	100	$10x^2$	$10x^2$	$10x^2$

The suite is evaluated for the single OD pair $(r, s) = (1, 4)$ under a discrete demand scan $Q_{rs} = 1{:}50$. For each Q_{rs}, equilibrium is computed for the before and after networks

under the same numerical settings. Performance is reported via total system travel time,

$$TSTT = \sum_{a \in A} x_a \, t_a(x_a),$$

and equivalently by average travel time $ATT = TSTT/Q_{rs}$. A case is classified as paradoxical at demand Q_{rs} if the after network exhibits a strictly higher equilibrium performance measure than the before network under the same Q_{rs} and the same numerical tolerance.

To represent nonlinear congestion growth in a form commonly used in planning practice, the thesis adopts the BPR function

$$t_a(x_a) = t_{0,a} \left[1 + \alpha \left(\frac{x_a}{c_a} \right)^{\beta} \right], \tag{4.4}$$

where $t_{0,a}$ denotes the free-flow travel time of link a, c_a denotes its nominal capacity, and (α, β) are shape parameters controlling the magnitude and curvature of the congestion term.

In the experimental configuration used in this thesis, $t_{0,a}$ and c_a are specified per link and interpreted according to the after-network link legend defined in Section 4.1, while α and β are treated as global parameters shared by all BPR-type links within a given experiment. This convention allows heterogeneous free-flow times and capacities without introducing additional per-link curvature parameters, and it also ensures that parameter vectors remain reproducible because their indices correspond to a fixed arc ordering.

For the nonlinear diamond setting in which all links in the after network are evaluated with the BPR form (4.4), the parameter values are fixed to $\alpha = 0.15$ and $\beta = 4.0$. Under the after-network legend $a = 1 : (1 \to 2)$, $a = 2 : (2 \to 4)$, $a = 3 : (1 \to 3)$, $a = 4 : (3 \to 4)$, $a = 5 : (2 \to 3)$, the per-link free-flow times are assigned as

$$t_{0,1} = 40, \quad t_{0,2} = 50, \quad t_{0,3} = 50, \quad t_{0,4} = 40, \quad t_{0,5} = 10,$$

and capacities are set uniformly as

$$c_1 = c_2 = c_3 = c_4 = c_5 = 2000.$$

A representative demand scan for this setting is from 200 to 6000 With a steplength of 200 for the single OD pair $(r, s) = (1, 4)$.

The same per-link architecture also supports a mixed specification in which only the newly added link adopts BPR costs, while all links shared between before and after states retain linear costs. In the diamond benchmark, the added link is $a = 5 : (2 \to 3)$ under the after-network legend. A reproducible mixed configuration therefore keeps the linear specification $t_a(x_a) = b_a + m_a x_a$ on the shared links $a = 1, \ldots, 4$ (with (b_a, m_a) fixed as documented for the benchmark), and assigns the BPR form (4.4) only to the added link

$a = 5$ using

$$t_{0,5} = 10, \qquad c_5 = 2000, \qquad \alpha = 0.15, \qquad \beta = 4.0.$$

In this mixed setting, the nonlinear congestion component introduced by the intervention is confined to the new connector itself, while neutrality is preserved for the baseline links through unchanged functional types and parameters on all shared arcs.

A critical modelling decision concerns the cost assigned to newly added links in the after network. To avoid ambiguity, two conventions are enforced throughout the thesis.

First, all links that are common to both before and after networks retain identical functional types and identical parameter values. This neutrality condition is essential for a fair comparison because it ensures that performance differences are attributable to the structural intervention rather than to hidden re-parameterisation.

Second, the cost of each added link is always specified explicitly (both functional type and parameter values). This may be done either by direct inclusion in the after-network parameter arrays or via an override convention that assigns parameters only to the added-link subset while leaving shared links unchanged. In either case, the specification makes it transparent whether paradoxical behaviour arises primarily from topology or from assumptions about the performance of new infrastructure.

Cost functions interact with demand in nontrivial ways. Under linear costs, travel time increases proportionally with flow; under quadratic and BPR costs, increasing marginal delay can dominate beyond certain utilisation levels, potentially shifting or suppressing paradox regimes. Accordingly, demand grids are treated as part of the experimental configuration record. Within any scan, the same demand grid Q_{rs} is applied to both before and after networks, and paradox detection is based on differences in average travel time rather than absolute cost levels.

For the classical reproduction benchmark, the thesis uses $Q_{rs} = 1{:}20$ (Section 4.1). For nonlinear scans (quadratic/BPR), wider demand ranges are documented alongside each experiment configuration to ensure that both low-congestion and high-congestion regimes are represented.

By adopting multiple cost-function families and explicit per-link parameterization, the computational framework does not presuppose the presence or absence of the Braess' paradox. Instead, paradox occurrence becomes an empirical property of a precisely defined parameter space under a documented demand grid and a controlled before–after comparison protocol. This perspective structures the later chapters: numerical scans in Chapter 5 map how paradox regions shift with topology, convexity, and behavioural assumptions, while the symbolic derivations in Chapter 6 are used to interpret selected numerical patterns and to clarify why certain parameter regimes admit or exclude paradoxical outcomes.

4.3. Deterministic versus stochastic equilibrium

The equilibrium conditions introduced earlier specify what constitutes a deterministic or stochastic equilibrium, but numerical procedures are required to compute equilibria in the networks considered in this thesis. This section documents the assignment models and solver settings used to compute deterministic user equilibrium (DUE) and stochastic user equilibrium (SUE) in a reproducible manner, under consistent conventions across network states, cost-function families, and demand levels.

DUE is computed with a method of successive averages (MSA) scheme on a fixed admissible path set $\mathcal{P}$. At iteration k, the current path-flow vector $\mathbf{f}^{(k)}$ induces link flows through the path–link mapping $\mathbf{x}^{(k)} = \Delta\mathbf{f}^{(k)}$. Given $\mathbf{x}^{(k)}$, link costs are evaluated as $t_a(x_a^{(k)})$, and path costs are obtained by summing link costs along each path,

$$C_p\big(\mathbf{x}^{(k)}\big) = \sum_{a\in A} \Delta_{pa}\, t_a\big(x_a^{(k)}\big)\,, \qquad p \in \mathcal{P}.$$

An auxiliary path-flow vector $\mathbf{y}^{(k)}$ is then constructed by assigning the entire demand Q_{rs} to the minimum-cost path set under the current costs. When multiple minimum-cost paths exist, the demand is split evenly across these paths to keep the auxiliary loading uniquely defined: if $\mathcal{P}_{\min} = \arg\min_{p\in\mathcal{P}} C_p(\mathbf{x}^{(k)})$, then

$$y_p^{(k)} = \begin{cases} Q_{rs}/|\mathcal{P}_{\min}|, & p \in \mathcal{P}_{\min}, \\ 0, & \text{otherwise.} \end{cases}$$

The MSA update applies the classical diminishing weight $1/k$ (with $k = 1, 2, \dots$),

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \frac{1}{k}\left(\mathbf{y}^{(k)} - \mathbf{f}^{(k)}\right). \tag{4.5}$$

This procedure is repeated until the convergence criterion stated below is met or the iteration budget is exhausted.

SUE is computed with the same MSA backbone as in (4.5), but replaces the deterministic auxiliary loading with a logit-based stochastic loading on the same path set $\mathcal{P}$. At each iteration, path choice probabilities are computed from current path costs using

$$P_p^{(k)} = \frac{\exp\big(-C_p(\mathbf{x}^{(k)})/\theta\big)}{\sum_{q\in\mathcal{P}} \exp(-C_q(\mathbf{x}^{(k)})/\theta)}, \qquad p \in \mathcal{P},$$

and the auxiliary path-flow vector is

$$\mathbf{y}^{(k)} = Q_{rs}\, \mathbf{P}^{(k)}.$$

The subsequent update of $\mathbf{f}^{(k)}$ follows (4.5) with the same $1/k$ averaging weights. Under

this design, DUE and SUE share the same representation of flows, the same evaluation of link and path costs, and the same stopping rule; they differ only in how auxiliary flows are generated from path costs.

Behavioural dispersion is controlled by θ. Smaller θ produces more concentrated choices around the lowest-cost paths, while larger θ yields more dispersed route choice across near-minimum alternatives. In the numerical experiments reported in this thesis, SUE is evaluated at three dispersion levels,

$$\theta \in \{0.2,\ 0.8,\ 5\},$$

so that the sensitivity of paradox patterns to route-choice dispersion can be examined under a fixed and explicitly stated set of behavioural assumptions. For any fixed θ, the computation is deterministic and produces identical outputs when rerun under the same configuration.

To stabilise convergence across large parameter scans and to reduce the risk of stagnation in networks with near-indifference among several routes, the solver incorporates a restart mechanism. During the iteration, progress is monitored using the convergence metric defined in the next subsection. If the best observed value of this metric fails to improve by at least 10^{-6} over 200 consecutive iterations, the current iterate is reset to the current auxiliary loading (i.e. $\mathbf{f}^{(k)} \leftarrow \mathbf{y}^{(k)}$). After a restart, the iteration counter used in the MSA weight is reset so that the averaging weight again starts from $1/k$ with $k = 1$. At most five restarts are permitted within a single equilibrium solve.

Convergence is assessed in link-flow space using the relative change between successive iterates, normalised by total demand:

$$\frac{\left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right\|_1}{\max(Q_{rs}, 1)} \leq \varepsilon. \tag{4.6}$$

The tolerance is fixed to $\varepsilon = 10^{-8}$, and the maximum iteration budget is 5×10^4 iterations. If the inequality in (4.6) is not satisfied within the iteration budget, the final iterate at termination is retained as the numerical equilibrium approximation under the stated settings.

Before/after comparisons are computed under identical numerical settings. For a given demand value Q_{rs} and equilibrium model (DUE or SUE), both network states are solved using the same MSA update rule (4.5), the same convergence tolerance ε, the same iteration budget, and the same restart settings. Moreover, costs on links shared between the two states are identical by construction, so that differences in equilibrium performance reflect the structural intervention and any explicitly specified cost assignment on added links, rather than differences in numerical treatment.

The controlled numerical design above ensures that paradox classification is interpretable as a difference between two equilibria computed to the same numerical standard. Using the same MSA backbone for DUE and SUE isolates behavioural dispersion (through

θ) as the intended source of differences between deterministic and stochastic outcomes, while fixed tolerances and restart rules support consistent before/after comparisons across demand levels and cost-function families.

4.4. Network families, interventions, and path set construction

This section specifies the network families used in the computational experiments and the precise rules by which the corresponding before and after networks are generated. Because the Braess' paradox is defined as an equilibrium comparison under a structural intervention, the intervention is stated unambiguously in terms of node labels, directed links, and admissible r–s paths. Throughout this section, the origin–destination pair is denoted by (r, s) and the associated demand by Q_{rs}.

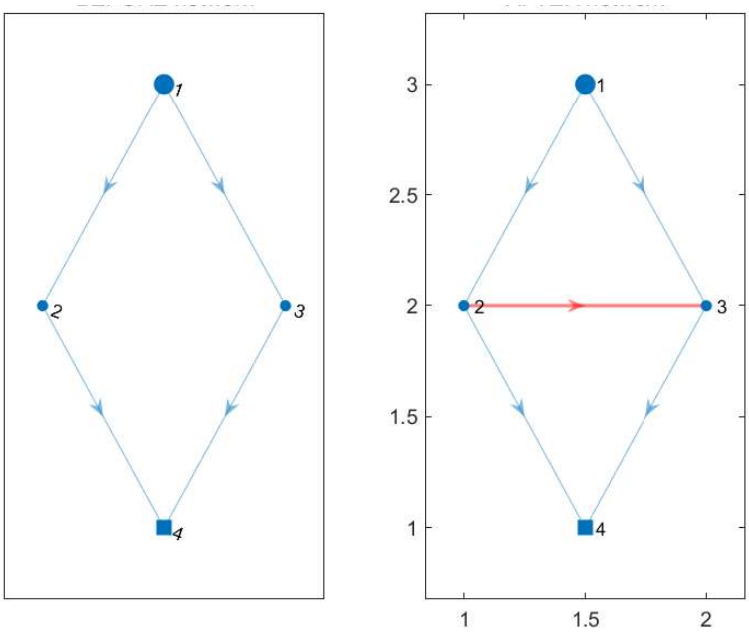


Figure 4.2: Diamond network (BEFORE vs AFTER). Node 1 is the origin r, node 4 is the destination s. The intervention adds the directed connector $2 \to 3$.

The diamond benchmark is defined on the node set $\{1, 2, 3, 4\}$ with origin–destination pair $(r, s) = (1, 4)$. The before network contains four directed links

$$(1 \to 2),\ (2 \to 4),\ (1 \to 3),\ (3 \to 4),$$

which induce exactly two simple r–s paths: 1–2–4 and 1–3–4. The after network is obtained by adding a single directed connector $(2 \to 3)$, which introduces the additional simple path 1–2–3–4. This benchmark is used as a controlled reference case because the admissible path set is small and the intervention is defined by a single added directed link.

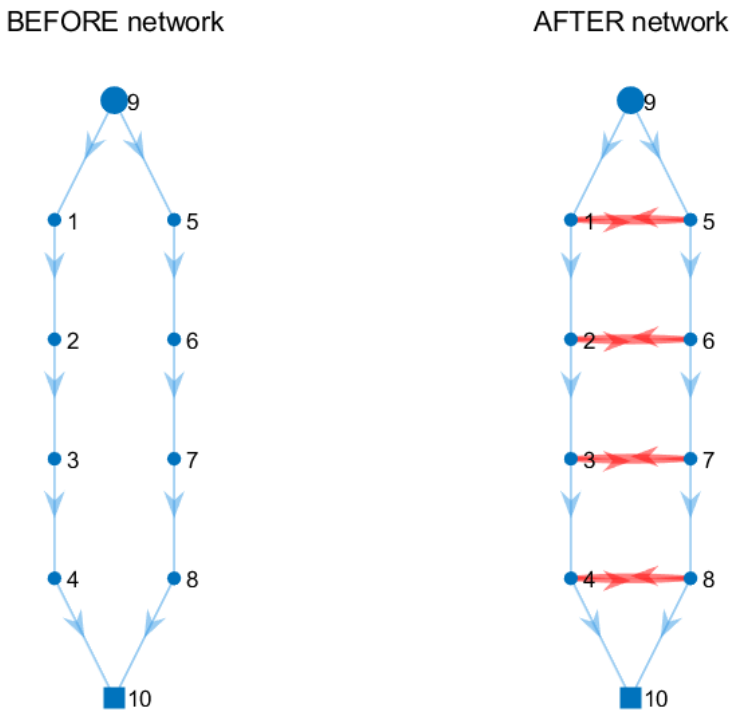


Figure 4.3: Ladder network (BEFORE vs AFTER) for $n = 3$. The origin and destination are $r = 9$ and $s = 10$. The intervention adds bidirectional bypass links $1 \leftrightarrow 5$, $2 \leftrightarrow 6$, $3 \leftrightarrow 7$, and $4 \leftrightarrow 8$.

The ladder family is parameterised by an integer $n \geq 0$. For each n, the network contains two directed rails of length $n + 1$ together with two terminal nodes (r, s). Node labels are assigned deterministically so that the two rails are represented by consecutive integer sequences: the first rail uses nodes

$$1, 2, \ldots, n + 1,$$

and the second rail uses nodes

$$n + 2, n + 3, \ldots, 2n + 2.$$

The terminal nodes are then defined as

$$r = 2n + 3, \qquad s = 2n + 4.$$

Baseline topology (BEFORE). The before ladder includes only directed rail links oriented from r toward s. Specifically, the terminal-to-rail links are $(r \to 1)$ and $(r \to (n + 2))$, the longitudinal rail links are

$$(i \to i + 1) \quad \text{for } i = 1, \ldots, n, \qquad (j \to j + 1) \quad \text{for } j = n + 2, \ldots, 2n + 1,$$

and the rail-to-terminal links are $((n+1) \to s)$ and $((2n+2) \to s)$. This baseline contains two directed rail routes from r to s and is acyclic by construction.

Intervention topology (AFTER). The structural intervention in the ladder family is defined by adding bypass links between the two rails at selected positions. For a given position index $i \in \{1, \ldots, n+1\}$, the corresponding bypass connects node i on the first rail to node $n+1+i$ on the second rail. In the main ladder experiments, each bypass is implemented as a bidirectional pair of directed links, so a selected position i contributes both $i \to (n+1+i)$ and $(n+1+i) \to i$. Interventions are generated reproducibly either by cumulative addition of bypass positions or by selecting arbitrary subsets of candidate positions.

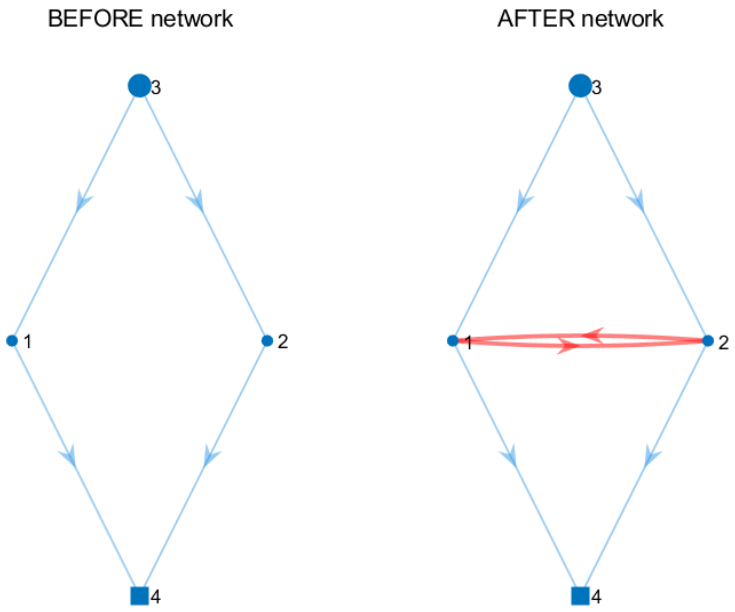


Figure 4.4: Ladder network (BEFORE vs AFTER) for $n = 0$. The origin and destination are $r = 3$ and $s = 4$. The intervention adds bidirectional bypass links $1 \leftrightarrow 2$.

Before analysing ladders with multiple rail segments, a transitional case with $n = 0$ is used to validate that replacing a single directed shortcut by a bidirectional bypass convention does not, by itself, introduce spurious paradoxical behaviour. When $n = 0$, the rail nodes are $\{1\}$ and $\{2\}$ and the terminals are $(r, s) = (3, 4)$. The before network consists of

$$(3 \to 1),\ (3 \to 2),\ (1 \to 4),\ (2 \to 4),$$

which yields two parallel r–s routes. A first after-network variant mirrors the classical linear benchmark while modifying only the directionality of the added connection: instead of adding a single directed connector, a bidirectional bypass $(1 \to 2)$ and $(2 \to 1)$ is introduced. All pre-existing links retain the classical coefficients, so any performance change can be attributed solely to the bidirectional representation of the bypass. This transitional check confirms whether the bidirectional convention affects equilibrium outcomes relative to the directed benchmark, thereby providing a controlled bridge between

the two-terminal setting and the multi-segment ladders.

The ladder experiments that follow adopt a rule-based cost assignment designed for both computational tractability and conceptual continuity with the classical Braess benchmark. First, using a small number of link classes with shared coefficients reduces the dimensionality of parameter specification as n grows, enabling systematic scans over interventions without re-tuning link-by-link parameters. Second, the resulting structure preserves the key qualitative ingredients of the classical paradox:

(i) congestion-sensitive links that generate endogenous delays,

(ii) flow-independent connectors that provide fixed baseline costs, and

(iii) a comparatively low-cost bypass that can attract flow and trigger adverse equilibrium reallocation.

All link travel times are specified in the affine-linear form

$$t_a(x_a) = b_a + m_a x_a, \qquad a \in A, \tag{4.7}$$

where x_a denotes the flow on directed link a, $b_a \geq 0$ is a constant component, and $m_a \geq 0$ is the congestion sensitivity. For ladder networks, coefficients are assigned deterministically by link class using the node labelling

$$U(i) = i, \qquad L(i) = i + (n+1), \qquad i = 1, \ldots, n+1,$$

together with terminals $r = 2n+3$ and $s = 2n+4$. Here, the two longitudinal chains are referred to as the U-chain (the directed sequence $U(1) \to U(2) \to \cdots \to U(n+1)$) and the L-chain (the directed sequence $L(1) \to L(2) \to \cdots \to L(n+1)$). This terminology is purely notational and does not rely on any geometric "upper/lower" or "left/right" placement in schematic drawings. The same class-based rules apply to every $n \geq 1$ and to every intervention realisation.

(i) Longitudinal chain links (alternating congestible and constant segments). Directed links that advance along the two longitudinal chains are assigned coefficients according to the segment index parity. Along the U-chain,

$$t_{U(i)\to U(i+1)}(x) = \begin{cases} x, & i = 1, 3, 5, \ldots, \\ 10, & i = 2, 4, 6, \ldots, \end{cases} \qquad i = 1, \ldots, n, \tag{4.8}$$

whereas along the L-chain the parity pattern is reversed,

$$t_{L(i)\to L(i+1)}(x) = \begin{cases} 10, & i = 1, 3, 5, \ldots, \\ x, & i = 2, 4, 6, \ldots, \end{cases} \qquad i = 1, \ldots, n. \tag{4.9}$$

Equivalently, congestible segments satisfy $b_a = 0$ and $m_a = 1$, while constant segments satisfy $b_a = 10$ and $m_a = 0$. This alternating assignment yields a structured ladder in

which congestion-sensitive and fixed-cost segments are interleaved along each longitudinal chain.

(ii) Terminal connectors (mixed slope/constant edges). The four connectors incident to the terminals are assigned as

$$t_{r\to U(1)}(x) = x, \qquad t_{r\to L(1)}(x) = 10, \qquad t_{U(n+1)\to s}(x) = 10, \qquad t_{L(n+1)\to s}(x) = x, \tag{4.10}$$

so that two terminal connectors are congestion-sensitive ($b_a = 0, m_a = 1$) and two are flow-independent constants ($b_a = 10, m_a = 0$).

(iii) Bypass links (low-cost constant shortcut). Each activated bypass position $i \in \mathcal{I} \subseteq \{1, \ldots, n+1\}$ introduces bidirectional cross-links between the rails, each assigned a small constant travel time:

$$t_{U(i)\to L(i)}(x) = 2, \qquad t_{L(i)\to U(i)}(x) = 2, \qquad i \in \mathcal{I}. \tag{4.11}$$

Hence bypasses satisfy $b_a = 2$ and $m_a = 0$ and act as attractive shortcuts whose equilibrium utilisation depends on congestion on the surrounding rail structure.

Default rule (unclassified edges). Any directed link not captured by the rules above is assigned the baseline congestible form $t(x) = x$ (i.e., $b_a = 0$ and $m_a = 1$).

Inheritance from BEFORE to AFTER. All links present in the before network retain their assigned coefficients unchanged in the after network. Only the newly added bypass links are introduced by the intervention, and their coefficients are assigned by (4.11). This construction ensures that before/after comparisons isolate the effect of added connectivity under a fixed behavioural and functional specification.

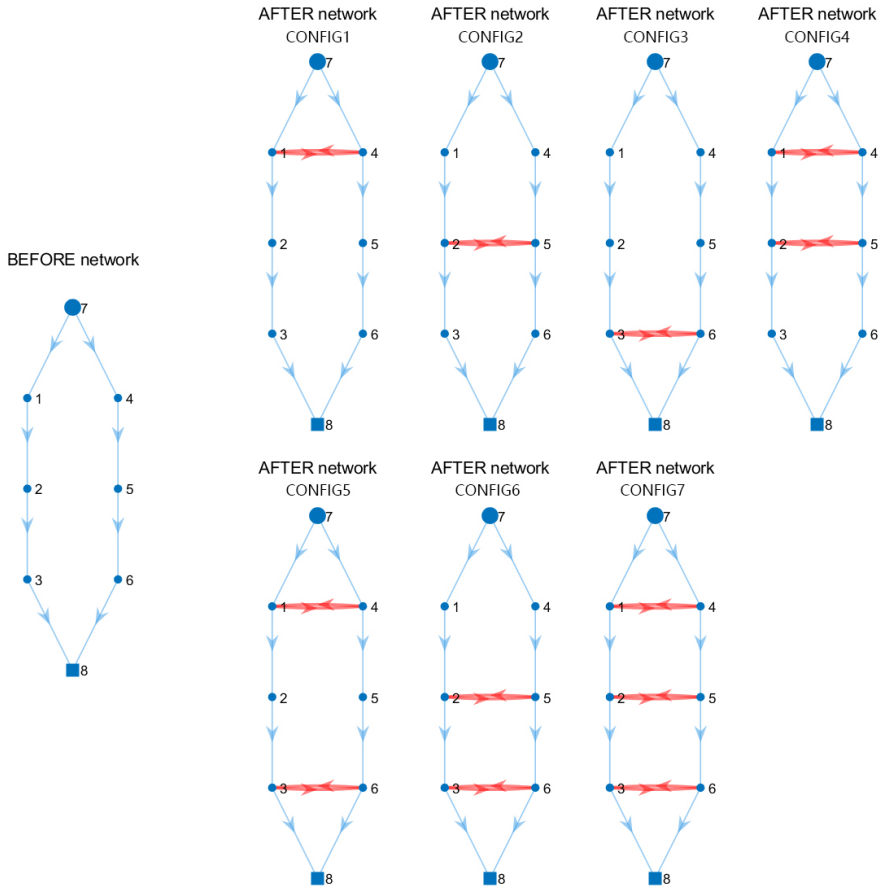


Figure 4.5: Ladder network (BEFORE vs AFTER) for $n = 2$ with all configurations. The origin and destination are $(r, s) = (7, 8)$. The candidate intervention set consists of the three bidirectional bypass pairs $1 \leftrightarrow 4$, $2 \leftrightarrow 5$, and $3 \leftrightarrow 6$; configurations Config1-Config7 correspond to all non-empty subsets of these candidates.

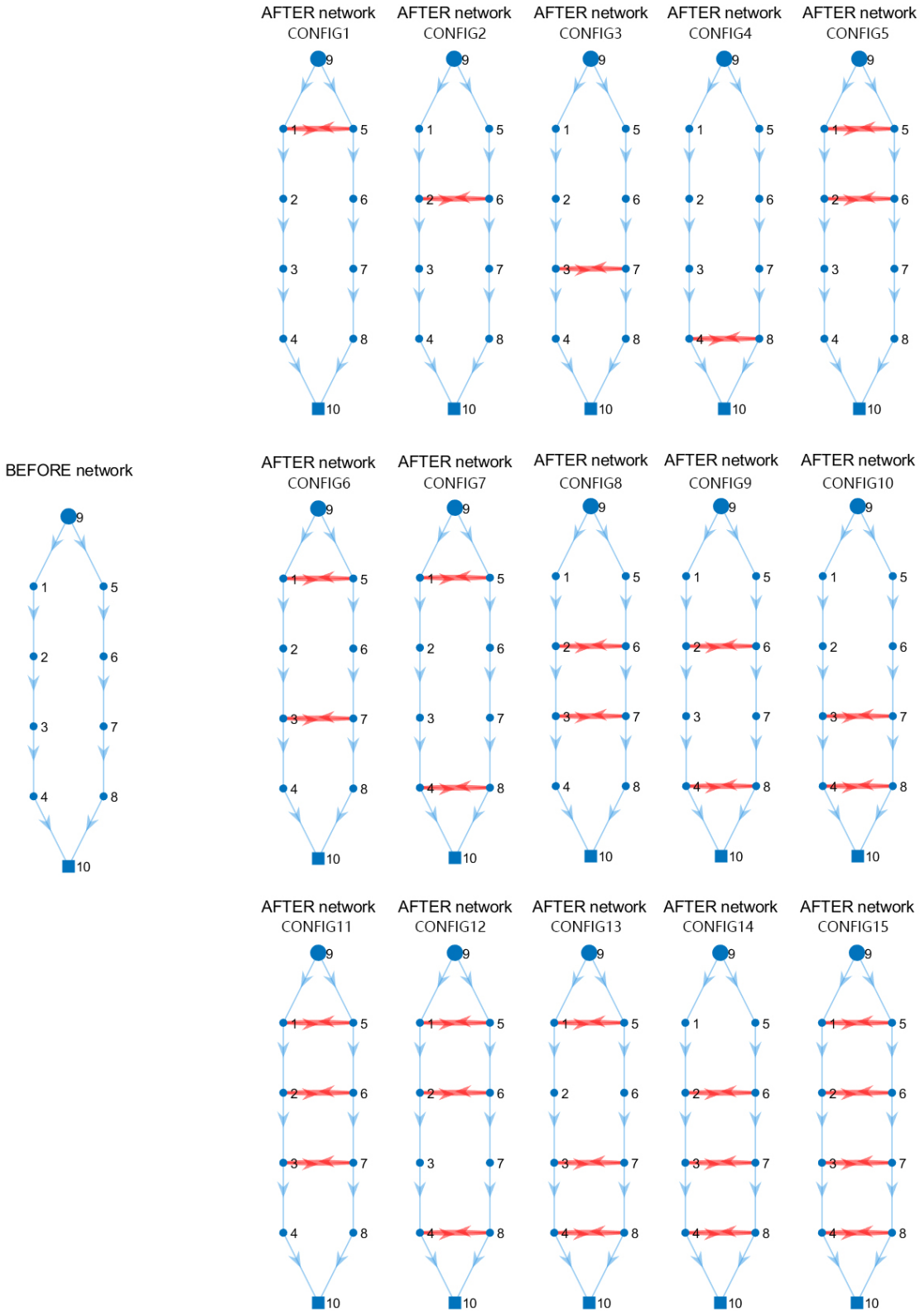


Figure 4.6: Ladder network (BEFORE vs AFTER) for $n = 3$ with all configurations. The origin and destination are $(r, s) = (9, 10)$. The candidate intervention set consists of the four bidirectional bypass pairs $1 \leftrightarrow 5$, $2 \leftrightarrow 6$, $3 \leftrightarrow 7$, and $4 \leftrightarrow 8$; configurations Config1-Config15 correspond to all non-empty subsets of these candidates.

The ladder computations reported in the main analysis focus on $n = 2$ and $n = 3$.

For $n = 2$, the rail nodes are $\{1, 2, 3\}$ and $\{4, 5, 6\}$ with terminals $(r, s) = (7, 8)$. Candidate bypass positions are $i \in \{1, 2, 3\}$, corresponding to the bidirectional bypass pairs $1 \leftrightarrow 4$, $2 \leftrightarrow 5$, and $3 \leftrightarrow 6$. Figure 4.5 enumerates the resulting after networks as the complete set of non-empty subsets of these candidates, yielding $2^3 - 1 = 7$ configurations. Specifically, Configurations 1–3 activate a single bypass (top, middle, and bottom, respectively); Configurations 4–6 activate two bypasses in all pairwise combinations; and Configuration 7 activates all three bypasses simultaneously. This organisation makes it possible to separate a shortcut-density effect (the number of active bypass positions) from any purely positional effect within the same ladder size.

For $n = 3$, the rail nodes are $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ with terminals $(r, s) = (9, 10)$. Candidate bypass positions are $i \in \{1, 2, 3, 4\}$, corresponding to the bidirectional bypass pairs $1 \leftrightarrow 5$, $2 \leftrightarrow 6$, $3 \leftrightarrow 7$, and $4 \leftrightarrow 8$. Accordingly, Figure 4.6 reports all $2^4 - 1 = 15$ non-empty subsets of the candidate set. The configuration index again follows a cardinality-based organisation: Configurations 1–4 activate exactly one bypass (from top to bottom); Configurations 5–10 activate exactly two bypasses (all pairwise combinations); Configurations 11–14 activate exactly three bypasses (all triple combinations); and Configuration 15 activates all four bypasses. This complete enumeration ensures that comparisons across configurations are not driven by an arbitrary selection of added links, but by controlled variation in the size and composition of the intervention set.

Demand is scanned using integer values of Q_{rs} from 1 to 20 with unit step, applied identically to the before and after networks. The intervention set is generated either cumulatively (increasing the number of active bypass positions one by one) or by selecting arbitrary subsets of candidate positions, as visualised in Figs. 4.5–4.6. In addition to these rail-aligned bypasses, the framework supports user-specified added-link lists; outcomes from such custom interventions are documented separately so that the main analysis remains focused on the canonical ladder family.

4.5. Numerical computation and symbolic computation

This section specifies the experimental protocols used to compute paired equilibria for the before and after networks and to document the outcomes in a reproducible form. The protocol consists of

(i) a demand scan over a fixed discrete set of values of Q_{rs},

(ii) uniform performance evaluation and paradox classification rules applied symmetrically to the two network states,

(iii) explicitly stated numerical stopping and restart settings for equilibrium computation, and

(iv) a symbolic procedure for deriving linear cost families in the two-terminal benchmark and checking them numerically.

Each experiment is defined by a fixed network instance (node set, directed link set, and OD pair (r, s)), a link cost specification, and an equilibrium model (DUE or SUE). The demand scan is implemented as a discrete grid of values of Q_{rs}, which is fixed for a given experiment and applied identically to the before and after networks.

In the classical two-terminal reproduction, Q_{rs} takes all integer values from 1 to 20 with step size 1. In the polynomial robustness suite, the same convention is extended to the integer range from 1 to 50 with step size 1. In nonlinear congestion experiments under BPR costs, the demand grid is scaled to the units implied by the capacity parameters, and a representative setting uses values from 200 to 6000 in increments of 200. For each experiment, the chosen demand grid is treated as part of the model specification and is recorded together with the network and cost parameters.

When additional parameters are scanned (for example, the SUE dispersion parameter θ or the choice of bypass positions in ladder networks), the demand scan remains the inner loop: for each parameter configuration, the full set of paired before/after equilibria is computed across the same Q_{rs} values. This organisation enforces matched comparisons by construction.

For each demand value Q_{rs}, the solver computes an equilibrium for the before network and an equilibrium for the after network under the same numerical settings. Performance is recorded using total system travel time,

$$TSTT = \sum_{a \in A} x_a^* \, t_a(x_a^*) \,,$$

and average travel time,

$$ATT = \frac{TSTT}{Q_{rs}}.$$

Paradox classification is based on the demand-dependent difference in average travel time,

$$\Delta ATT(Q_{rs}) = ATT_{\text{after}}(Q_{rs}) - ATT_{\text{before}}(Q_{rs}).$$

Two numerical thresholds are used to ensure that classification is not driven by solver noise. Differences are treated as numerical zeros when $|\Delta ATT| \leq 10^{-2}$, and a demand point is classified as paradoxical when $\Delta ATT > 10^{-3}$. The same rule is applied uniformly across all demand scans so that paradox identification remains comparable across network families and cost specifications.

All equilibrium solves use a Method of Successive Averages update with step size $1/k$, where k starts at 1 and increases by one at each iteration. Convergence is assessed in link-flow space using the relative change between successive iterates, normalised by total demand:

$$\frac{\left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right\|_1}{\max(Q_{rs}, 1)} \leq 10^{-8}.$$

The maximum iteration budget is 5×10^4 per equilibrium computation.

To reduce the risk of stagnation, a restart safeguard is applied. If the convergence metric fails to improve by at least 10^{-6} over a window of 200 consecutive iterations, the current iterate is reset to the current auxiliary solution implied by the current costs, and the local averaging sequence is restarted. At most five restarts are permitted for a given demand point. The same stopping rule, iteration budget, and restart parameters are applied to the before and after networks. If either state fails to reach the convergence threshold within these limits, that demand point is flagged as numerically unreliable and is excluded from paradox classification.

For SUE runs, the dispersion parameter is fixed to the three values $\theta \in \{0.2, 0.8, 5\}$, and each run uses exactly one of these values as part of its configuration.

For each demand value in a scan, the recorded output includes: the demand Q_{rs}; the equilibrium values of ATT_{before} and ATT_{after}; the derived difference ΔATT; the paradox flag under the stated thresholds; and equilibrium link-flow vectors for both network states written in the link-legend ordering of that state. When SUE is used, the value of θ is stored alongside the same row. These records provide a complete and self-contained description of each paired equilibrium comparison at each demand level.

Symbolic computation is used to derive families of linear link cost functions for the two-terminal benchmark under DUE, subject to prescribed equilibrium and performance relationships. Linear costs are written in slope–intercept form,

$$t_a(x_a) = a_a x_a + b_a,$$

which is equivalent to the alternative notation $t_a(x_a) = b_a + m_a x_a$ under the identification $a_a \equiv m_a$.

The symbolic derivation is performed on the diamond benchmark using the after-network link legend

$$a = 1 : (1 \to 2), \quad a = 2 : (2 \to 4), \quad a = 3 : (1 \to 3), \quad a = 4 : (3 \to 4), \quad a = 5 : (2 \to 3).$$

To preserve the symmetry of the benchmark while reducing the number of free parameters, the four baseline links are grouped into two classes. The two outer links share a common pair (a_U, b_U),

$$t_1(x_1) = a_U x_1 + b_U, \qquad t_4(x_4) = a_U x_4 + b_U,$$

and the two inner links share a common pair (a_M, b_M),

$$t_2(x_2) = a_M x_2 + b_M, \qquad t_3(x_3) = a_M x_3 + b_M.$$

The added connector is indexed as $a = 5$ and is assigned its own linear parameters,

$$t_5(x_5) = a_5 x_5 + b_5.$$

The baseline coefficients are sampled on the discrete grid

$$a_U \in \{2, 6, 10\}, \qquad b_U \in \{0, 20, 40\}, \qquad a_M \in \{1, 2, 3\}, \qquad b_M \in \{40, 50, 60\},$$

together with the demand values

$$Q_{rs} \in \{2, 6, 10, 14, 18\}.$$

For each sampled baseline specification, the symbolic routine treats (a_5, b_5) as unknowns and solves for them so that a prescribed target difference in average travel time is achieved at the selected demand level. The target set is

$$\Delta ATT_{\text{target}} \in \{1, 0, -1\}.$$

Across all combinations, the symbolic stage evaluates $3 \times 3 \times 3 \times 3 \times 5 \times 3 = 1215$ instances. When a symbolic solution contains a single undetermined scalar degree of freedom, it is fixed to a concrete feasible value to obtain a fully specified parameter set; in practice this value equals 10 when admissible under the feasibility constraints and equals 1 otherwise.

Each derived parameter set is then cross-checked numerically by solving the before and after equilibria at the same (r, s) and the same Q_{rs} using the numerical stopping and restart settings stated above, and by verifying that the computed ΔATT matches the intended classification under the stated thresholds.

To operationalise the concept of equivalent cost families in a transparent and reproducible manner, three illustrative configurations are specified on the classical Braess diamond network. Their role is methodological: they provide fully defined inputs for the symbolic derivation protocol used to construct admissible families for the added connector, rather than functioning as an exhaustive parameter scan. In each configuration, the background network (links 1–4) and the demand level are fixed, while the added link $2 \to 3$ (link 5) is treated as the design degree of freedom and parameterised by an affine cost

$$t_5(x) = a_5 x + b_5. \tag{4.12}$$

For a given configuration, the symbolic routine derives a closed-form expression for the equilibrium performance difference

$$\Delta \equiv ATT^{(A)} - ATT^{(B)}, \tag{4.13}$$

as a function of (a_5, b_5) under equilibrium assignment. Imposing a prescribed target shift $\Delta = \Delta_{\text{target}}$ yields a one-parameter admissible set in the (a_5, b_5) plane. The derived equivalence relations and their numerical cross-checks are reported in Chapter 5; the present subsection specifies the configuration inputs and motivates their selection.

The three configurations are chosen to test the persistence of equivalent-family structure under progressively relaxed regularity of the background environment. The first

configuration reproduces a canonical Braess-style benchmark and therefore serves as a reference calibration for the symbolic derivation. The second configuration introduces small, structured perturbations in demand and background link parameters, intended to verify that the admissible family is not a knife-edge artefact of a single calibration but remains present under local variation. The third configuration deliberately increases heterogeneity across links by combining distinct slopes and intercepts, thereby producing an asymmetric background that better reflects plausible urban corridor differences in fixed delays and congestion sensitivity. Taken together, these three settings support a robustness argument: the existence of an admissible one-parameter family for the connector should persist from a classical benchmark to perturbed and heterogeneous environments.

The case-study inputs are defined as follows. Across all cases, background links $i = 1, \ldots, 4$ follow fixed affine costs $t_i(x) = a_i x + b_i$, while link 5 follows Eq. (4.12).

(i) Configuration 1:

$$Q = 6, \quad t_1(x) = 10x,\ t_2(x) = x + 50,\ t_3(x) = x + 50,\ t_4(x) = 10x, \quad \Delta_{\text{target}} = 9.$$

(ii) Configuration 2:

$$Q = 5, \quad t_1(x) = 12x,\ t_2(x) = x + 40,\ t_3(x) = x + 60,\ t_4(x) = 8x, \quad \Delta_{\text{target}} = 11.$$

(iii) Configuration 3:

$$Q = 4, \quad t_1(x) = 6x+5,\ t_2(x) = 2x+30,\ t_3(x) = x+40,\ t_4(x) = 12x, \quad \Delta_{\text{target}} = 6.$$

For compact reference, Table 4.3 summarises the inputs used in the symbolic demonstrations.

Table 4.3: Inputs for the three illustrative configurations used in the symbolic derivation of equivalent cost families on the Braess diamond network. The connector (link 5) is parameterised as $t_5(x) = a_5 x + b_5$ and solved under the equilibrium constraint $\Delta = \Delta_{\text{target}}$.

Case	Q	$t_1(x)$	$t_2(x)$	$t_3(x)$	$t_4(x)$	Δ_{target}
1	6	$10x$	$x + 50$	$x + 50$	$10x$	9
2	5	$12x$	$x + 40$	$x + 60$	$8x$	11
3	4	$6x + 5$	$2x + 30$	$x + 40$	$12x$	6

Figure 4.7 visualises the three configurations (demand levels and background costs) and highlights the added connector $t_5(x) = a_5 x + b_5$. The figure is placed in the methodology chapter because it defines the inputs for the symbolic-equivalence demonstrations; Chapter 5 then reports the resulting equivalence relations and their numerical equilibrium validations.

The symbolic solution for each configuration typically yields an affine one-parameter locus

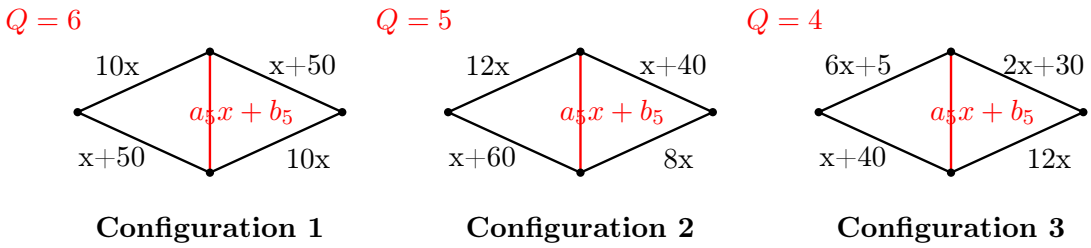


Figure 4.7: Three Braess-diamond configurations used as inputs for the symbolic derivation of equivalent cost families for the added link $2 \to 3$ (link 5). Background links are fixed per case, while the connector is parameterised as $t_5(x) = a_5x + b_5$.

in (a_5, b_5) space, together with an excluded value corresponding to a singular denominator in the closed-form expression. A set of sample points is then selected along the locus and passed to the numerical equilibrium solver to confirm that the realised equilibrium reproduces $\Delta = \Delta_{\text{target}}$ within the reporting tolerances adopted in Chapter 5. This symbolic-to-numerical cross-check ensures that the equivalence relations correspond to consistent equilibrium outcomes rather than purely algebraic artefacts.

4.6. Converting numerical cases into urban design

The computational framework described in the preceding sections produces systematic numerical evidence on equilibrium responses to network interventions. The purpose of this section is to define an explicit translation protocol that connects these numerical cases to interpretable planning and urban design narratives. The protocol does not change the equilibrium model. Instead, it specifies how network elements (nodes, directed links, and cost functions) and model outputs (equilibrium link flows and performance measures) are interpreted as stylised representations of vehicular movement and infrastructure interventions.

The translation proceeds by mapping

(i) network topology to a canonical spatial structure,

(ii) link cost functions to street performance characteristics, and

(iii) equilibrium flow changes to movement redistribution patterns under a controlled before/after comparison.

Throughout, the analysis concerns vehicular traffic assignment under DUE/SUE; pedestrian movement is not modelled in this thesis and is not used for paradox claims.

The first step is to define a consistent spatial interpretation of each network family. Networks are represented computationally as directed graphs with node identifiers and link lists. For design interpretation, each family is assigned a fixed schematic embedding that preserves adjacency and direction while providing a stable spatial reading across all experiments.

For the diamond benchmark, node 1 represents the origin and node 4 represents the destination, with intermediate nodes 2 and 3. The before network consists of two alternative two-link routes from 1 to 4. The after network adds the directed connector $2 \rightarrow 3$, which introduces an additional feasible route that combines links from both branches. In spatial terms, this corresponds to introducing an additional vehicular connection between two existing parallel route options within the same corridor system.

For ladder networks, the embedding corresponds to two parallel directed rails connecting an origin node to a destination node through repeated segments. In the configuration shown in Fig. 4.3, the origin is node 9 and the destination is node 10. The rail nodes on the left side are $\{1, 2, 3, 4\}$ and the rail nodes on the right side are $\{5, 6, 7, 8\}$. The after network adds cross-connectors between the two rails at specified segment indices (for example, $1 \leftrightarrow 5$, $2 \leftrightarrow 6$, $3 \leftrightarrow 7$, $4 \leftrightarrow 8$ in the illustrated case). This spatial reading avoids ambiguous labels such as "upper" and "lower" and keeps correspondence between the text and the node labels shown in the network figures.

A second step is to interpret link cost functions as stylised representations of link performance under increasing traffic volume. In the numerical experiments, each directed link a is assigned a travel-time function $t_a(x_a)$ that maps link flow x_a to travel time. Linear specifications represent cases where marginal delay increases proportionally with flow over the modelled range. BPR specifications represent cases where marginal delay increases nonlinearly as flow approaches capacity.

In design terms, these function families are interpreted as different congestion sensitivity regimes rather than as detailed geometric design models. The parameters are treated as modelling inputs that control relative attractiveness and congestion responsiveness across links. This interpretation is used only to support qualitative reading of mechanisms, while the numerical paradox definition remains entirely in terms of computed equilibrium performance differences between the before and after networks.

The neutrality convention is preserved throughout: links shared between the before and after networks retain identical cost functions and parameter values, so performance changes are attributable to the intervention link set and its explicitly specified parameters.

Each numerical case produces equilibrium link-flow vectors for both network states. These vectors provide the basis for movement-redistribution interpretation. For a given demand value Q_{rs}, let $\mathbf{x}^{\text{before}}$ and $\mathbf{x}^{\text{after}}$ denote equilibrium link flows in the before and after networks, expressed in the corresponding link-legend ordering.

Movement redistribution is described only through quantities that are directly available from the equilibrium outputs, such as changes in link flows on specific connectors or changes in the utilisation of the two rails in ladder networks. Statements about redistribution are therefore conditional: an intervention is described as shifting movement toward a subset of links only when the computed equilibrium satisfies $x_a^{\text{after}} > x_a^{\text{before}}$ for those links and the corresponding reductions occur on alternative links, under the same demand and numerical tolerance. The model does not represent intersection storage, signal spillover, or queue propagation across links. Consequently, the mechanism language in this thesis is restricted to equilibrium reallocation and congestion-related travel-time increases captured by the specified link performance functions, rather than to queue spillover or

spillback dynamics.

Demand scanning produces demand-dependent performance curves and paradox indicators. In design interpretation, different demand levels are treated as different loading conditions within the same corridor system. For a discrete grid of Q_{rs} values, "low", "intermediate", and "high" demand refer to positions within the scanned range. For example, in the classical reproduction with integer demand from 1 to 20, intermediate demand refers to values in the central portion of this range (for instance, values around 8 to 14), while low and high demand refer to values closer to the lower and upper ends of the grid, respectively. This definition is used only to describe where along the scanned demand axis a paradox interval occurs; the paradox criterion itself remains the sign and thresholded magnitude of $\Delta ATT(Q_{rs})$.

In the computational framework, an intervention is defined as adding a specified set of directed links to the before network to obtain the after network. In spatial terms, this corresponds to introducing additional vehicular connections between existing route options or adding cross-connectors between parallel rails in ladder networks. The intervention set is always stated at the level of node pairs, consistent with the node labels shown in the figures.

For ladder networks, the framework supports two controlled intervention protocols. First, bypass links can be added incrementally starting from the before network: the after network is generated by adding one cross-connector, then adding additional cross-connectors one by one, so that the marginal effect of each added bypass can be evaluated under identical cost parameters and demand grids. Second, subsets of bypass positions can be specified explicitly, allowing evaluation of any non-empty subset of the available cross-connector positions. Non-canonical added-link lists are computationally supported under the same numerical protocol, but they are documented separately so that the main analysis focuses on the canonical rail-aligned bypass interventions.

Stochastic user equilibrium introduces dispersion in route choice through the logit specification. In the experimental protocol, the dispersion parameter is evaluated at three fixed values, $\theta \in \{0.2, 0.8, 5\}$. Smaller values correspond to more dispersed route choice, while larger values approach deterministic cost-minimising behaviour. Since the SUE computation is deterministic for a given θ and network configuration, the resulting paradox classifications remain reproducible under the stated numerical tolerances and stopping rules.

The translation protocol also applies to analytically derived parameter families in the diamond benchmark. When symbolic derivations produce linear cost families expressed in the same linear form $t_a(x_a) = b_a + m_a x_a$, the resulting parameter sets can be interpreted using the same spatial embedding and the same neutrality convention. This enables consistent comparison between analytically characterised regimes and numerically observed demand-dependent paradox regions without introducing additional interpretive constructs beyond link performance and equilibrium redistribution.

The numerical outputs are translated into design-relevant evaluation statements by combining two elements:

(i) demand-dependent changes in average travel time through $\Delta\mathrm{ATT}(Q_{rs})$, and

(ii) the accompanying equilibrium link–flow changes that indicate which links gain or lose flow under the intervention.

This pairing prevents interpretation from relying on a single scalar indicator and keeps the spatial reading grounded in the computed equilibrium outcomes.

The translation protocol completes the methodological chain from equilibrium definitions to computational evidence and to spatially interpretable narratives. All interpretations remain conditional on the model class used in this thesis: static equilibrium assignment with specified link performance functions, controlled before/after comparisons, and demand scanning under fixed numerical tolerances.

4.7. Summary of the experimental plan

Sections 4.1–4.6 established a computational and interpretive architecture that links equilibrium theory, numerical experimentation, and spatial design reasoning. This section provides a concise evaluation of the experimental plan, clarifying its methodological adequacy for paradox investigation and identifying its principal strengths and limitations.

A primary achievement of the plan lies in its strict controlled-comparison standard. For every experiment, before and after networks share identical OD demand, cost functions on common links, solver parameters, and convergence criteria. The only intentional difference is the structural intervention itself and, where specified, the cost of newly added links. This neutrality convention is essential in paradox research, ensuring that observed performance differences arise from connectivity changes rather than hidden recalibration. The layered progression from the classical diamond benchmark to cost-function extensions, stochastic choice, and scalable ladder networks further strengthens internal validity by introducing one modelling degree of freedom at a time. As a result, changes in paradox behaviour can be attributed systematically to topology, cost convexity, or behavioural dispersion rather than uncontrolled interactions.

The plan also incorporates explicit measures to ensure numerical robustness. A common MSA-based backbone is used for both deterministic and stochastic equilibria, with identical solver tolerances, restart strategies, and convergence tests applied symmetrically to before and after networks. Paradox classification is based on average travel time differences with a small positive threshold, reducing the risk of misclassifying numerical noise as genuine paradox effects. Convergence diagnostics and conservative exclusion of unreliable demand points further reinforce the credibility of large-scale parameter scans. These conventions are especially important given that paradox detection relies on small differences between two equilibrium computations.

In terms of coverage, the experimental design spans three key dimensions relevant to the thesis objectives. Topologically, the plan progresses from a minimal benchmark to scalable ladder networks, enabling investigation of whether paradox mechanisms persist under repeated motifs and growing path sets. In cost specification, the use of per-link affine and BPR functions allows heterogeneous congestion regimes to be explored, distinguishing

structural paradox effects from artefacts of a particular functional form. Behaviourally, embedding stochastic user equilibrium with a tunable dispersion parameter allows systematic assessment of how realistic route-choice variability modulates paradox boundaries. Together, these dimensions provide a parameter space sufficiently rich to support both numerical atlas construction and later symbolic cross-checking.

A further strength of the plan is its emphasis on interpretability. Numerical outputs are structured not only as scalar paradox indicators but also as link-flow diagnostics aligned with consistent link legends. This ensures that paradox classifications can be traced back to concrete equilibrium re-routing mechanisms. Combined with the spatial embedding protocol introduced in Section 4.6, the plan guarantees that every numerical intervention corresponds to a recognisable design action such as adding connectors or increasing permeability. Consequently, later chapters can translate equilibrium outcomes into urban design reasoning without ad hoc reinterpretation.

Reproducibility is treated as an explicit design principle. Each experiment is defined by a configuration record specifying network generation, cost parameters, demand grids, equilibrium models, and solver options. Results are stored in structured scan tables, enabling regeneration of all figures and paradox maps. Path enumeration rules and admissibility criteria are fixed across before and after states and reported where relevant, preventing hidden biases from asymmetric path-set construction.

At the same time, the plan has deliberate scope limitations. The experiments are conducted in static assignment with a single OD pair, abstracting from dynamic congestion propagation, learning effects, and multi-OD interactions. Path sets are restricted to simple paths, and network geometries remain stylised rather than geographically embedded. These simplifications limit direct empirical prediction for specific cities, but they are consistent with the thesis aim: to isolate fundamental equilibrium mechanisms and translate them into general design-relevant principles rather than site-specific forecasts.

Overall, the experimental plan provides a balanced methodological foundation: sufficiently rigorous for systematic paradox investigation, sufficiently flexible to explore topology–cost–behaviour interactions, and sufficiently interpretable to support spatial translation. It therefore establishes the necessary basis for the numerical results in Chapter 5 and the symbolic and design-oriented analyses that follow in subsequent chapters.

5 | Results and Discussion

This chapter reports the core results of the thesis and develops a structured discussion of Braess-type paradoxes under equilibrium traffic assignment. Building on the mathematical formulation established in Chapter 3 and the computational framework introduced in Chapter 4, the analysis progresses from canonical benchmark networks to generalized topological families and systematic explorations of parameter space.

The presentation is organised progressively. Section 5.1 replicates the classical Braess' paradox on the diamond network under deterministic user equilibrium, providing a numerical benchmark that validates the modelling conventions and solution procedures adopted throughout the thesis. The subsequent sections extend the analysis along three complementary dimensions:

(i) sensitivity to demand level and cost-function parameters;

(ii) behavioural assumptions, through controlled comparisons between deterministic and stochastic user equilibrium; and

(iii) structural generalisation from the diamond network to ladder networks and multi-bypass configurations.

Across these experiments, the Braess' paradox is treated as a regime phenomenon rather than a binary attribute of a single network instance. Results are therefore interpreted in terms of demand-dependent paradox windows, equilibrium flow regimes, and regions of feasibility in cost-parameter space. Numerical findings are accompanied by internal consistency checks, diagnostic reporting, and robustness considerations to ensure that the observed performance shifts reflect genuine equilibrium responses rather than solver artefacts.

The chapter concludes by synthesising the empirical regularities that emerge across topology, demand, behavioural dispersion, and cost specification, with an emphasis on their implications for model-based network evaluation. The translation of these equilibrium patterns into spatial interpretation and design-oriented reasoning is developed separately in Chapter 6, and is not pursued within the scope of the present results chapter.

5.1. Classical Braess' paradox

This section reports the benchmark reproduction of the classical Braess' paradox using the computational reporting conventions established in Chapter 4. Deterministic user-equilibrium solutions are computed for demands $Q = 1, \ldots, 20$ under two configurations:

the baseline diamond network (before) and the same network with the additional connector $u_5 : 2 \to 3$ (after). For each demand, the outputs include the equilibrium average travel time in the before- and after-network, their difference $\Delta ATT = ATT^{(A)} - ATT^{(B)}$, and the corresponding equilibrium link-flow vectors. For readability, The specific calculation results are presented in Section A.1 and all reported numerical values are rounded to at most three decimal places.

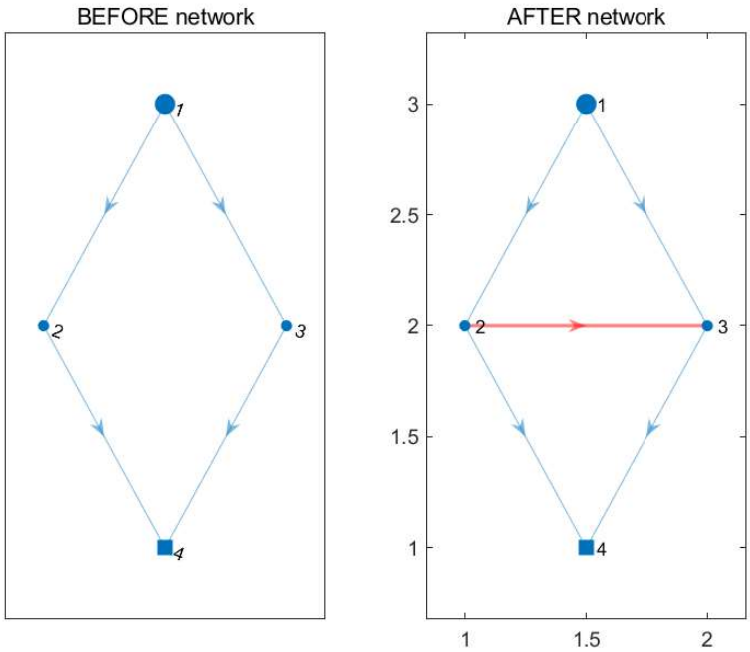


Figure 5.1: Diamond network (BEFORE vs AFTER). Node 1 is the origin, node 4 is the destination. The added connecting link is $u_5 : 2 \to 3$.

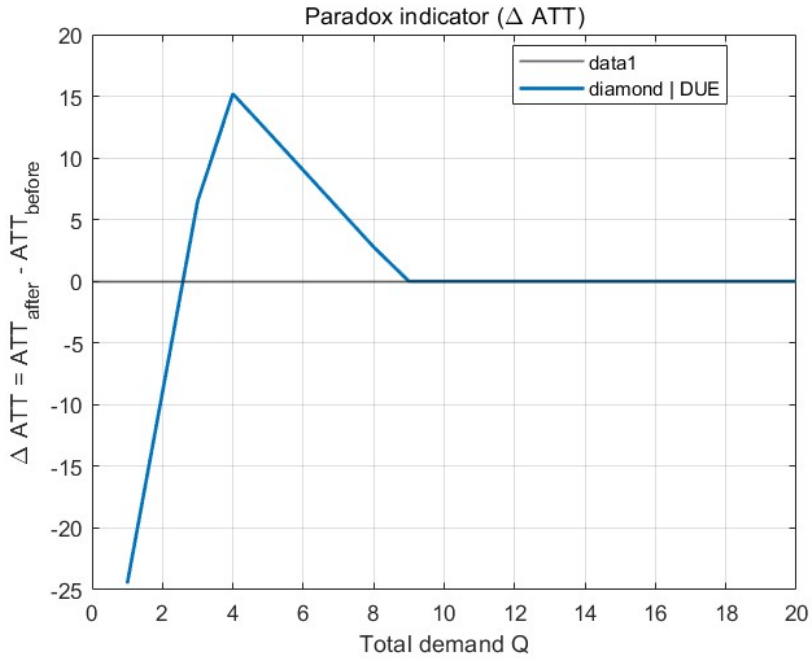


Figure 5.2: Paradox indicator as a function of demand: $\Delta ATT(Q) = ATT^{(A)}(Q) - ATT^{(B)}(Q)$.

Link flows are reported as vectors in the link order $[x_{u_1}\ x_{u_2}\ x_{u_4}\ x_{u_3}]$ for the before-network and $[x_{u_1}\ x_{u_2}\ x_{u_4}\ x_{u_3}\ x_{u_5}]$ for the after-network, consistent with the scan output conventions.

Three demand regimes can be identified from Table A.2. At low demand ($Q = 1$–2), the additional connector reduces equilibrium average travel time ($\Delta ATT < 0$). The link-flow vectors show that the after-network equilibrium concentrates flow on the connector route, with the direct links to the destination from node 2 and the direct link from the origin to node 3 carrying zero flow (e.g., [2 0 0 2 2] at $Q = 2$). In this regime, added connectivity is beneficial under equilibrium response.

A bounded paradox window occurs at intermediate demand ($Q = 3$–8), where the after-network exhibits strictly higher equilibrium average travel time ($\Delta ATT > 0$). The associated link-flow patterns indicate that the connector remains active ($x_{u_5} > 0$) while the equilibrium reallocates demand in a highly unbalanced manner: most flow is carried by the two links adjacent to the origin and destination (x_{u_1} and x_{u_3}), whereas the two alternative links (x_{u_2} and x_{u_4}) carry only a residual share. For instance, at $Q = 4$, $x_{u_1} \approx x_{u_3} \approx 3.69064$ while $x_{u_2} \approx x_{u_4} \approx 0.309363$, and the connector carries $x_{u_5} \approx 3.38127$. This redistribution is consistent with the paradox mechanism: the added connector attracts flow and reshapes the equilibrium so that congestion is concentrated on a subset of critical links, yielding a higher network-wide average travel time. Within the scanned range, the maximum deterioration is observed at $Q = 4$ ($\Delta ATT = 15.197$), after which paradox intensity declines as the connector flow decreases.

For higher demand ($Q \geq 9$), the two configurations become performance-equivalent

($\Delta ATT = 0$). The link-flow vectors show that the connector is no longer used ($x_{u_5} = 0$), and the remaining link flows coincide with the baseline symmetric split (e.g., [4.5 4.5 4.5 4.5 0] at $Q = 9$). In this regime, the topology change is endogenously neutralised by equilibrium route choice, so that the augmented network reduces to the same effective utilised subnetwork as the baseline configuration.

Overall, the benchmark results confirm that Braess-type deterioration is demand-contingent, and that the paradox window is characterised by a strictly positive connector flow together with a pronounced concentration of equilibrium traffic on a subset of links. This validated output format provides the reference point for the subsequent sections, where the same reporting conventions are used to compare broader network families and equilibrium regimes.

5.2. Demand and Cost Sensitivity of the Braess' Paradox

Section 5.1 established the classical benchmark and documented the demand-dependent paradox window under the reference specification. Building on that verified baseline, this section reports robustness scans under two nonlinear cost families that extend the analysis beyond the affine setting:

(i) strictly convex polynomial costs and

(ii) BPR-type costs.

The network definition, solver settings, and parameterisation conventions are those specified in Chapter 4; here the emphasis is placed exclusively on the computed equilibrium outcomes and on how the paradox indicator changes across cost families and demand regimes. For readability, The specific calculation results are presented in Sectione A.2 and all reported numerical values are rounded to at most three decimal places.

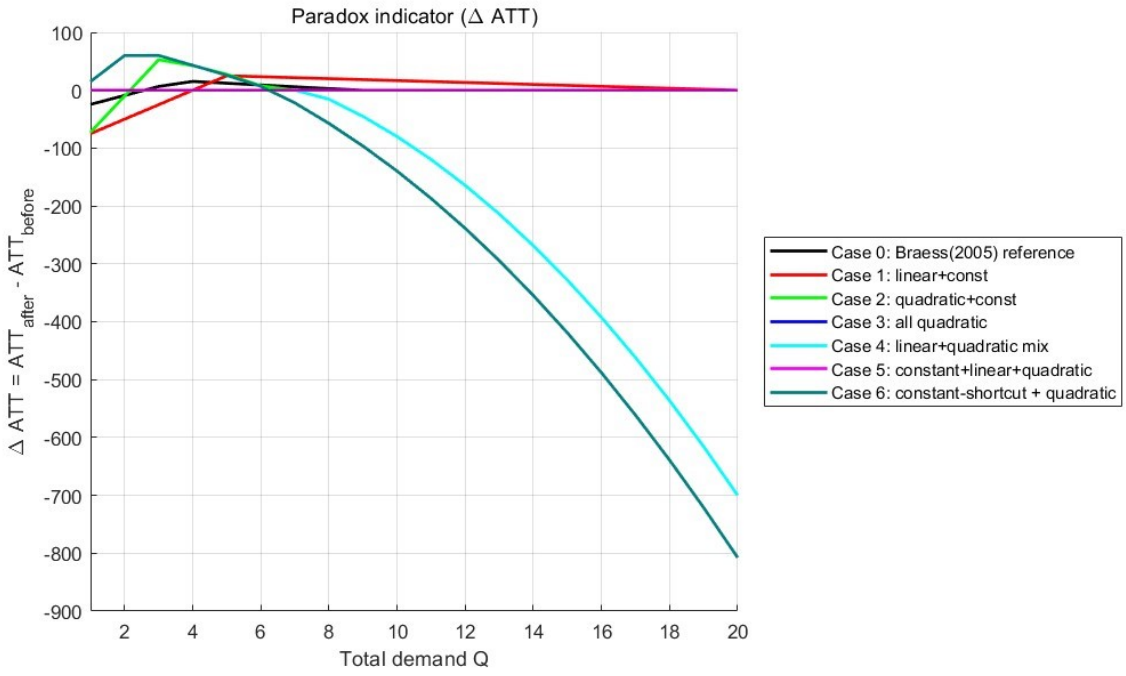


Figure 5.3: Multi-case comparison for convex polynomial specifications (Cases 0–6): the paradox indicator ΔATTfunctions of total demand Q.

The classical benchmark reported in Table A.2 provides the reference behaviour for interpreting the robustness scans under convex polynomial families. Figure 5.3 summarises the full demand sweep for all polynomial cases (including the benchmark), by jointly displaying

(i) the average travel time in the before network,

(ii) the average travel time in the after network,

(iii) the paradox indicator $\Delta ATT = ATT^{(A)} - ATT^{(B)}$, and

(iv) the corresponding total system cost difference $\Delta TSTT = TSTT^{(A)} - TSTT^{(B)}$.

Across cases, two general regularities can be read directly from Figure 5.3. First, the absolute growth rate of ATT with demand is strongly case-dependent: specifications that include quadratic terms generate substantially larger travel-time levels at moderate and high demand, which is visible in the separation of the ATT curves in the top panels. Second, the sign and magnitude of ΔATT are not invariant to the cost family: the paradox is therefore not purely topological, but contingent on how congestion externalities are encoded by the cost curvature.

In the benchmark case (Case 0), the paradox remains a bounded intermediate-demand phenomenon. As shown in Table 5.1, the after network yields higher average travel time than the before network for a restricted range of demands, while ΔATT returns to approximately zero once the shortcut becomes endogenously irrelevant at equilibrium. The

link-flow vectors corroborate this regime shift: within the paradox window the shortcut carries a non-negligible share of flow (for example, at the peak point the after-network vector reallocates flow toward the connector), whereas outside the window the shortcut flow collapses to (approximately) zero and the before/after equilibria coincide up to numerical tolerance.

Relative to the benchmark, Case 1 (linear+constant) preserves the qualitative Braess mechanism but alters its quantitative signature. The multi-case plot shows that the deviation between the before and after ATT curves becomes positive over a broader intermediate range, and the corresponding ΔATT curve remains above the zero line over a longer interval than in Case 0. This shift is consistent with the scan tables: at the onset of paradoxical behaviour, the equilibrium can become highly concentrated on the shortcut-enabled route set, reflected by a substantial shortcut flow and near-vanishing flow on one of the original branches. In other words, compared to the benchmark where the connector induces a partial redistribution, Case 1 exhibits a stronger reallocation toward the connector-containing alternatives, which sustains a positive ΔATT across a wider demand regime before the effect attenuates as demand increases and the equilibrium reverts toward the original branches.

By contrast, the quadratic-based families (Cases 2 and 3) suppress paradoxical outcomes under the scan criterion. In Figure 5.3, the corresponding ΔATT traces remain close to zero, indicating that adding the shortcut does not produce a systematic deterioration in equilibrium performance. The scan tables clarify the behavioural reason: shortcut flow is either negligible or rapidly driven toward zero as demand grows, so the after-network equilibrium effectively reproduces the before-network equilibrium. Under these convex penalties, the marginal congestion increase associated with concentrating flow on the connector-enabled composite route is sufficiently strong that the connector does not remain an equilibrium-attractive option in the regime where it would need to be used to generate the classical paradox.

Cases 4–6 provide a further contrast that is visible in the lower panels of Figure 5.3. For these mixed or strongly convex specifications, the plotted ΔATT and $\Delta TSTT$ curves can become markedly negative at higher demands, indicating that the shortcut improves (rather than worsens) equilibrium performance in the computed solutions. Interpreting these patterns in the same comparative framework as Case 0, the sign reversal means that added connectivity can act as a relief mechanism under certain convex cost landscapes: the extra link increases route-set flexibility and supports load-balancing that reduces the equilibrium cost level. At the same time, the substantial magnitudes observed in the system-cost panel are a direct consequence of convex growth: when ATT itself increases rapidly with Q, even modest relative changes in equilibrium allocation can produce large absolute differences in ATT and hence in $TSTT$.

Taken together, the robustness scan shows that the classical Braess pattern is not structurally guaranteed once cost curvature is modified. The benchmark paradox (Case 0) remains a mesoscopic effect confined to intermediate demand, while Case 1 demonstrates that mild modifications can preserve the paradox but shift its demand range and intensify shortcut-driven concentration. In contrast, stronger convexity (Cases 2–3) tends to neutralise the paradox by making the shortcut unattractive in equilibrium, and several mixed

convex families (Cases 4–6) can even reverse the sign of the performance change, turning the additional link into a net equilibrium improvement. Figure 5.3 provides a compact visual synthesis of these differences and supports the interpretation that the presence, extent, and intensity of Braess-type effects depend on the interaction between topology and the curvature distribution of link cost functions.

This subsection replaces affine costs with BPR-type congestion functions (as defined in Section 4.2) and repeats the before/after equilibrium comparison under deterministic user equilibrium. Two placements of nonlinearity are considered. Case A introduces BPR only on the added bypass while retaining the original functional form on the shared links, whereas Case B applies BPR uniformly to all links in both network states. The scan outputs are reported using the same indicators as Section 5.1: $ATT^{(B)}$, $ATT^{(A)}$, $\Delta ATT = ATT^{(A)} - ATT^{(B)}$, the paradox flag, and the equilibrium link-flow vectors.

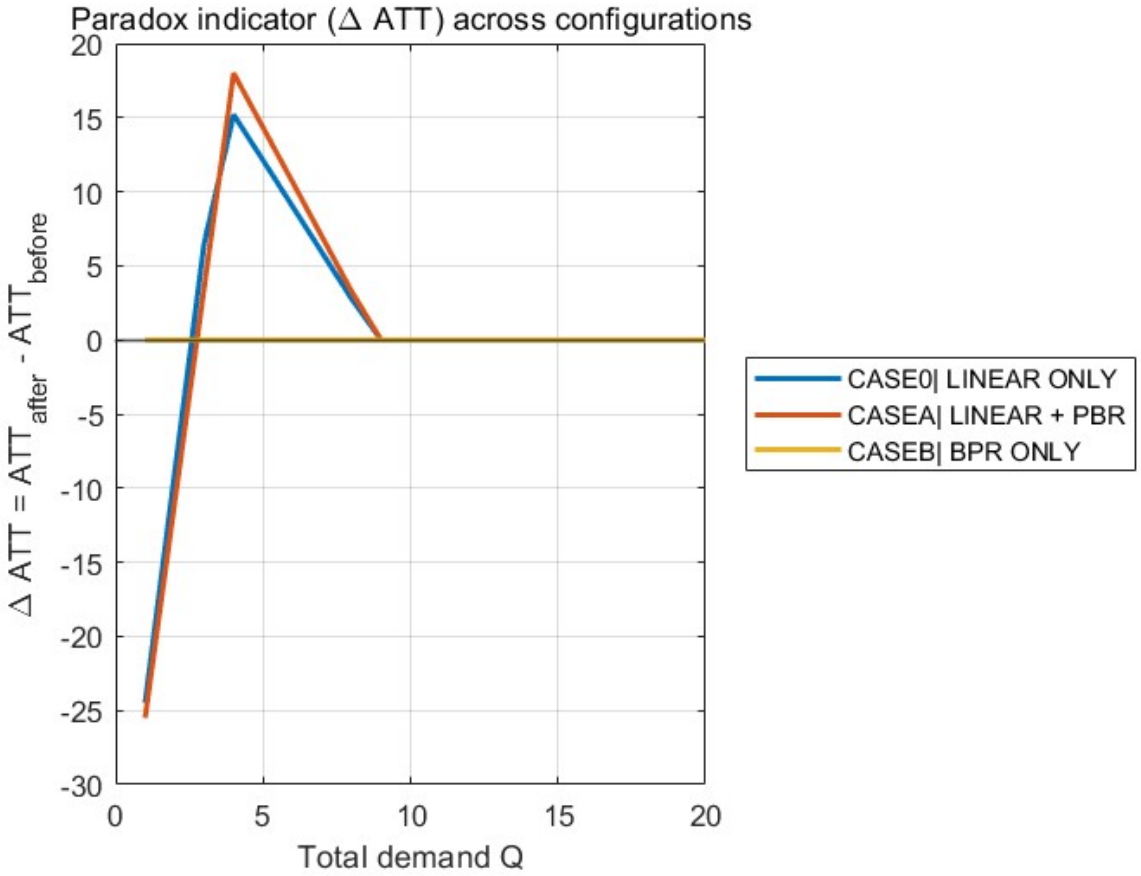


Figure 5.4: Multi-case comparison for BPR cost function:the paradox indicator ΔATT functions of total demand Q.

Case A: BPR on the added bypass only (mixed specification). Table A.9 reports the full scan for $Q = 1, \ldots, 20$. Consistent with the benchmark interpretation in Section 5.1, three demand regimes are observed:

(i) improvement at very low demand ($\Delta ATT < 0$),

(ii) a bounded paradox window where $\Delta ATT > 0$, and

(iii) a high-demand regime where $\Delta ATT = 0$ and the bypass becomes endogenously unused.

The link-flow vectors in Table A.9 clarify the mechanism behind the bounded paradox window. Within $Q = 3$–8, the after-network equilibrium assigns a strictly positive flow to the bypass ($x_5 > 0$), and the redistribution concentrates a large fraction of total demand onto the bypass-containing structure. At $Q = 4$, for instance, the after-network almost fully collapses onto the bypass-supported configuration: $x_1 \approx x_4 \approx 4$ and $x_5 \approx 4$, while x_2 and x_3 become nearly zero. Relative to the benchmark in Section 5.1, where x_2 and x_3 remain strictly positive in the paradox window, this near-degenerate concentration amplifies congestion on the two steeper shared links (the links that also appear on the bypass-supported route), raising the common experienced cost and producing the observed peak ΔATT. As demand increases toward $Q = 8$, bypass usage persists but declines (x_5 decreases from about 4 at $Q = 4$ to about 0.73 at $Q = 8$), indicating that the BPR curvature progressively erodes the bypass's relative attractiveness. For $Q \geq 9$, the bypass flow collapses exactly to zero ($x_5 = 0$) and the before/after link-flow vectors coincide, explaining why $\Delta ATT = 0$ and the paradox disappears.

Case B: BPR on all links (fully nonlinear specification). Table A.10 reports the full scan for the demand range used in the BPR experiment (Section 4.2). Across the entire range, $ATT^{(B)}$ and $ATT^{(A)}$ coincide and $\Delta ATT = 0$ (within the numerical resolution of the scan), hence no paradox is detected. The link-flow vectors show that the bypass is never used at equilibrium ($x_5 = 0$ for all scanned demands), so the after-network equilibrium is effectively identical to the before-network equilibrium.

The fully nonlinear scan provides a structurally different robustness outcome from the benchmark in Section 5.1. Because $x_5 = 0$ for every demand level in Table A.10, the added bypass never becomes part of the equilibrium support set, and the before/after equilibria coincide by construction. The redistribution observed as demand increases therefore takes place entirely within the original two-branch structure. This is visible in the evolution of the link-flow split: the equilibrium remains concentrated on one branch at low demand (e.g., $[Q\ Q\ 0\ 0\ 0]$ up to $Q = 1800$) and progressively shifts toward a more balanced allocation as congestion rises (e.g., at $Q = 6000$, the split becomes [3091.49 3091.49 2908.51 2908.51 0]). Crucially, however, this internal rebalancing does not create any before/after performance wedge, hence $\Delta ATT = 0$ and no paradox is detected.

The mixed BPR placement (Case A) preserves the benchmark qualitative structure but modifies the equilibrium redistribution intensity. First, the paradox window remains bounded and occurs over the same demand interval as the classical scan ($Q = 3$–8). Second, the peak deterioration is larger in Case A than in the classical benchmark. For example, at $Q = 4$ the classical scan reports a positive ΔATT with a strictly interior flow pattern on the shared links (both x_2 and x_3 remain positive), whereas Case A exhibits an almost complete collapse of the equilibrium onto the bypass-supported configuration with

$x_2 \approx x_3 \approx 0$ and $x_5 \approx Q$. This concentration increases the load on the two steeper shared links that jointly support the bypass-containing routes (here reflected by $x_1 \approx x_4 \approx Q$), raising the common equilibrium cost and increasing ΔATT relative to the benchmark. Finally, the paradox vanishes in Case A for $Q \geq 9$ for the same equilibrium reason as in the benchmark: the bypass becomes endogenously unused ($x_5 = 0$), and the after-network equilibrium collapses back to the before-network equilibrium.

By contrast, under uniform BPR nonlinearity (Case B) the bypass is never activated at equilibrium, so the intervention is neutral with respect to equilibrium performance over the entire scanned range. Relative to the classical benchmark, where the bypass is used over an intermediate regime and then abandoned, Case B suppresses bypass usage entirely and therefore suppresses paradox formation under the tested calibration. Taken together, the two scans show that introducing nonlinear congestion does not, by itself, determine paradox formation; rather, paradoxical worsening depends on whether the added link becomes part of the equilibrium support and on how its presence reshapes (or fails to reshape) the equilibrium link-flow pattern on the shared links.

5.3. DUE versus SUE: Behavioural Dispersion and the Paradox Window

Section 5.2 showed that, under deterministic user equilibrium (DUE), the classical diamond network exhibits a bounded paradox window: the added link is beneficial at very low demand, detrimental over an intermediate range, and becomes endogenously unused at sufficiently high demand so that before/after performance coincides. This section inserts the stochastic user equilibrium (SUE) scan outputs and explains how behavioural dispersion changes the onset, persistence, and disappearance of the paradox window, using the same indicator as in Section 5.1, $\Delta ATT = ATT_{\text{after}} - ATT_{\text{before}}$, and the same numerical zero rule adopted in the scan routine.For readability, The specific calculation results are presented in Sectione A.3 and all reported numerical values are rounded to at most three decimal places.

The DUE benchmark in Table A.2 reports improvement at $Q = 1, 2$ and a paradox window at $Q = 3$–8. A defining feature is the sharp upper cutoff: from $Q \geq 9$ the scan returns $\Delta ATT = 0$ and the after-network assigns exactly zero flow to the added link, so the after state collapses to the effective topology of the before network. This sharp deactivation provides the reference for interpreting SUE, where non-minimal routes can retain positive flow shares.

Tables A.11–A.13 report the SUE scans for three temperature settings ($\theta = 0.2, 0.8, 5$). Two regularities stand out when compared with the DUE benchmark.

First, the onset threshold is stable: in all three SUE scans the sign switch occurs at $Q \approx 3$, matching the DUE onset.

Second, the upper boundary expands systematically with dispersion. The paradox window

extends from $Q = 3$–8 (DUE) to $Q = 3$–9 ($\theta = 0.2$), to $Q = 3$–10 ($\theta = 0.8$), and to a long tail up to $Q = 18$ for $\theta = 5$ (with the tail end classified by the scan as non-paradox once ΔATT falls below the numerical zero threshold). This extension occurs because, under SUE, the added link is not abandoned discontinuously; instead its flow decays gradually with demand, so the after-network remains (slightly) different from the before-network over a longer demand range.

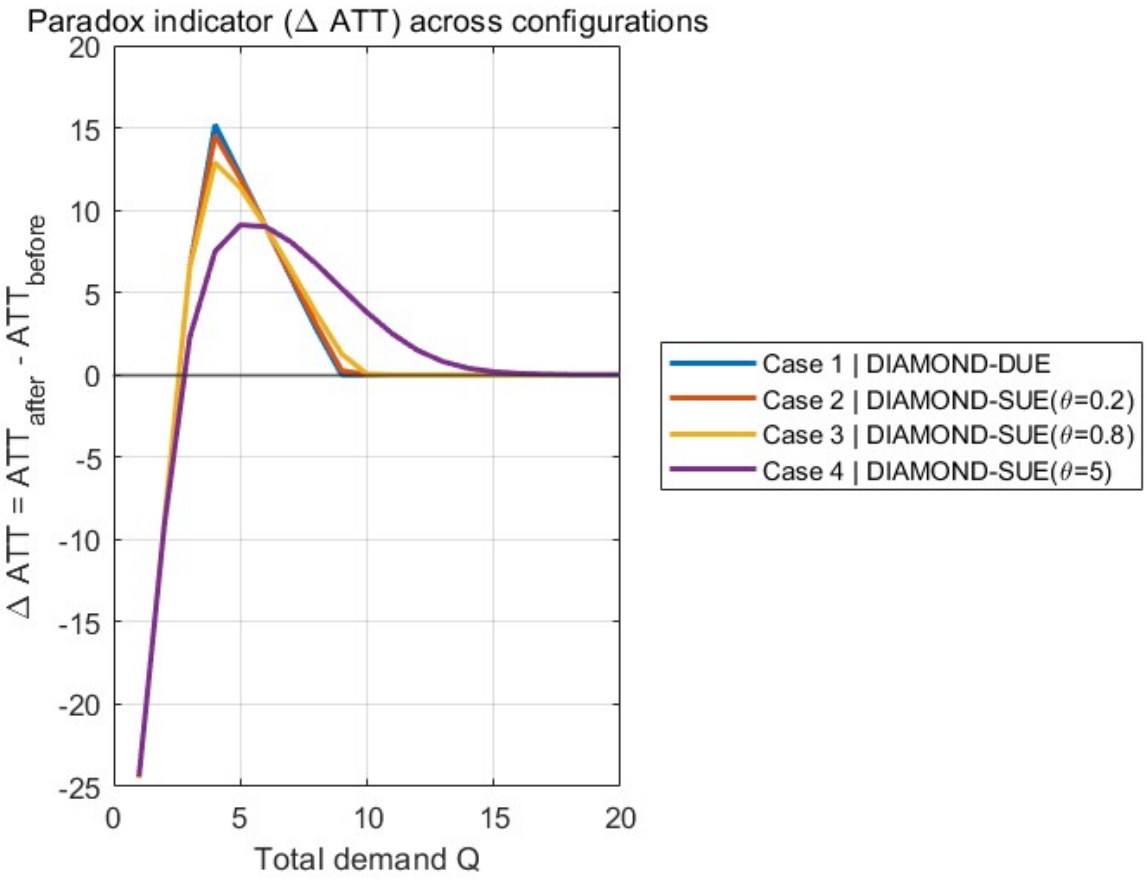


Figure 5.5: Multi-case comparison for DUE vs SUE: the paradox indicator ΔATT functions of total demand Q.

Table 5.1: Observed paradox windows under DUE and SUE (from scan outputs; temperature convention).

Assignment regime	Paradox window (scan)
DUE (Table 5.1)	$Q = 3$–8
SUE, $\theta = 0.2$	$Q = 3$–9
SUE, $\theta = 0.8$	$Q = 3$–10
SUE, $\theta = 5$	$Q = 3$–18 (tail truncated by numerical zero rule)

Behavioural dispersion changes not only where the paradox ends, but also how large it is across demand.

At the centre of the classical window, SUE remains paradoxical but the peak degradation is attenuated as θ increases. For example, at $Q = 4$ the DUE scan reports a large positive ΔATT (Table 5.1), while the SUE scans report smaller values: $\Delta ATT = 14.54$ ($\theta = 0.2$), 12.881 ($\theta = 0.8$), and 7.4944 ($\theta = 5$). This pattern is consistent with dispersion preventing extreme concentration of flow onto the bypass-supported structure, thereby reducing the maximum performance penalty even while keeping the paradox mechanism active.

Beyond the DUE cutoff, the contrast reverses: DUE returns exactly to $\Delta ATT = 0$ from $Q \geq 9$, whereas SUE retains a positive but shrinking ΔATT. At $Q = 10$, the scan reports $\Delta ATT \approx 0$ for $\theta = 0.2$ (effectively DUE-like), $\Delta ATT = 0.044887$ for $\theta = 0.8$, and a much larger $\Delta ATT = 3.7843$ for $\theta = 5$. The planning-relevant implication is that dispersion does not remove the paradox; it makes its disappearance gradual rather than abrupt.

Table 5.2: DUE vs SUE at representative demand levels (values read directly from the scan outputs).

Regime	Q	ATT_{before}	ATT_{after}	ΔATT
DUE (Table 5.1)	2	61.0	52.0	−9.0
DUE (Table 5.1)	6	83.0	92.0	9.0
DUE (Table 5.1)	10	105.0	105.0	0.0
SUE $\theta = 0.2$	2	61.0	52.0	−9.0
SUE $\theta = 0.2$	6	83.0	92.0	9.0
SUE $\theta = 0.2$	10	105.0	105.0	3.472×10^{-10}
SUE $\theta = 0.8$	2	61.0	52.0	−9.0
SUE $\theta = 0.8$	6	83.0	92.0	9.0
SUE $\theta = 0.8$	10	105.0	105.04	0.044887
SUE $\theta = 5$	2	61.0	51.843	−9.1571
SUE $\theta = 5$	6	83.0	92.0	9.0
SUE $\theta = 5$	10	105.0	108.78	3.7843

The link-flow vectors explain the expanded upper tail. Under DUE, at $Q \geq 9$ the after-network assigns exactly zero flow to the added link, so before/after coincide. Under SUE, the fifth component (added link flow) decays continuously with demand and remains positive for longer when θ is larger.

For example, at $Q = 10$ the scan reports an added-link flow of approximately 6.94×10^{-11} for $\theta = 0.2$ (numerically negligible), 8.97×10^{-3} for $\theta = 0.8$ (small but not zero), and 0.694 for $\theta = 5$ (substantial). This is why ΔATT is essentially zero for $\theta = 0.2$ at $Q = 10$ but remains clearly positive for $\theta = 5$.

At the far end of the $\theta = 5$ scan, the paradox does not disappear by a structural collapse of the after-network; rather, it becomes operationally invisible under the numerical zero rule. For instance, ΔATT is still positive at $Q = 19$ (0.0073041), but the scan classifies it as non-paradox because it falls below the adopted zero tolerance threshold.

Compared to the DUE benchmark (Table 5.1), SUE preserves the same qualitative three-regime logic (benefit at very low demand, worsening at intermediate demand, diminishing effect at high demand) but changes its geometry. The paradox onset remains anchored near $Q = 3$, suggesting that the emergence mechanism is structurally robust. However, as dispersion increases, the paradox peak is attenuated while the upper tail extends, because the added link is not abandoned abruptly. From a planning interpretation, this means that interventions can remain performance-relevant over a wider range of demand when travellers are heterogeneous or imperfectly cost-minimising, even if the strongest paradoxical penalty is reduced.

5.4. Ladder networks: structural generalisation of the Braess' paradox

The preceding results were established on the canonical four-node diamond network, which offers a minimal yet analytically transparent setting for isolating equilibrium effects. Assessing whether paradoxical performance degradation is a topological artefact or a structurally persistent property motivates an extension to ladder networks. A ladder network is formed by two parallel rails connected through intermediate rungs; additional cross-links (bypasses) act as local shortcuts. Because such shortcuts reshape the feasible r–s path set under the same demand scan and reporting conventions, ladder networks provide a controlled basis for evaluating how the paradox depends on bypass directionality, the adopted link-cost family, network length, and bypass placement. For readability, The specific calculation results are presented in Sectione A.4 and all reported numerical values are rounded to at most three decimal places.

In the classical Braess benchmark, the intermediate connection between the two central nodes is modelled as a single directed shortcut. A bidirectional representation is equally plausible in spatial settings (e.g., two-way streets or paired turn movements), and it expands the admissible path set by enabling additional simple routes that exploit the reverse direction of the bypass. To isolate the role of directionality, the diamond benchmark is therefore repeated with a bidirectional bypass ($b \leftrightarrow c$), while preserving the same demand scan ($Q = 1, \ldots, 20$) and the same equilibrium reporting structure adopted in Section 5.1.

Figure 5.6 contrasts the $\Delta ATT(Q)$ profiles for the one-way and two-way bypass variants (the one-way curve corresponds to the baseline reported in Section 5.1). Both representations exhibit the canonical paradox signature: a beneficial regime at low demand ($\Delta ATT < 0$) followed by a detrimental regime at intermediate demand ($\Delta ATT > 0$). The bidirectional bypass, however, changes the high-demand regime. Table A.14 shows that ΔATT becomes strictly negative for $Q = 14, \ldots, 20$ under the bidirectional bypass, indicating that the after-network regains an advantage at high demand. This contrasts

with the one-way benchmark, where the shortcut becomes endogenously unused at sufficiently high demand and ΔATT collapses toward zero. The difference is structural: the reverse-direction bypass introduces additional balancing routes, allowing flow redistribution without forcing all travellers through the same congested intermediate segment. As a result, the shortcut does not need to be abandoned to restore equilibrium consistency at high Q.

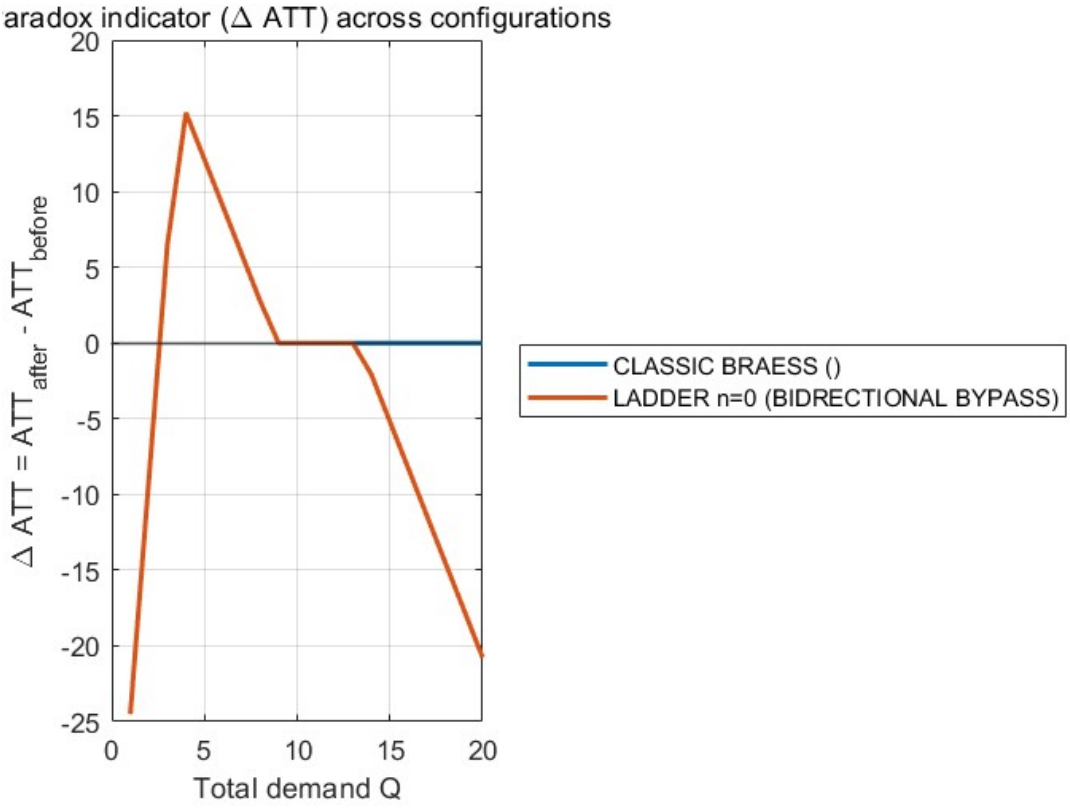


Figure 5.6: Classical diamond network: comparison between one-way and two-way bypass representations (DUE, ΔATT versus Q).

The ladder experiments adopt a cost specification designed for systematic scaling with network size (the ladder preset introduced in Chapter 4), which differs from the classical diamond cost family used in Section 5.1. To make the effect of this modelling choice explicit, the degenerate ladder case ($n = 0$) is used as a controlled bridge between the diamond topology and the ladder construction rules. Under $n = 0$, the node set reduces to a diamond-like structure, while the ladder conventions (including the bidirectional bypass convention used for ladder scans) and the ladder cost preset are retained.

Figure 5.7 and Table A.15 report the resulting $\Delta ATT(Q)$ profile. A direct comparison with the classical benchmark tables (Table 5.1 for the one-way bypass; Table A.14 for the bidirectional bypass) highlights two systematic effects of the cost-family shift.

First, the absolute travel-time scale changes substantially: for example, at $Q = 1$ the ladder-preset baseline yields $ATT^{(B)} = 10.5$ (Table A.15), whereas the classical benchmark yields $ATT^{(B)} = 55.5$ (Table A.14 and Table 5.1). This reflects the different free-flow and congestion parameterisation embedded in the ladder preset.

Second, the paradox window shifts. Under the ladder preset, the degenerate ladder exhibits a longer detrimental regime: $\Delta ATT > 0$ holds from $Q = 6$ through $Q = 15$ (Table A.15), and the high-demand regime converges to $\Delta ATT = 0$ from $Q = 16$ onward. Relative to the bidirectional-diamond benchmark, where ΔATT becomes negative at high demand (Table A.14), the ladder preset promotes an "endogenous deactivation" pattern in which the bypass becomes effectively irrelevant at high Q rather than persistently beneficial. This distinction is important for interpretation: directionality effects in the diamond (Section 5.4) act primarily through path-set expansion, whereas the ladder preset modifies the congestion trade-offs that determine whether the bypass remains attractive as demand grows.

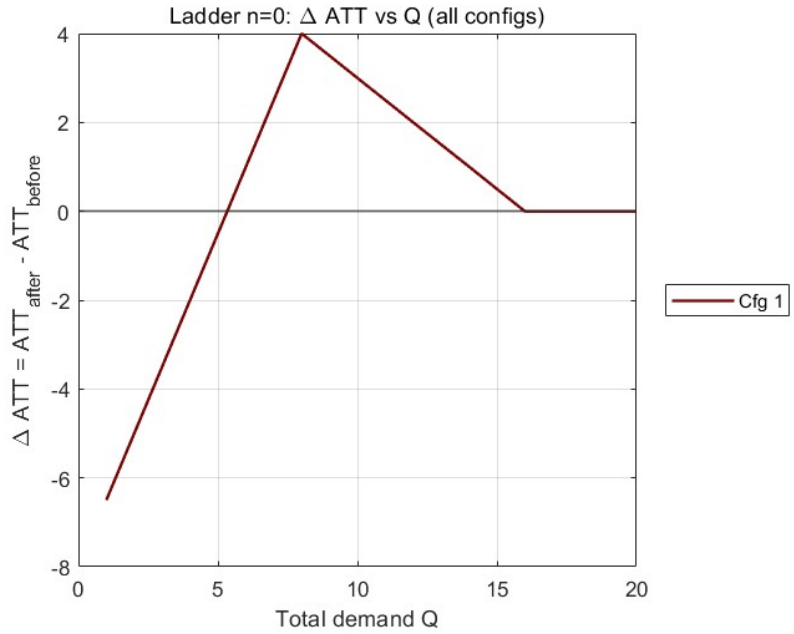


Figure 5.7: Ladder $n = 0$: ΔATT versus Q (all configurations; here only one configuration exists).

For $n \geq 2$, multiple bypass positions exist along the ladder, and the after-network is generated by activating different non-empty subsets of these positions. The scan outputs therefore contain multiple configurations indexed by `ConfigID`, each associated with a bypass count $k =$ `NumBypassPos` and a set of bypass locations `BypassPos`. Figures 5.8–5.9 report $\Delta ATT(Q)$ curves for all configurations at $n = 2$ and $n = 3$, while Figure 5.10 overlays all curves across $n \in \{0, 2, 3\}$.

Two cross-n commonalities are immediately visible when comparing Figures 5.7–5.10.

First, the low-demand regime is frequently beneficial ($\Delta ATT \leq 0$), consistent with a shortcut acting as a genuine path-length reduction when congestion externalities are weak. Second, the high-demand regime tends to converge toward $\Delta ATT \approx 0$, matching the deactivation pattern already observed in the $n = 0$ bridge case (Table A.15). This indicates that, under the ladder preset, increasing demand eventually removes the marginal equilibrium relevance of bypasses for most configurations.

At the same time, increasing ladder length expands the configuration space and introduces richer intermediate-demand behaviour. For $n = 2$, seven configurations exist (all non-empty subsets of three bypass positions); for $n = 3$, fifteen configurations exist (all non-empty subsets of four positions). A salient empirical feature in Figures 5.8–5.9 is the presence of exactly coincident curves: multiple configurations produce indistinguishable $\Delta ATT(Q)$ sequences over the entire scan. This overlap is a substantive equilibrium-equivalence signal rather than a plotting artefact, because it holds pointwise for every scanned $Q = 1, \ldots, 20$.

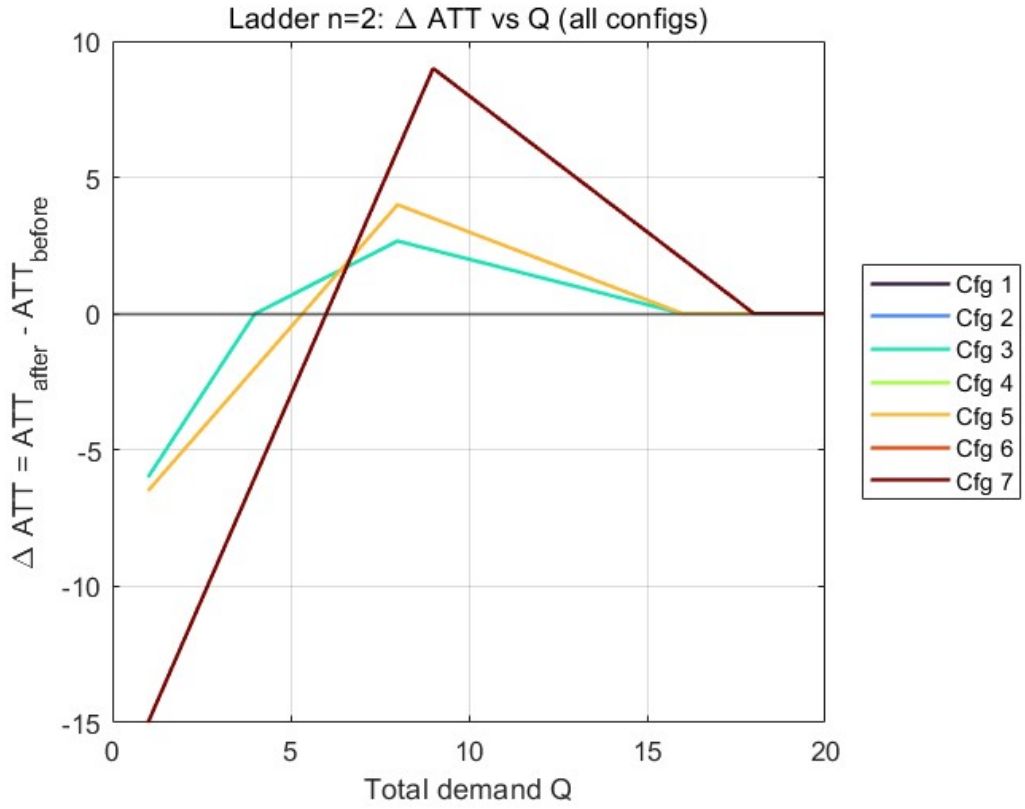


Figure 5.8: Ladder $n = 2$: ΔATT versus Q across all configurations (each curve corresponds to one `ConfigID`).

For $n = 2$, the seven nominal configurations collapse into three distinct performance outcomes. Specifically, the following curves are exactly coincident in Figure 5.8:

(i) Cfg 1 and Cfg 3;

(ii) Cfg 2, Cfg 4, Cfg 6, and Cfg 7;

(iii) Cfg 5 forms the remaining distinct curve.

Hence, only three distinct $\Delta ATT(Q)$ trajectories emerge under this cost preset and demand range.

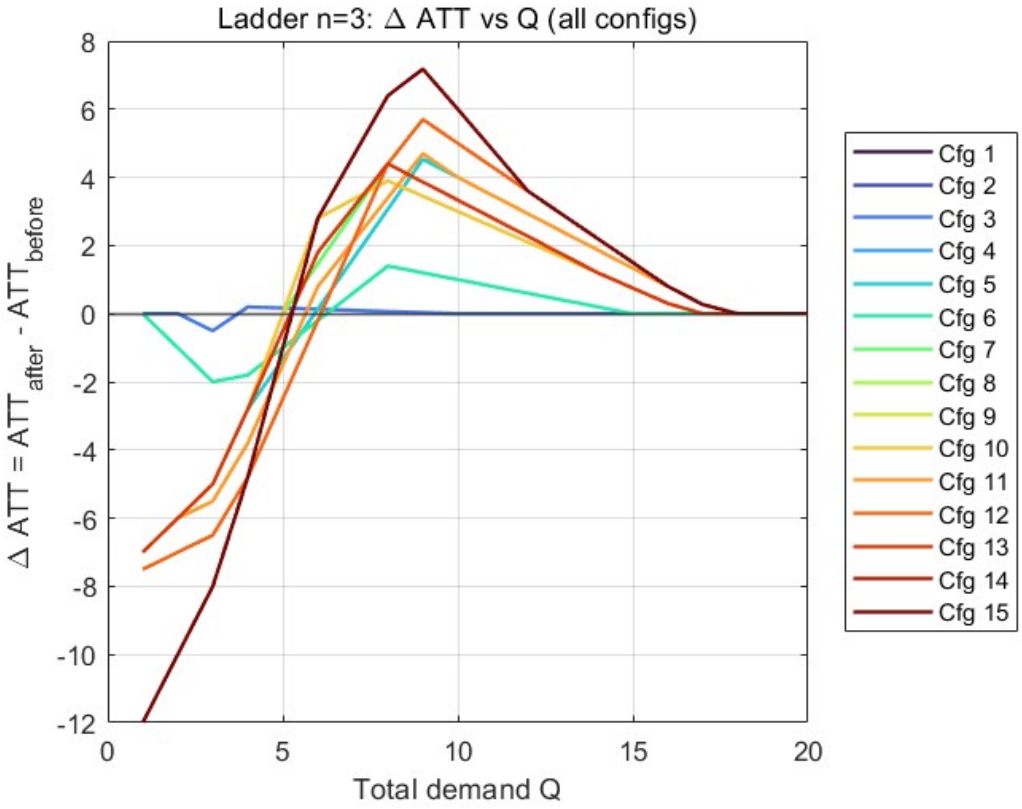


Figure 5.9: Ladder $n = 3$: ΔATT versus Q across all configurations (each curve corresponds to one `ConfigID`).

For $n = 3$, fifteen configurations collapse into ten distinct $\Delta ATT(Q)$ curves. In Figure 5.9, the following configuration pairs are perfectly coincident over $Q = 1, \ldots, 20$:

(i) Cfg 1 with Cfg 6,

(ii) Cfg 2 with Cfg 5,

(iii) Cfg 8 with Cfg 11,

(iv) Cfg 9 with Cfg 12,

(v) Cfg 14 with Cfg 15.

(vi) All other configurations (Cfg 3, Cfg 4, Cfg 7, Cfg 10, Cfg 13) produce distinct curves.

As a result, the number of distinct aggregate performance outcomes is smaller than the number of nominal after-network topologies.

Exact overlap of $\Delta ATT(Q)$ curves arises for two main reasons. First, under the present cost assignment the ladder admits structural symmetries, so that different bypass placements can generate the same set of equilibrium-relevant route alternatives and therefore the same equilibrium flows. This is most evident for $n = 2$, where placing the bypass at the two symmetric endpoint positions produces identical $\Delta ATT(Q)$ trajectories. Second, and more importantly for the larger ladder, additional bypasses can be redundant at equilibrium. Once an effective switching opportunity is already available, adding further bypass links may remain unused throughout the scan, or may reproduce the same switching pattern, so that link flows and ATT are unchanged despite a strictly richer topology. The overlap pairs identified for $n = 3$ illustrate this directly: in each case, augmenting an already active bypass set by one additional position leaves $\Delta ATT(Q)$ unchanged, indicating that the extra connectivity is equilibrium-irrelevant over the scanned demand range.

To disentangle shortcut density from precise placement, configurations are grouped by bypass count k. For both ladder sizes, the mean paradox intensity within each (n, k) group attains its maximum at the same intermediate demand level ($Q = 9$) and increases monotonically with k. For $n = 2$, the peak mean ΔATT at $Q = 9$ increases from 4.555 ($k = 1$) to 7.166 ($k = 2$) and 8.999 ($k = 3$). For $n = 3$, the corresponding peak mean values at $Q = 9$ are smaller for sparse shortcuts but grow similarly with density, from 2.304 ($k = 1$) to 3.907 ($k = 2$), 5.360 ($k = 3$), and 7.180 ($k = 4$). Thus, denser shortcut provision systematically amplifies the strongest paradox regime under this preset, even though the high-demand deactivation pattern persists.

Selected demand levels further clarify how density and placement interact. At a moderate demand level ($Q = 6$), $n = 2$ shows weak and highly placement-sensitive effects: the paradox can vanish entirely at higher shortcut density (for $k = 3$, ΔATT is identically zero), indicating complete bypass deactivation at this demand. By contrast, for $n = 3$ at $Q = 6$ the mean ΔATT increases with k, yet the within-group range can extend below zero for $k \in \{1, 2, 3\}$, showing that even at fixed density some placements remain beneficial while others are detrimental. Only the fully saturated case ($k = 4$) yields a single outcome at $Q = 6$.

At a higher demand level inside the paradox window ($Q = 15$), paradox incidence becomes nearly universal for both ladder sizes and the mean ΔATT increases with shortcut density. Moreover, placement-driven variability generally shrinks as k increases: at larger k, added bypasses increasingly fill the available switching opportunities, making the equilibrium response less sensitive to the exact placement pattern.

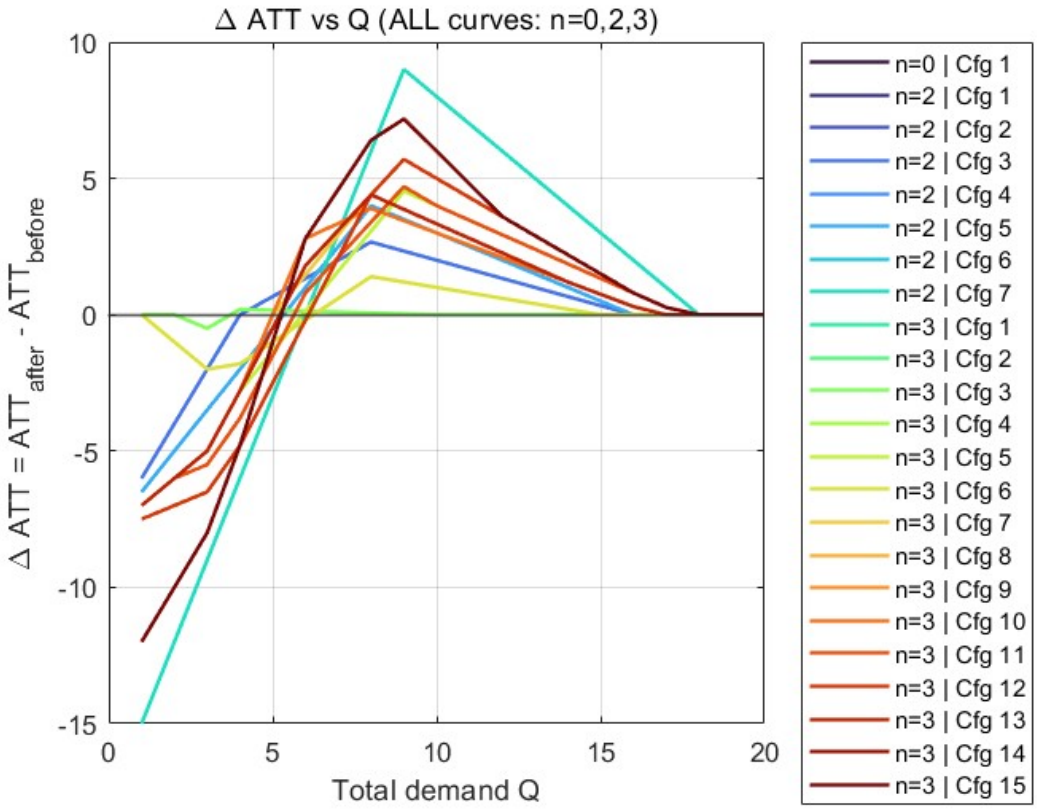


Figure 5.10: ΔATT versus Q with all curves overlaid for $n \in \{0, 2, 3\}$ (legend indicates n and `ConfigID`).

Taken together, Figures 5.8–5.10 show that enlarging the ladder from $n = 2$ to $n = 3$ greatly increases the number of admissible bypass subsets, but the resulting equilibrium responses remain highly structured. Across both sizes,

(i) the qualitative sign pattern persists (often $\Delta ATT \leq 0$ at low Q, a positive intermediate window, and $\Delta ATT \approx 0$ at high Q);

(ii) the strongest paradox effects concentrate around the same intermediate demand level (here $Q \approx 9$); and

(iii) many nominal interventions are either symmetry-related or redundant at equilibrium, producing exactly coincident $\Delta ATT(Q)$ curves.

The coexistence of exact overlap classes (equilibrium-equivalent interventions) and within-k sign variability at moderate demand (placement-sensitive interventions) shows that, even under a fixed cost specification, ladder networks contain both performance-irrelevant and performance-critical bypasses within the same size class.

5.5. Symbolic structure of Braess' paradox conditions and parameter maps

While Sections 5.2–5.5 established the existence and disappearance of the Braess' paradox through numerical equilibrium assignment, a deeper understanding requires an explicit characterization of the parameter space in which the paradox can occur. This section therefore shifts from numerical scanning to a symbolic and semi-analytic perspective, combining closed-form equilibrium conditions with large-scale parameter atlas computations and visualization.

The objective of this section is threefold:

(i) to derive symbolic relationships between the shortcut cost parameter b_5 and other network parameters under deterministic equilibrium assumptions;

(ii) to verify these symbolic boundaries against numerical equilibrium solvers; and

(iii) to visualize the resulting feasible regions in parameter space, following the request to explicitly plot the results under fixed normalization conditions ($x = 1$, $z = 1$).

We consider the classical diamond network with four nodes and five directed links. Let the cost functions on the four outer links be affine,

$$t_a(x_a) = a_a x_a + b_a \tag{5.1}$$

For the diamond network, links are grouped so that upper and lower outer links share identical (a_a, b_a), while the central shortcut link has constant cost b_5. The total demand is denoted by D.

Under deterministic user equilibrium (DUE), all used paths must have equal and minimal travel cost. In the before network (without the shortcut), symmetry implies that the flow splits evenly across the two outer routes. In the after network, the shortcut may attract flow depending on the relative magnitudes of b_5 and the marginal costs on the outer links.

The paradox occurs if and only if

$$ATT^{(A)}(Q) > ATT^{(B)}(Q) \tag{5.2}$$

subject to the equilibrium flow conditions. These inequalities can be rewritten as algebraic constraints involving b_5, D, and the affine cost parameters (a_U, b_U, a_M, b_M).

To systematically explore this space, a symbolic parameter atlas was constructed using the batch solver framework. For each sampled tuple (D, a_U, b_U, a_M, b_M), the admissible interval of b_5 satisfying the paradox inequalities was computed analytically and then cross-validated using the numerical solver described in Chapter 3.

The atlas results confirm that, for fixed outer-link parameters, the admissible b_5 region is typically an interval whose endpoints vary smoothly with demand and slope parameters.

Importantly, the existence of such an interval is not guaranteed: for certain parameter combinations the interval collapses, implying that no paradox can occur regardless of the shortcut cost.

Following the request to explicitly visualize the symbolic results, the atlas was projected onto two-dimensional parameter maps by fixing normalization parameters $x = 1$ and $z = 1$. Figures 5.11–5.16 illustrate representative projections.

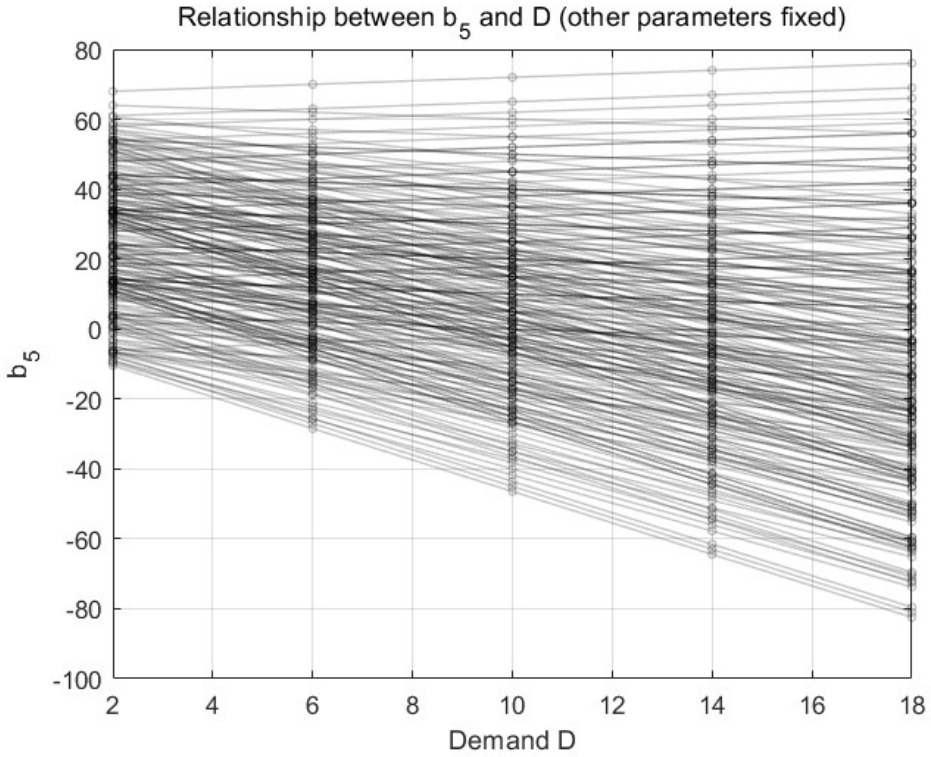


Figure 5.11: Relationship between shortcut cost b_5 and demand D with other parameters fixed. Each polyline corresponds to a feasible symbolic solution from the atlas.

Figure 5.11 shows that, for fixed slopes and constants, the admissible b_5 decreases monotonically with demand. This directly explains the empirically observed paradox window: as demand increases, the shortcut must be sufficiently cheap to remain attractive, but not so cheap as to eliminate congestion redistribution.

As shown in Figure 5.12, increasing the slope a_U systematically shifts the admissible b_5 interval downward. This reflects the fact that stronger congestion sensitivity on the outer links amplifies the incentive to use the shortcut, thereby requiring tighter conditions for paradox emergence.

Figure 5.13 demonstrates an approximately linear dependence between b_5 and b_U. Raising the baseline travel time on the outer links relaxes the paradox condition, allowing a higher shortcut cost to still induce paradoxical behavior.

The dependence on a_M (Figure 5.14) exhibits greater dispersion, indicating a secondary

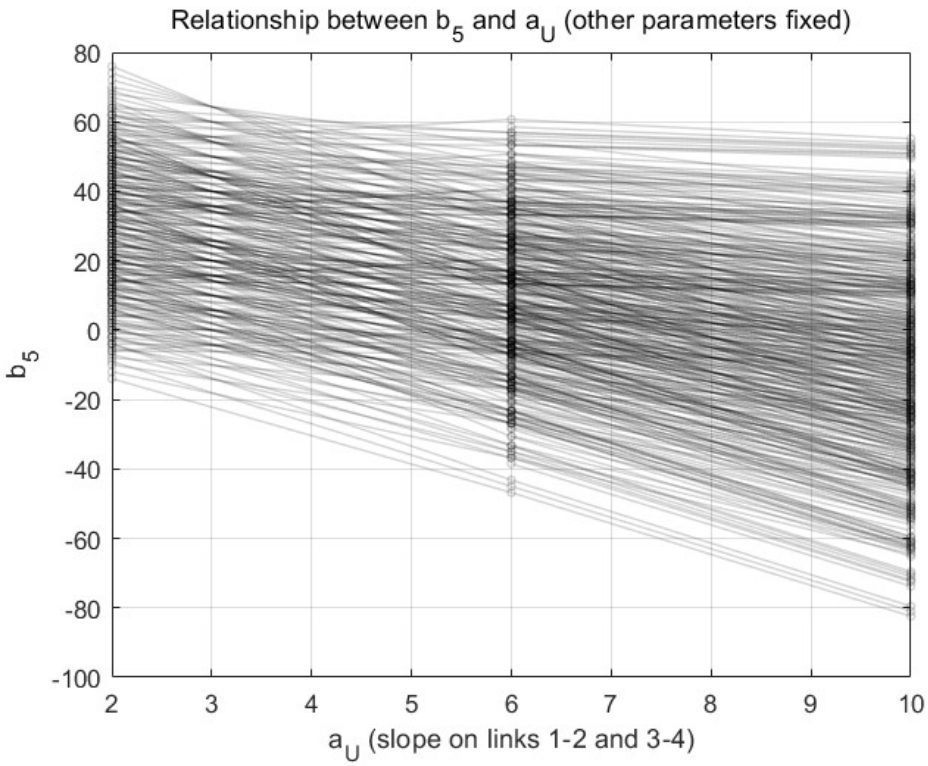


Figure 5.12: Relationship between b_5 and slope a_U on the outer links.

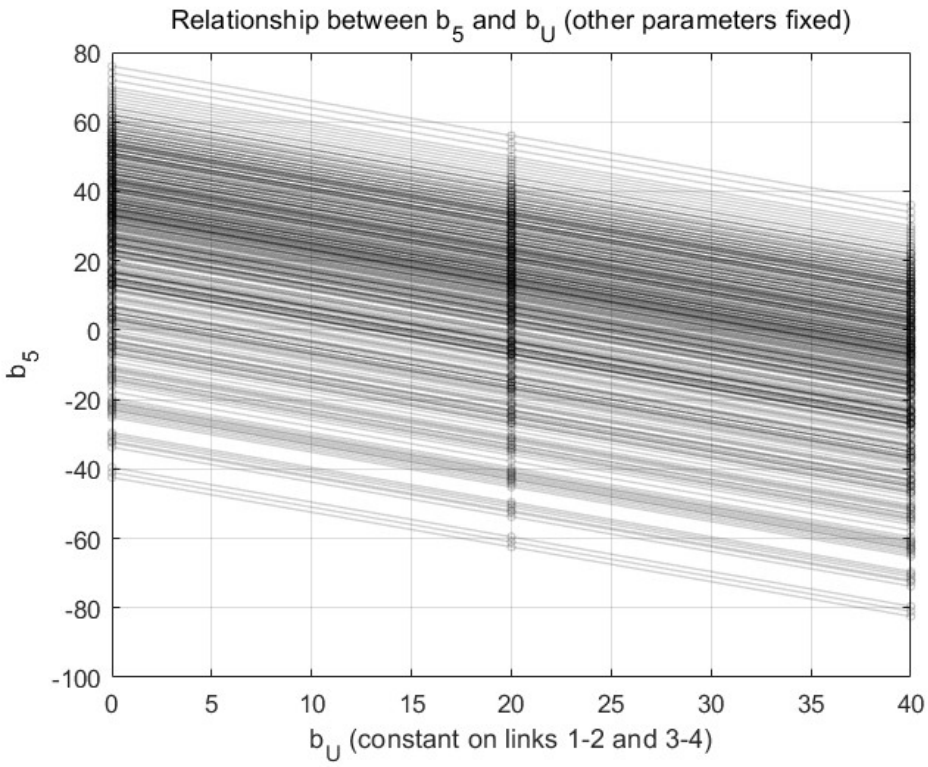


Figure 5.13: Relationship between b_5 and constant term b_U on the outer links.

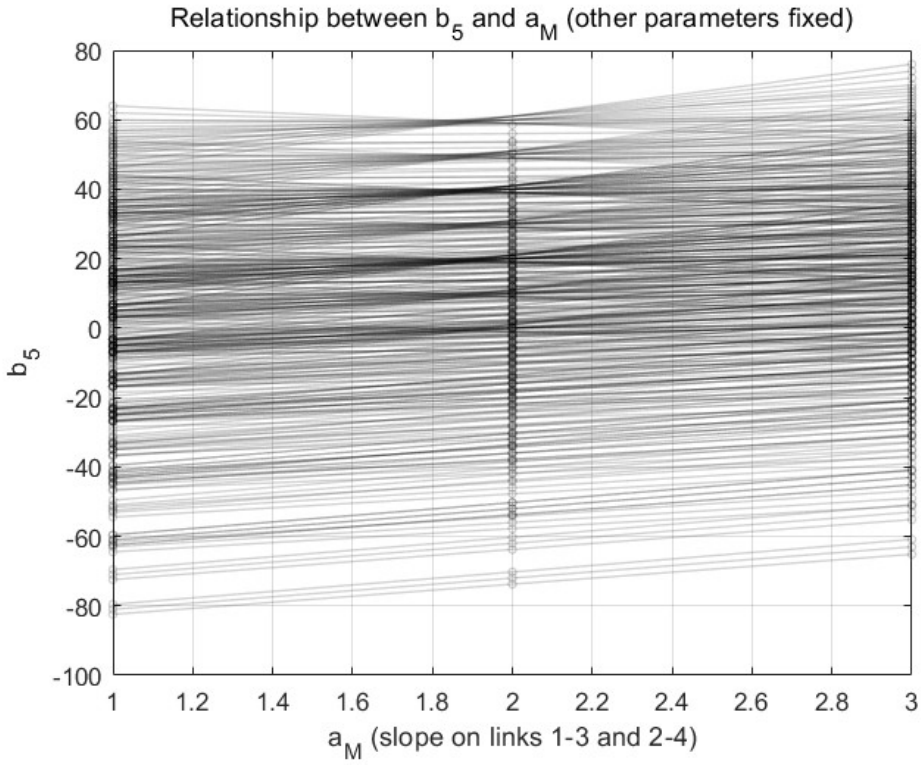


Figure 5.14: Relationship between b_5 and slope a_M on the middle links.

but non-negligible role. While the dominant effect stems from the outer-link slopes, variations in a_M modulate the width of the paradox region.

Finally, Figure 5.15 shows that increasing b_M shifts the feasible b_5 interval upward in a nearly parallel fashion, consistent with the additive structure of the path costs.

Figure 5.16 confirms that Δt_{target} primarily controls the width rather than the location of the admissible b_5 interval, providing a useful tuning parameter for symbolic exploration.

Taken together, these visualizations demonstrate that the Braess' paradox is not a discrete anomaly but a structured phenomenon embedded in a high-dimensional parameter space. For the diamond network under affine costs, the paradox region forms a family of slanted bands whose position and thickness are governed by demand and marginal congestion sensitivities.

This symbolic perspective explains several numerical observations from earlier sections: the existence of finite paradox windows, their disappearance at high demand, and their sensitivity to cost curvature. Moreover, it provides a principled foundation for extending the analysis to generalized networks and nonlinear cost functions, as explored in the next section.

Beyond the atlas-style symbolic characterization of paradox-feasible regions, three explicit symbolic demonstrations are reported here to establish equivalent cost families for the added connector $2 \to 3$ (link 5). In each case, the baseline network and demand are

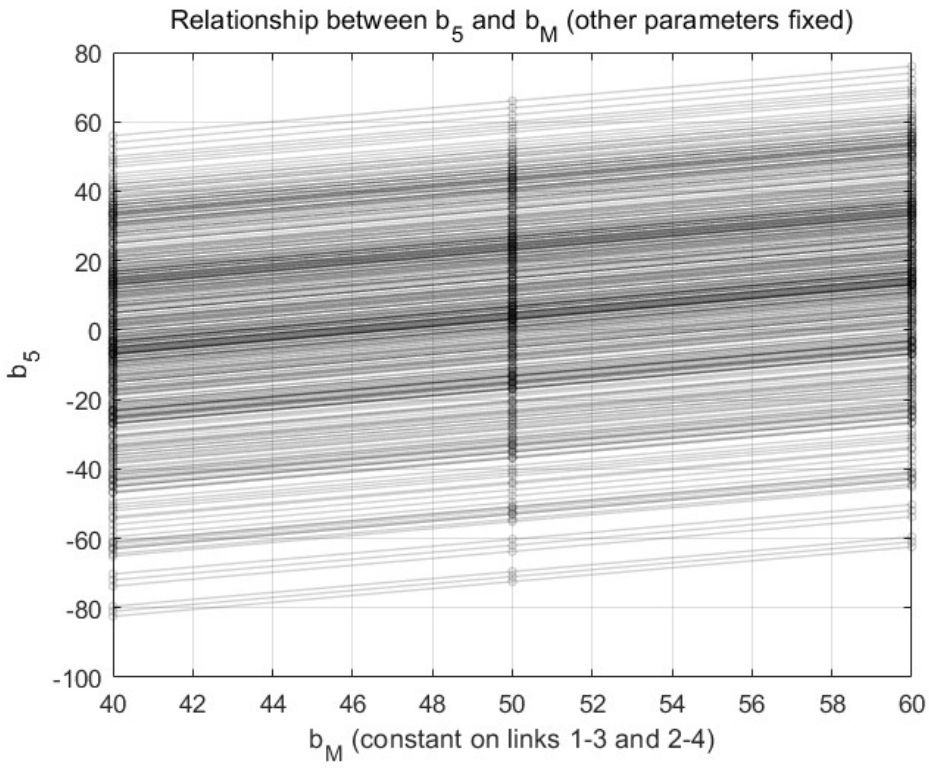


Figure 5.15: Relationship between b_5 and constant term b_M on the middle links.

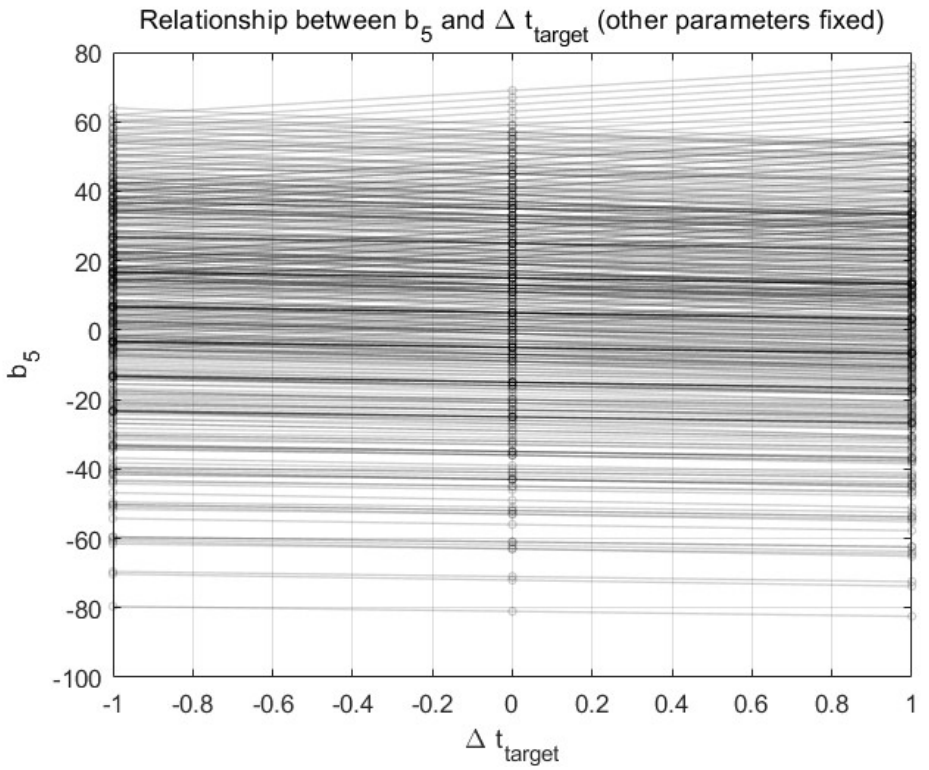


Figure 5.16: Relationship between b_5 and target travel-time difference Δt_{target}.

fixed, the added link is parameterized as $t_5(x) = a_5 x + b_5$, and a prescribed target shift in equilibrium performance is enforced. The result is not a unique pair (a_5, b_5), but a continuous one-parameter family.

Configuration 1. The first configuration fixes demand and baseline link parameters and prescribes a target performance shift $\Delta ATT_{\text{target}}$. The symbolic derivation yields a closed-form difference function

$$\Delta ATT(a_5, b_5) = -\frac{9(b_5 - 23)}{2a_5 + 11},$$

and enforcing $\Delta ATT = \Delta ATT_{\text{target}}$ produces the equivalent-family constraint

$$b_5 = 12 - 2a_5, \qquad a_5 \neq -\frac{11}{2}.$$

Table 5.3 reports sampled admissible pairs along this family (all numerically validated to reproduce the prescribed equilibrium shift).

Table 5.3: Equivalent cost-family samples for Configuration 1 (all satisfy $b_5 = 12 - 2a_5$ and reproduce the prescribed $\Delta ATT_{\text{target}}$).

#	a_5	b_5
1	0.00	12.00
2	0.25	11.50
3	0.50	11.00
4	0.75	10.50
5	1.00	10.00
6	1.25	9.50
7	1.50	9.00
8	1.75	8.50
9	2.00	8.00
10	2.25	7.50

Configuration 2. Under an altered baseline environment, the symbolic derivation yields

$$\Delta ATT(a_5, b_5) = -\frac{95(22b_5 - 585)}{22(22a_5 + 117)},$$

and enforcing the prescribed target shift gives

$$b_5 = \frac{27261}{2090} - \frac{242}{95}a_5, \qquad a_5 \neq -\frac{117}{22}.$$

Sampled admissible pairs are listed in Table 5.4.

Table 5.4: Equivalent cost-family samples for Configuration 2 (all satisfy the affine equivalence relation and reproduce the prescribed $\Delta ATT_{\text{target}}$).

#	a_5	b_5
1	0.00	13.0440
2	0.25	12.4070
3	0.50	11.7700
4	0.75	11.1330
5	1.00	10.4960
6	1.25	9.8593
7	1.50	9.2225
8	1.75	8.5856
9	2.00	7.9488

Configuration 3. A further baseline variation yields

$$\Delta ATT(a_5, b_5) = -\frac{10(b_5 - 20)}{3a_5 + 14},$$

and enforcing the prescribed target shift produces

$$b_5 = \frac{58}{5} - \frac{9}{5}a_5, \qquad a_5 \neq -\frac{14}{3}.$$

Table 5.5 reports validated samples.

Table 5.5: Equivalent cost-family samples for Configuration 3 (all satisfy the affine equivalence relation and reproduce the prescribed $\Delta ATT_{\text{target}}$).

#	a_5	b_5
1	0.00	11.60
2	0.25	11.15
3	0.50	10.70
4	0.75	10.25
5	1.00	9.80
6	1.25	9.35
7	1.50	8.90
8	1.75	8.45
9	2.00	8.00
10	2.25	7.55

Synthesis. Across all three configurations, the symbolic derivations produce one-parameter admissible families in (a_5, b_5) space, and the numerical equilibrium solver confirms that sampled points along each family reproduce the same prescribed equilibrium performance

shift. This provides direct evidence that equilibrium constraints leave a nontrivial design degree of freedom in the realization of the added connector, supporting the spatial non-uniqueness interpretation developed in Chapter 6.

5.6. Results evaluation and discussion

The preceding sections of Chapter 5 developed the thesis results in a progressive manner: first reproducing the canonical Braess' paradox in the diamond network under deterministic user equilibrium, then extending the analysis to demand and cost sensitivity, behavioural dispersion under SUE, structural generalisation to ladder networks, and finally to symbolic parameter-space characterisation and atlas-style maps. This section evaluates these results as a coherent body of evidence and synthesises the main interpretive implications. The discussion is organised around four themes:

(i) internal consistency and credibility of the numerical evidence;

(ii) what the results imply about the mechanism and robustness of the paradox across modelling choices;

(iii) how numerical and symbolic perspectives complement each other; and

(iv) what the findings suggest for network evaluation and planning-oriented interpretation.

A first point of evaluation concerns whether the reported paradox patterns can be trusted as equilibrium phenomena rather than numerical artefacts. Multiple internal consistency checks support this credibility. In the benchmark reproduction (Section 5.1), equilibrium conditions are satisfied in both the before and after networks, with equal travel times on used paths and no unilateral incentive to deviate. Moreover, reported performance measures satisfy the identity $\mathrm{TSTT} = Q \cdot ATT$ exactly in the benchmark tables, indicating that path flows, link flows, and performance calculations are numerically coherent. Similar consistency holds across the demand scan outputs, where paradox classification is based on paired before–after solves at identical demand and identical link performance specifications for common links.

Beyond scalar performance measures, the experiment outputs preserve link-flow diagnostics (e.g. shortcut usage u_5 in the diamond and bypass usage patterns in ladders). These diagnostics are crucial for credibility because they allow mechanism-based interpretation: paradox classification is supported by a tangible redistribution of equilibrium flows, rather than by a marginal difference in aggregated travel time alone. The presence of clear regime transitions—especially the endogenous extinction of shortcut flow at high demand under DUE—provides an additional robustness indicator. Numerical noise typically produces irregular, unstable sign changes; in contrast, the results display structured windows and consistent thresholds aligned with equilibrium logic.

Finally, the numerical design choices implemented in Chapter 4 (common solver backbone, identical tolerances for before/after, and a small positive threshold for paradox detection)

provide methodological assurance that the sign of ΔATT is not an artefact of inconsistent convergence. Therefore, the evidence base is sufficiently reliable to support interpretation in terms of equilibrium mechanism and comparative statics.

A central finding across the numerical results is that the Braess' paradox is not a binary property of a network, but a demand-conditioned phenomenon. In the classical diamond with linear costs, the paradox emerges only within a bounded intermediate range of demand: it is absent at low demand, present over a well-defined window, and absent again at high demand. This pattern is not incidental; it reflects three distinct congestion regimes.

At low demand, congestion externalities are weak, so a new connector genuinely expands feasible routing without creating strong negative feedback. In this regime the after-network can dominate the before-network in average travel time, and the shortcut is beneficial. As demand increases into an intermediate range, congestion becomes strong enough that route-choice incentives interact nonlinearly with network structure. The added connector induces a self-reinforcing rerouting pattern: travellers exploit the locally attractive composite route, increasing congestion on critical links and raising the equilibrium cost on all used paths. This produces a positive ΔATT and defines the paradox window.

At sufficiently high demand, a qualitatively different mechanism dominates: the shortcut becomes endogenously unattractive. Under DUE, travellers abandon the connector entirely, restoring an equilibrium that is effectively identical to that of the before network. This produces $\Delta ATT \approx 0$ and marks an upper threshold $Q_{\max}$ beyond which the paradox disappears. In this sense, the paradox is mesoscopic: it does not arise in the free-flow limit, and it does not persist under extreme congestion where the connector is self-limiting. Instead, it is confined to a range where externalities are strong but not yet overwhelming.

This regime structure is methodologically important because it motivates atlas-style reporting: a single demand point cannot characterise the paradox. The results suggest that any planning interpretation of a network intervention must be conditional on demand scenarios (e.g. peak versus off-peak, future growth versus current conditions), rather than being framed as universally beneficial or harmful.

A second evaluative theme is the dependence of paradox behaviour on cost-function form. The cost sensitivity experiments indicate that paradox presence and magnitude are not purely topological; they depend strongly on how marginal delay grows with flow. Linear and mildly convex specifications preserve paradox behaviour at intermediate demand, but strongly convex specifications (e.g. all-quadratic cases) can suppress paradox by penalising shortcut usage and preventing the equilibrium from concentrating flow on the paradox-inducing composite route.

The BPR experiments reinforce this interpretation. When nonlinearity is introduced only on the added bypass (mixed linear–BPR), paradox behaviour can still occur, and the overall pattern remains demand-window-like: beneficial at low demand, paradoxical at

intermediate demand, and neutral at higher demand when bypass usage decays. However, when BPR is applied uniformly across all links under the reported parameter set, the paradox is not observed in the scanned range. This contrast highlights that the relevant question is not whether costs are nonlinear, but whether the added link becomes sufficiently attractive relative to the surrounding network so as to induce a detrimental equilibrium reallocation. Uniform nonlinearity can flatten relative advantages and prevent the bypass from dominating route-choice incentives, whereas a low free-flow bypass embedded in a network with different marginal sensitivities can recreate the classical mechanism.

From an evaluation perspective, these findings support two claims. First, paradox mechanisms are structurally robust across multiple cost families, but their existence is not guaranteed; it is conditional on relative performance parameterisation. Second, paradox mapping must treat cost curvature and link heterogeneity as first-class variables. This justifies the per-link parameterisation architecture adopted in Chapter 4 and motivates the later symbolic atlas work: paradox is best understood as a region in parameter space, not a single-case curiosity.

The DUE versus SUE comparison provides a behavioural evaluation of paradox robustness. Across all reported temperature values θ, the onset threshold of the paradox remains relatively stable (around $Q \approx 3$ in the reported scans). This indicates that the emergence of paradox at intermediate demand is primarily structural: once congestion reaches a certain level, the added connector can trigger inefficient equilibrium reallocation regardless of whether route choice is perfectly deterministic.

Where dispersion matters most is in the persistence and termination of the paradox. Under DUE, the bypass is abandoned sharply beyond an upper threshold, collapsing the after equilibrium back to the before equilibrium. Under SUE, bypass usage decays gradually rather than discontinuously. Because logit choice assigns non-zero probability to non-minimal paths, the bypass can retain positive flow share even when it is not strictly optimal, delaying complete deactivation. As a consequence, the paradox window extends into higher demand levels under higher dispersion, often appearing as a long tail of small but positive ΔATT values that only vanish once they fall below the numerical zero tolerance.

This is a substantive conceptual implication: behavioural realism does not necessarily eliminate paradox, but it can broaden the operational range over which an added link affects system performance. In planning terms, this suggests that information heterogeneity, navigation systems, and preference diversity may not prevent paradoxical degradation; instead, they can spread its influence across a wider range of conditions, though possibly with reduced intensity near the tail.

The ladder experiments evaluate paradox robustness under topology scaling. The key conclusion is that paradoxical behaviour is not confined to minimal networks. Even with a single bypass in a ladder, the paradox persists as a local-to-global effect: adding a connector at one rung can worsen OD-level performance. This supports the thesis position

that Braess-type phenomena can be relevant to corridor-like or grid-like structures where multiple alternative routes exist.

When multiple bypasses are added, the results suggest two additional structural effects. First, increasing shortcut density tends to amplify paradox intensity, with peak mean ΔATT rising as bypass count increases. Second, configuration-specific variability (placement effects) diminishes as bypass density increases: the paradox becomes less dependent on where a single bypass is inserted and more dependent on the overall presence of many cross-links. This resembles a transition from a placement-sensitive regime (sparse interventions) to a structurally dominated regime (dense connectivity), which is analytically plausible because once many connectors exist, equilibrium flow patterns are constrained by the overall coupled structure rather than by one localized detail.

From the standpoint of methodological evaluation, ladder results justify the thesis emphasis on scalable families and systematic enumeration. They also motivate the shift to atlas-style summaries: as topology grows, the number of possible interventions grows combinatorially, and interpretability depends on aggregated measures (peak paradox magnitude, paradox rate, or interval length) paired with representative diagnostic cases.

Section 5.5 introduced a symbolic perspective by deriving algebraic constraints for paradox feasibility in parameter space and constructing an atlas validated against numerical solver outputs. This complements the numerical results in an essential way. Numerical scans reveal what happens across specific parameter grids, but they do not by themselves explain why a paradox window appears, why it disappears, or how boundaries move with parameters. Symbolic conditions provide this explanatory layer by showing that paradox feasibility typically forms an interval (or band) in the shortcut parameter b_5 conditional on demand and marginal cost coefficients.

In interpretive terms, the atlas perspective unifies multiple earlier findings: bounded paradox windows in demand correspond to shifting feasibility bands; cost sensitivity corresponds to shifts and collapses of these bands under changes in slopes and baselines; and BPR suppression can be interpreted as parameter settings that prevent the feasibility region from existing in the scanned range. Even when full symbolic derivations are feasible only for simplified affine settings, the atlas functions as an interpretive scaffold: it clarifies that paradox is a structured geometric object in parameter space, not an anomalous point.

Synthesising the results, several planning-relevant implications emerge. First, connectivity interventions cannot be evaluated solely by local intuition; equilibrium reallocation can make an apparently beneficial shortcut globally harmful under intermediate congestion. Second, paradox risk is scenario-dependent: it may be absent in low-demand conditions yet emerge under peak or growth scenarios. Third, design and policy levers that alter relative performance (e.g. capacity, speed, priority, or pricing) can shift or eliminate paradox regions, suggesting that paradox risk can be managed rather than treated as an inevitability. Fourth, behavioural dispersion affects how sharply paradox risk appears and disappears, implying that real-world heterogeneity may extend the range of conditions in

which a connector influences outcomes.

At the same time, the results also clarify the scope of claims. The experiments are static and single-OD; therefore, the findings should be interpreted as mechanism-based evidence and comparative statics, not as direct forecasts for a specific urban network without additional modelling layers. Nevertheless, the structured patterns observed across topology, costs, and behaviour provide a robust analytical basis for anticipating unintended consequences and motivate the thesis' later transition toward spatial interpretation and design guidelines.

Overall, the results of Chapter 5 satisfy the thesis methodological objectives: they reproduce a canonical benchmark, demonstrate structured demand and parameter dependence, verify robustness across behavioural assumptions, generalise mechanisms to scalable topologies, and connect numerical patterns to symbolic parameter-space structure. This integrated evidence base provides a defensible platform for the subsequent chapters, where the focus shifts from documenting paradox regions to interpreting them as families of equivalent cost structures and, ultimately, as spatial interventions with design meaning.

6 | From Equivalent Cost Families to Spatial Interpretation

The preceding chapters have developed the analytical core of this thesis. Chapters 3 and 4 established the equilibrium formulation and numerical solution framework for deterministic and stochastic traffic assignment. Chapter 5 then extended classical Braess-type analysis by systematically exploring how equilibrium outcomes vary across network topologies, congestion functions, and behavioral assumptions. Through numerical scanning and symbolic derivation, the thesis demonstrated that paradoxical equilibrium behavior is neither confined to a single stylized network nor dependent on a unique numerical parameter setting. Instead, it is embedded within continuous families of cost specifications and structural interventions that yield identical equilibrium outcomes.

Up to this point, the discussion has remained largely within the internal language of transportation network theory: links, nodes, cost functions, equilibrium flows, and performance measures. While this abstraction is essential for theoretical rigor, it raises a fundamental question for planning and design disciplines: how do these abstract network results relate to real spatial networks, street layouts, and urban interventions? A mathematical network is not merely a graph; in practice, it is a spatial artifact shaped by geometry, regulation, hierarchy, and political decision-making. Every modification of a link cost function corresponds, in physical terms, to some combination of street length, speed limit, lane allocation, intersection control, or modal priority. Every modification of topology corresponds to opening, closing, filtering, or reconfiguring streets. If equilibrium theory predicts that certain cost and topology manipulations can generate paradoxical outcomes, the relevance of these findings depends on how they map onto real spatial design actions.

This mapping problem has long constituted a gap between transportation science and urban design. Traffic assignment models are routinely employed in planning practice to forecast flows and evaluate infrastructure projects, yet their outputs are often treated as technical predictions rather than as elements of a broader spatial reasoning process. Urban design, conversely, routinely engages with questions of street connectivity, permeability, hierarchy, and spatial quality, yet rarely incorporates equilibrium-based network analysis in a systematic way. The Braess' paradox literature illustrates this divide particularly clearly: mathematically sophisticated results remain largely confined to theoretical network analysis, while urban design discourse relies primarily on qualitative or empirical arguments regarding connectivity and permeability.

The present section seeks to bridge this divide. It interprets the equivalent cost families

discovered in Chapter 5 not merely as mathematical curiosities, but as indicators of a deeper structural property of urban mobility systems: equilibrium behavior constrains networks only up to functional equivalence, not spatial uniqueness. In other words, multiple distinct spatial configurations of a street network can generate identical equilibrium flows and identical aggregate travel-time outcomes. This insight has profound implications for how transportation modeling, planning evaluation, and urban design reasoning should be integrated.

In conventional transportation modeling practice, a given network design is typically treated as a fixed spatial object, for which equilibrium flows are computed. The planner then assesses whether the predicted flows and performance indicators are acceptable. If not, the design is modified and the process repeated. This iterative approach implicitly assumes that each equilibrium outcome corresponds to a unique spatial design. Equivalent cost families challenge this assumption. They reveal that equilibrium outcomes do not uniquely identify network design; instead, a single equilibrium regime may correspond to a multiplicity of spatial configurations. Consequently, equilibrium modeling does not yield a single design solution, but rather defines a design space within which multiple spatial realizations are functionally equivalent in equilibrium terms.

This realization shifts the epistemological role of equilibrium modeling in planning. Rather than serving solely as a predictive tool, equilibrium analysis becomes a generative and diagnostic instrument. It can be used not only to evaluate a given design, but also to identify families of alternative designs that share the same equilibrium behavior. This inversion of perspective transforms equilibrium theory into a framework for exploring design degrees of freedom rather than merely predicting traffic flows.

From the standpoint of urban design, this is a powerful insight. Urban street design is inherently concerned with balancing multiple objectives: mobility, safety, environmental quality, economic activity, and social life. Often, multiple spatial layouts can provide comparable mobility performance but differ substantially in spatial quality. Until now, transportation modeling has offered limited guidance on how to systematically navigate these trade-offs. Equivalent cost families provide precisely such guidance: they identify sets of spatial configurations that are equivalent in equilibrium mobility terms, thereby allowing designers to choose among them based on broader urban objectives.

This perspective reframes the Braess' paradox itself. Traditionally, the paradox is introduced as a cautionary tale: adding a new link can worsen traffic. The deeper lesson is not that connectivity is inherently harmful, but that connectivity must be structured rather than indiscriminately increased. Equivalent cost families show that multiple connectivity patterns can be equilibrium-equivalent; some may be spatially desirable, others spatially harmful. The paradox thus becomes a manifestation of design non-uniqueness under equilibrium constraints.

This interpretation aligns closely with contemporary urban design and mobility policy paradigms. Modern street design increasingly prioritizes human-scale environments, multimodal integration, controlled permeability, and place-making. Interventions such as pedestrianization, modal filtering, road diets, and shared-space schemes all modify effective link cost functions and network topology. Yet many of these interventions have

been implemented empirically, often relying on trial-and-error or qualitative reasoning. By contrast, equilibrium equivalence provides a theoretical foundation for understanding why such spatial interventions may preserve or even improve mobility performance despite reducing traditional vehicular capacity.

Moreover, empirical planning research has documented counterintuitive traffic responses: induced demand following capacity expansion, disappearing traffic after road closures, bypasses that fail to relieve congestion, and rat-running through residential streets triggered by minor connectivity changes. These observations suggest that equilibrium responses to spatial interventions are non-linear and system-wide. Equivalent cost families provide a mathematical representation of precisely such non-linear response structures. They offer a language for expressing how different spatial manipulations can produce identical or unexpectedly divergent equilibrium outcomes.

The objective of this section is therefore to articulate a systematic translation between three domains:

(i) abstract equilibrium theory and cost-function equivalence;

(ii) network-topological equivalence and non-unique functional realizations; and

(iii) spatial street design and urban-intervention interpretation.

By constructing this translation, the section situates the Braess' paradox not as an isolated equilibrium anomaly, but as an entry point into a broader theory of design under equilibrium constraints. The outcome is a framework in which transportation modeling, network science, and urban design reasoning become mutually intelligible rather than compartmentalized disciplines.

To accomplish this, the section proceeds in a sequence of conceptual expansions. Subsection 6.1 revisits the equivalent cost families established in Chapter 5, restating their derivation and emphasizing their theoretical implications for equilibrium identifiability and robustness. Subsequent subsections introduce the conceptual shift from parameter equivalence to design degrees of freedom, formulate the principle of spatial non-uniqueness, translate abstract cost parameters into street-design operations, and relate these mechanisms to empirical observations in contemporary urban design practice. Through this progression, the Braess' paradox is transformed from a counterintuitive equilibrium artifact into a systematic design diagnostic.

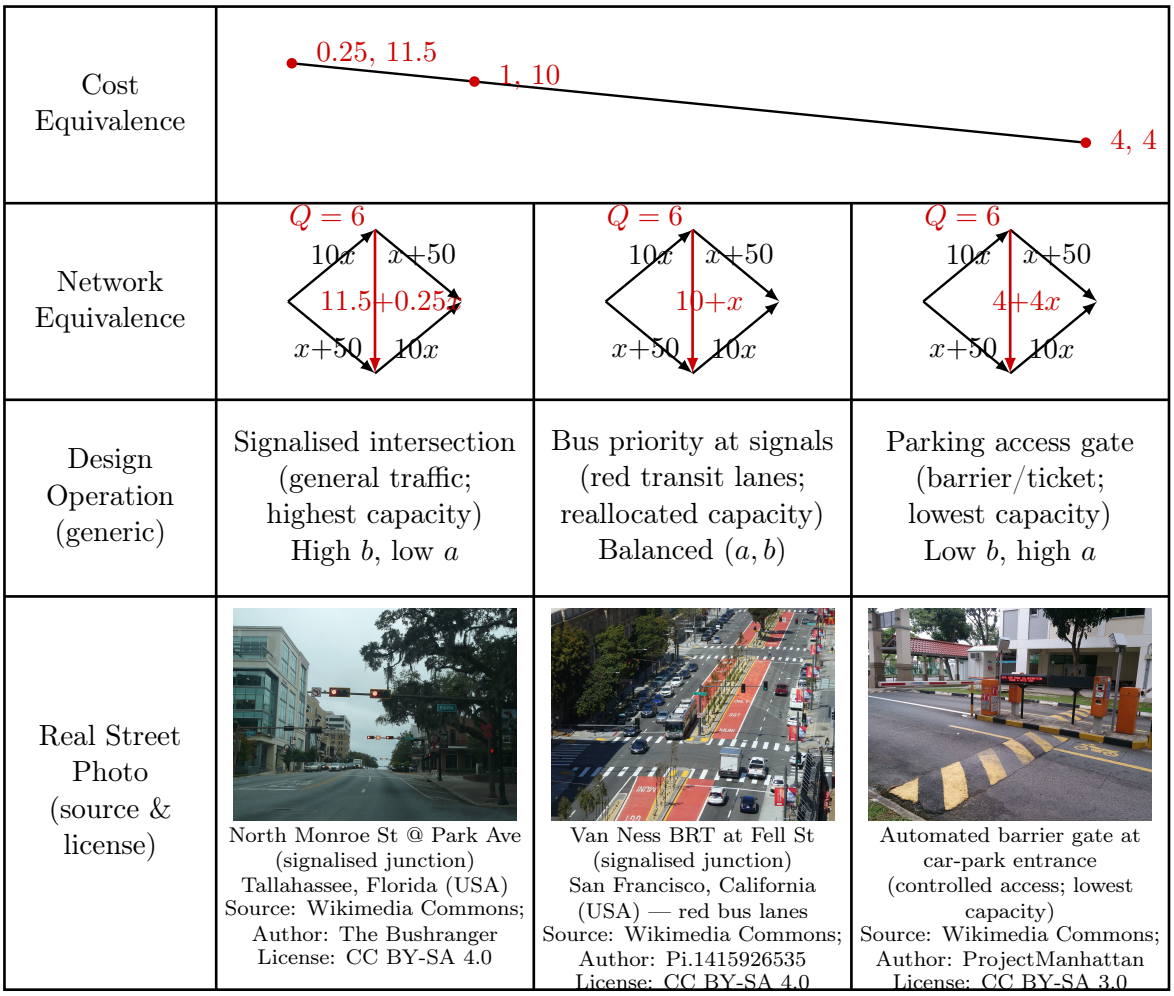


Figure 6.1: Four-level translation from equivalent cost families to spatial interpretation: (top) a continuous equivalence line in cost-parameter space; (second) distinct but equilibrium-equivalent network realizations with different added-link cost functions; (third) generic design-operation interpretations; (bottom) real-street analogues with sources and licenses, chosen to make node control explicit and to suggest an intuitive capacity gradient from a general signalised junction (left), to a bus-priority junction with red transit lanes (middle), to a barrier-controlled access gate (right).

Figure 6.1 summarises this translation as a four-level bridge from parametric cost equivalence to spatial interpretation. In particular, the bottom row grounds the abstract (a, b) trade-offs in three street-level control regimes with decreasing effective capacity for general traffic. First, the signalised junction at North Monroe St @ Park Ave (Tallahassee) represents conventional intersection control. In modelling terms, variations in cycle length, green split, and coordination primarily add an approximately flow-invariant component of delay—captured as a higher intercept effect (high b)—while the marginal congestion response remains comparatively modest (low a), consistent with the "signal control / delay" interpretation. Second, the Van Ness BRT treatment at Fell St (San Francisco)

illustrates bus priority implemented through red transit lanes and signal control that reallocate intersection capacity away from general traffic. Because part of the approach and/or green time is reserved for transit, the remaining general-traffic capacity is reduced; consequently, as general flow increases, delay can rise more rapidly, corresponding to a higher effective marginal sensitivity (higher a) together with a policy-induced baseline shift (a non-negligible b). This motivates the "balanced (a, b)" reading as a joint intercept–slope adjustment rather than a pure delay shift. Third, the automated car-park barrier gate exemplifies hard access control: the gate imposes a service-rate constraint that admits vehicles one-by-one, so queueing delay escalates sharply as arrivals approach the gate capacity. In stylised cost terms, this corresponds to a strong marginal deterrence (high a) while the free-flow component can remain comparatively small when demand is low (low b). These cases are not presented as empirical validations of Braess-equivalence; rather, they act as concrete operational analogues that render the theoretical design degrees of freedom implied by equivalent cost families legible in spatial and control terms.

6.1. Recap of Equivalent Cost Families

The notion of equivalent cost families developed in Chapter 5 constitutes the theoretical hinge upon which the spatial interpretation advanced in this section rests. It is therefore essential to restate this result in full conceptual depth, not merely as a repetition of prior derivations, but as a rearticulation of their broader implications for equilibrium theory, model identifiability, and network design logic.

Classical presentations of the Braess' paradox introduce a single numerical example in which a particular network, endowed with a specific set of link cost functions, exhibits an increase in total equilibrium travel time after the addition of a shortcut link. While this construction is pedagogically effective, it has often led to the misconception that paradoxical behavior is associated with a delicate coincidence of parameters. Under such an interpretation, the paradox could be dismissed as an exceptional edge case rather than a structural feature of equilibrium networks.

The analytical and numerical investigations carried out in Chapter 5 decisively refute this interpretation. By deriving closed-form equilibrium expressions for linear-cost networks and performing extensive parameter scanning for nonlinear BPR-type networks, the thesis demonstrates that paradoxical outcomes persist across entire continuous regions of parameter space. The classical Braess example is revealed to be merely a single point embedded within a broader manifold of equivalent cost configurations. This manifold constitutes what is termed an equivalent cost family.

Mathematically, an equivalent cost family is defined as a set of link cost parameterizations that generate identical equilibrium link flows and identical equilibrium performance measures under a fixed demand level. Equivalence is thus defined not by the shape of cost functions themselves, but by the equilibrium state they induce. Two networks may possess different free-flow travel times, congestion sensitivities, or capacity parameters, yet if they produce the same equilibrium outcome, they are functionally equivalent in

equilibrium terms.

This equivalence concept fundamentally alters how equilibrium models should be interpreted. Instead of mapping a given network specification to a unique equilibrium outcome, equilibrium relations define a many-to-one mapping: multiple network specifications correspond to the same equilibrium. The equilibrium solution therefore underdetermines the network specification. This property, although rarely discussed in transportation literature, is a general feature of systems defined by implicit equilibrium constraints.

The discovery of equivalence families also has an immediate implication for the interpretation of the Braess' paradox: paradoxical behavior is not a knife-edge phenomenon. It occupies finite-volume regions in parameter space. Consequently, small perturbations of cost parameters, such as those arising from empirical calibration uncertainty, do not destroy the paradox. The paradox is structurally stable, and must therefore be regarded as a realistic feature of equilibrium traffic assignment rather than a pathological curiosity.

To understand how equivalence arises, consider the general form of equilibrium conditions in a network with parametric link cost functions. Under deterministic user equilibrium, Wardrop's principle requires that all used routes between a given origin–destination pair have equal and minimal travel time. Expressed in link-flow variables, this yields a system of nonlinear equations in which equilibrium flows are functions of cost parameters.

Symbolic derivation in Chapter 5 shows that these equilibrium equations depend on cost parameters only through certain algebraic combinations. Individual parameters do not appear independently; instead, equilibrium conditions collapse parameter space into lower-dimensional structures. When two parameter vectors produce identical equilibrium equations, they necessarily produce identical equilibrium solutions. These vectors thus belong to the same equivalence family. In linear-cost networks, this mechanism can be derived explicitly: closed-form solutions reveal linear relationships among parameters that preserve equilibrium conditions, defining continuous affine subspaces in parameter space. In nonlinear networks, numerical continuation reveals analogous continuous families embedded in higher-dimensional nonlinear manifolds.

This phenomenon is closely related to parameter identifiability in inverse problems. If equilibrium relations do not uniquely determine parameter values, then multiple parameterizations are observationally indistinguishable from equilibrium data alone. The equivalent cost family is therefore the set of parameterizations that are observationally equivalent under equilibrium behavior. As a consequence, calibration based solely on equilibrium flow data cannot uniquely determine underlying cost parameters; additional independent information—such as speed measurements, queue lengths, or control delay—is required to resolve this non-identifiability.

It is essential, however, to emphasize that equivalence is conditional on demand level and equilibrium regime. Two networks may be equivalent under one demand scenario and diverge under another. Moreover, equivalence defined with respect to average travel time does not imply equivalence in reliability, emissions, safety, or equity. Equivalent cost families therefore define local equivalence within specified modeling scope, not universal

identity across all performance dimensions. Nevertheless, even such conditional equivalence is meaningful for planning because it identifies sets of alternative configurations that satisfy the same equilibrium mobility criterion under explicitly stated assumptions.

The deeper implication of equivalent cost families is that equilibrium modeling does not yield a unique functional representation of the network. If multiple cost parameterizations produce identical equilibrium outcomes, then equilibrium analysis alone cannot distinguish among them. This insight has rarely been acknowledged in transportation modeling practice, where calibrated parameters are often interpreted as uniquely determined physical quantities. Yet, when interpreted from a design perspective, non-uniqueness implies freedom: if multiple functional structures yield the same equilibrium behavior, then multiple physical network designs may achieve identical mobility outcomes.

To see why this matters for spatial interpretation, it is necessary to recall that link cost functions represent physical attributes of streets and traffic operations. Free-flow travel time reflects geometric distance and speed regulation; congestion sensitivity reflects lane capacity, intersection control, and conflict points; BPR parameters reflect saturation characteristics and control regimes. Altering cost-function parameters therefore corresponds to altering physical design and regulatory choices. If two parameterizations belong to the same equivalence family, then two physically different street designs can produce identical equilibrium outcomes. One realization may correspond to a longer street with higher speed; another to a shorter street with lower speed; another to a street with fewer lanes but optimized signal control. Equilibrium theory treats these as distinct networks, yet their equilibrium behavior is identical. This establishes the crucial conceptual bridge to urban design: equilibrium outcomes do not uniquely specify spatial form, but define equivalence classes of spatial configurations.

Moreover, the equivalence discovered in Chapter 5 is not restricted to cost parameters alone. In ladder and multi-bypass networks, different arrangements of shortcut placement and capacity distribution were shown to produce identical equilibrium patterns. Equivalence thus extends beyond parameterization into topology: different network geometries can be equilibrium-equivalent. For urban street networks composed of repeated motifs and hierarchical structures, this implies that connectivity changes may substantially alter local spatial conditions while preserving global equilibrium patterns under certain regimes, further reinforcing the conclusion that equilibrium analysis underdetermines spatial layout.

Recognising equivalence families therefore transforms equilibrium modelling into a design-space definition tool. Instead of computing a single equilibrium outcome for a given network, the analyst can derive sets of network configurations that share the same equilibrium outcome. These sets define feasible design spaces constrained by equilibrium relations. Within this space, additional criteria—safety, urban quality, environmental impact, and political feasibility—can guide selection among functionally equivalent options. In this sense, equilibrium modelling becomes one layer of constraint within a multi-criteria design process rather than a single deterministic predictor.

Finally, the existence of equivalent cost families reframes the meaning of the Braess'

paradox itself. The paradox is no longer simply a warning that adding a link may worsen performance. It is evidence that equilibrium networks possess hidden degrees of freedom: multiple cost and topology configurations are consistent with identical equilibrium states, and paradoxical outcomes arise in certain regions of this equivalence structure. Understanding the paradox in this way replaces the simplistic narrative of "adding links is bad" with a deeper structural insight: equilibrium behavior is governed by relations among parameters, not by individual parameters in isolation. Consequently, spatial interventions must be evaluated not locally but within the relational structure defined by equivalence.

6.2. From Mathematical Equivalence to Design Freedom

The existence of equivalent cost families established in the previous subsection leads to a fundamental conceptual reorientation. If multiple cost-function parameterizations generate identical equilibrium outcomes, then equilibrium theory does not prescribe a unique network realization. Instead, it defines a family of functionally equivalent networks. Mathematical equivalence in parameter space therefore translates into degrees of freedom in spatial and infrastructural design.

This shift is not a minor technical nuance but a profound change in how transportation modeling can support planning practice. Conventional traffic modeling workflows treat equilibrium assignment as an evaluative instrument: a spatial network is specified, equilibrium flows are computed, and performance indicators are reported. Design is then iteratively modified in response to these outputs. Implicit in this workflow is the assumption that each equilibrium outcome corresponds to a uniquely defined spatial design. Equivalent cost families reveal that this assumption is unfounded. Equilibrium relations constrain feasible designs, but they do not uniquely determine them. The outcome of equilibrium modeling is thus not a single optimal design but a structured design space.

Recognizing this distinction is essential for integrating transportation modeling and urban design. Urban design is not the search for a single optimal form, but the exploration of multiple admissible forms within performance constraints. Structural engineering does not prescribe a unique building shape; it defines feasible stress envelopes. Environmental simulation does not prescribe a unique façade; it defines thermal comfort ranges. Zoning codes do not prescribe a unique urban form; they define regulatory bounds. In precisely the same sense, equilibrium modeling defines admissible mobility-performance envelopes rather than unique spatial prescriptions. This analogy is crucial: it positions equilibrium analysis not as an external technical evaluation but as an internal generative constraint within the design process itself.

From a mathematical perspective, equilibrium conditions define a system of relations linking link flows, path costs, and cost-function parameters. Solutions of these relations describe equilibrium states. Equivalent cost families arise precisely because these relations do not uniquely determine all parameters. Some parameters can vary while leaving equilibrium flows unchanged. In formal terms, equilibrium equations underdetermine the

parameter space.

For planning practice, this underdetermination has decisive implications. A prescriptive model yields a unique recommended intervention. A constraint-based model defines permissible regions of design. Equivalent cost families demonstrate that equilibrium theory belongs to the latter category. It constrains but does not prescribe.

This distinction is central to resolving long-standing tensions between engineering and design cultures in planning. Engineers have traditionally viewed transport models as tools for determining "the correct" infrastructure intervention, while designers have emphasized contextual adaptation, spatial quality, and political feasibility. The existence of equivalence families shows that equilibrium models themselves do not yield a single "correct" answer. Instead, they define a feasible solution space within which design judgment is indispensable. The quantitative model does not replace design discretion; it frames it.

Moreover, interpreting equilibrium as constraint rather than prescription has methodological consequences for planning evaluation. It implies that comparing two designs solely by computing equilibrium flows may be insufficient if both designs fall within the same equivalence family. In such cases, equilibrium metrics alone cannot distinguish alternatives, and planners must explicitly consider additional criteria. This challenges the prevailing technocratic tendency to treat travel time savings as the dominant decision metric.

Equilibrium constraints thus become analogous to structural constraints in architecture: necessary conditions for feasibility, but not sufficient criteria for form selection. Recognizing this fundamentally repositions equilibrium modeling as a participatory design instrument rather than a black-box forecasting tool.

To operationalize design freedom, link cost parameters must be reinterpreted as spatial and operational design variables. In standard transport modeling, cost-function parameters are treated as descriptive coefficients representing existing conditions. However, in a design context, these parameters correspond to physical and regulatory attributes that can be modified.

Free-flow travel time reflects geometric distance and speed regulation. It can be altered by re-aligning streets, shortening routes, implementing traffic calming, adjusting speed limits, or introducing shared-space designs that modify effective speed. Congestion sensitivity reflects effective capacity and intersection performance. It can be modified by changing lane allocation, curb management, intersection geometry, signal timing, access control, or modal priority. In BPR-type formulations, parameters such as capacity and exponent values reflect saturation characteristics influenced by control technology, enforcement, and behavioral compliance.

Therefore, each point in cost-parameter space corresponds to a specific combination of geometric design, capacity provision, and operational regulation. Equivalent cost families then correspond to sets of spatial designs that differ physically yet produce identical equilibrium mobility outcomes. This translation gives cost-parameter equivalence direct spatial meaning.

This insight resolves a subtle but important ambiguity in traditional modeling practice. Calibration procedures typically infer cost-function parameters from observed flow-speed data. However, equivalent families imply that multiple parameter sets can reproduce the same equilibrium flows. Calibration therefore does not uniquely recover physical conditions; it recovers one member of an equivalence family. Recognizing this ambiguity clarifies why different modeling teams, using different calibration datasets or functional forms, often produce different parameter values while predicting similar equilibrium flows.

From a design perspective, this ambiguity becomes an opportunity. Instead of being constrained by a single inferred parameter set, planners can explore alternative parameter realizations consistent with equilibrium behavior, each corresponding to different spatial design choices. Cost parameters thus become levers for design exploration rather than fixed descriptors of reality.

Once cost parameters are understood as design variables, equivalent cost families become explicit representations of design degrees of freedom. Within a given equivalence family, designers can vary street geometry, lane allocation, speed regulation, and control strategies while preserving equilibrium travel times and flow distributions.

In practical terms, this means that a target equilibrium regime can be realized through multiple design strategies. A corridor may achieve a given equilibrium performance by geometric widening, by intelligent signal coordination, by implementing adaptive traffic control, by dedicating transit lanes that alter capacity distribution, or by introducing congestion pricing that modifies effective cost functions. All these interventions correspond to different points in cost-parameter space that may lie on the same equivalence family.

This design freedom has significant economic and political implications. Large-scale infrastructure expansion is capital-intensive and disruptive, while operational or regulatory interventions may achieve equivalent equilibrium effects at lower cost and with less spatial disruption. Equivalent cost families provide a rigorous analytical basis for comparing such alternatives on equal functional footing. The decision between building a new road, reallocating existing road space, or implementing pricing and control measures can thus be framed as a choice among equilibrium-equivalent design realizations.

Furthermore, design degrees of freedom allow local contextual adaptation. In historic urban fabrics, geometric widening may be impossible; operational optimization may be preferable. In newly developing districts, geometric restructuring may be feasible; regulatory interventions may be secondary. Equivalent families enable both contexts to achieve similar equilibrium mobility outcomes through different spatial strategies.

Design freedom under equilibrium equivalence extends beyond continuous parameter variation to include topological and operational modifications. Chapter 5 demonstrated that different placements of bypass links or connectors in ladder networks can produce identical equilibrium patterns. This reveals that equivalence is not confined to cost parameters but also encompasses network topology.

For urban networks, this is particularly significant. Connectivity interventions may be

realized through physical streets, transit-only links, pedestrian passages, signal-priority corridors, or filtered permeability schemes. Each represents a distinct topological or operational configuration. Yet under certain conditions, they may belong to the same equilibrium equivalence class.

This introduces topological degrees of freedom in spatial design. In dense urban environments where constructing new streets may be infeasible, operational or modal reconfigurations may realize the same equilibrium effects as physical expansion. For example, a new bus-only connector with signal priority may functionally substitute for a general-traffic shortcut. Similarly, removing a general-traffic connection while preserving pedestrian and cycling permeability may achieve equilibrium-equivalent redistribution of vehicular flows.

Recognizing topological equivalence also clarifies the mechanism behind rat-running phenomena. Small connectivity changes in local street networks can shift equilibrium route sets, producing disproportionate flow redistribution even when physical modifications are minor. From an equivalence perspective, such changes move the network from one equivalence region to another, explaining why small spatial interventions can trigger large traffic responses.

Urban design decisions rarely prioritize mobility alone. Streets are simultaneously movement corridors, public spaces, ecological assets, and economic environments. Consequently, design evaluation must integrate mobility performance with safety, comfort, environmental sustainability, and place quality.

Equivalent cost families provide a rigorous method for decoupling mobility performance from other objectives. If multiple spatial configurations are equilibrium-equivalent in mobility terms, designers are free to choose among them based on non-mobility criteria. This transforms mobility from a dominating objective into a constraint within a broader multi-objective design problem.

This is particularly relevant for contemporary sustainable mobility strategies. Policies promoting walking, cycling, and public transport often involve reallocating road space away from private vehicles. Conventional traffic models may predict congestion impacts from such reallocation. However, equivalence analysis can reveal design configurations in which spatial reallocation and operational adjustment preserve equilibrium performance while enhancing urban quality.

Thus, equilibrium equivalence does not merely describe mobility phenomena; it enables principled negotiation between mobility and place. It provides the analytical foundation for a design approach in which urban quality improvements are pursued within equilibrium-feasible mobility envelopes, rather than being treated as trade-offs against mobility performance.

The analysis above leads to a fundamental conclusion: equilibrium modeling does not yield a unique spatial network design. Equivalent cost families define sets of spatially distinct but functionally identical networks. This non-uniqueness is not a modeling flaw but an intrinsic property of equilibrium systems.

Recognizing spatial non-uniqueness completes the conceptual bridge from mathematical equivalence to spatial interpretation. It implies that transportation networks are not uniquely determined by equilibrium performance criteria. Instead, they exist within equivalence classes of possible spatial realizations. The role of planning and design is to navigate these equivalence classes, selecting configurations that satisfy mobility constraints while fulfilling broader urban objectives.

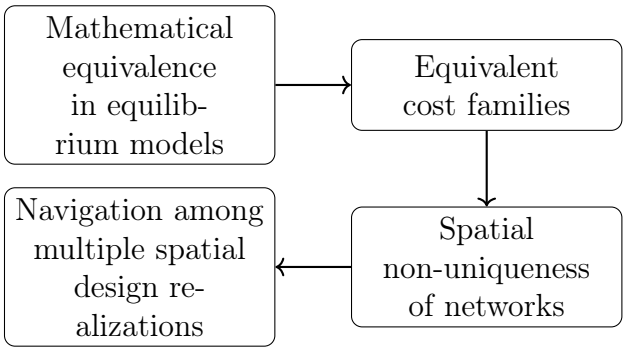


Figure 6.2: Conceptual transition from mathematical equivalence to spatial non-uniqueness in equilibrium network design.

6.3. Network Equivalence and Spatial Non-Uniqueness

The recognition of equivalent cost families and design degrees of freedom leads to a deeper theoretical implication: equilibrium networks do not possess a unique spatial realization. If multiple cost-parameter configurations and even multiple topological arrangements generate identical equilibrium outcomes, then equilibrium theory constrains networks only up to functional equivalence, not spatial determinacy. This subsection formalizes this principle as spatial non-uniqueness of equilibrium networks and establishes its conceptual consequences for urban network interpretation.

The argument developed here is intentionally cumulative. It proceeds from the abstract level of functional equivalence, to the network-structural level of topology and geometry, and finally to the concrete level of spatial street design and planning methodology. Through this progression, spatial non-uniqueness is not presented as an intuitive claim but as a necessary consequence of equilibrium theory once equivalence families are acknowledged.

In equilibrium modeling, networks are represented as directed graphs endowed with link cost functions. The equilibrium solution determines link flows and travel times on this abstract representation. Traditionally, it is implicitly assumed that each abstract network corresponds to a specific spatial street layout. Equivalent cost families challenge this assumption by demonstrating that multiple abstract networks—differing in parameterization or even topology—can produce identical equilibrium behavior.

If equilibrium behavior is identical, these abstract networks are functionally equivalent.

When cost parameters are interpreted as physical design attributes, functional equivalence implies that multiple spatial networks can realize the same equilibrium regime. In other words, there exists a many-to-one mapping from spatial network realizations to equilibrium states. This mapping is surjective but not injective: many spatial forms correspond to one equilibrium outcome. Equilibrium theory therefore does not select a single spatial configuration; it defines equivalence classes of spatial networks that share identical equilibrium behavior, any member of which is feasible from a mobility-performance standpoint.

It is useful to emphasize that this non-uniqueness is not a trivial modeling artifact but a direct consequence of equilibrium conditions being expressed as relational equalities among travel times rather than absolute prescriptions of infrastructure form. Wardrop's principle requires equality of route costs at equilibrium, but it does not specify how these costs must be generated physically. As long as relative cost relations are preserved, equilibrium flows remain unchanged. Spatial non-uniqueness is thus embedded in the very formulation of equilibrium.

This insight parallels phenomena in other fields governed by equilibrium constraints. In statics, multiple internal force distributions can satisfy equilibrium of a structure; additional constitutive relations are needed to determine a unique stress field. In thermodynamics, multiple microstates correspond to the same macroscopic equilibrium state. In transportation networks, multiple spatial microstructures correspond to the same equilibrium flow macrostate. The analogy clarifies that non-uniqueness is not a defect but an inherent feature of equilibrium-based reasoning.

Spatial non-uniqueness manifests across three interrelated dimensions: topology, geometry, and operation. Each dimension corresponds to a different way in which physical networks can vary while preserving equilibrium behavior.

At the topological level, different connectivity arrangements can generate identical equilibrium route sets. Chapter 5 demonstrated that in ladder networks, distinct bypass placements may produce identical equilibrium flows. For urban networks, this implies that multiple street-connection patterns may be functionally interchangeable from an equilibrium perspective: a connector inserted at one location may have the same equilibrium effect as a connector inserted elsewhere, provided it induces the same structure of route alternatives and preserves the relevant cost relations among competing paths. This observation has direct relevance for debates on permeability: gridded connectivity, selective filtering, cul-de-sac hierarchies, or superblock structures may, under certain parameter conditions, belong to the same equilibrium equivalence class, thereby achieving similar mobility outcomes despite radically different morphologies.

At the geometric level, streets of different length, width, and alignment can generate identical effective cost-function parameters. A longer street with higher permitted speed may produce the same free-flow travel time as a shorter street with lower speed limit. A narrower street with optimized signal control may achieve the same effective capacity as a wider street with conventional control. Intersection geometry, turning radii, curbside management, and lane markings all influence effective congestion parameters without

necessarily altering network topology. Consequently, geometric form is not uniquely determined by equilibrium performance; two cross-sections with very different spatial character may nonetheless produce the same equilibrium link cost under different operational regimes.

At the operational level, regulatory or technological measures can replicate the equilibrium effects of physical construction. Signal coordination, adaptive traffic control, dynamic lane assignment, congestion pricing, or modal priority can all modify effective link costs without altering geometry. These operational measures therefore lie in the same equivalence classes as certain geometric interventions. Together, topology, geometry, and operation form a triad of realization dimensions: transformations along any of these dimensions may preserve equilibrium outcomes, allowing heterogeneous interventions to be compared within a single analytical framework.

The principle of spatial non-uniqueness can be stated precisely as follows:

Given a fixed demand and equilibrium assignment model, there exists a mapping from spatial network configurations to equilibrium flow states that is many-to-one. Therefore, multiple spatial network realizations may correspond to the same equilibrium outcome. Equilibrium analysis alone cannot uniquely determine spatial network form.

This statement has significant methodological implications. It implies that equilibrium models cannot, by themselves, prescribe unique spatial designs. Additional criteria beyond equilibrium mobility performance are required to select among equivalent spatial realizations. It also clarifies a recurring empirical observation in planning practice: transport modeling exercises often identify multiple design alternatives yielding similar predicted performance. Rather than being dismissed as modeling ambiguity, this multiplicity should be recognized as a structural property of equilibrium-constrained design. The model is not failing to discriminate; the underlying equilibrium relations genuinely permit multiple realizations.

Spatial non-uniqueness also reframes the Braess' paradox. Traditionally, the paradox is interpreted as showing that adding a new link may worsen equilibrium performance. Under the lens of non-uniqueness, the paradox is instead evidence that interventions move the network from one equivalence class to another, potentially into a less desirable equilibrium regime. Adding a street does not deterministically improve or worsen performance; it relocates the network within the space of equilibrium equivalence classes. Whether this relocation is beneficial depends on the relational structure of the entire network, not on the local intervention alone. This interpretation dissolves the simplistic narrative of "connectivity is harmful" and replaces it with a relational account: the impact of an intervention depends on how it reshapes the equilibrium relations among competing routes.

Spatial non-uniqueness has direct consequences for planning methodology. If multiple spatial configurations are equilibrium-equivalent, then travel-time metrics alone cannot discriminate among them. Planning decisions must therefore explicitly incorporate additional criteria such as urban quality, safety, environmental impact, economic development,

and social equity. In this sense, equilibrium analysis defines admissible solution families, while design and governance select preferred spatial realizations within those families.

This provides a theoretical justification for design-led and participatory approaches: rather than delegating selection to model outputs, planners can use equilibrium analysis to map the feasible equivalence class and then evaluate alternatives through multi-criteria reasoning. Spatial non-uniqueness also supports incremental and reversible interventions. Tactical urbanism, pilot street closures, pop-up bike lanes, and adaptive signal trials can be interpreted as exploratory moves within (or between) equivalence classes, enabling learning without irreversible commitments. Furthermore, the framework offers a rigorous explanation for empirical observations of "disappearing traffic": when streets are closed or capacities reduced, the network may shift to a different realization within an equivalence family, redistributing flows without global performance degradation under certain regimes.

Finally, while equivalence is defined for fixed demand and behavioral assumptions, non-uniqueness remains meaningful under uncertainty. Spatial realizations that are equivalent at one demand level may diverge at another. This does not negate non-uniqueness; rather, it motivates robustness-oriented selection: planners can seek spatial configurations that remain within desirable equilibrium classes across multiple plausible demand and behavioral scenarios. Spatial non-uniqueness thus underpins resilient planning strategies that avoid brittle dependence on a single forecast.

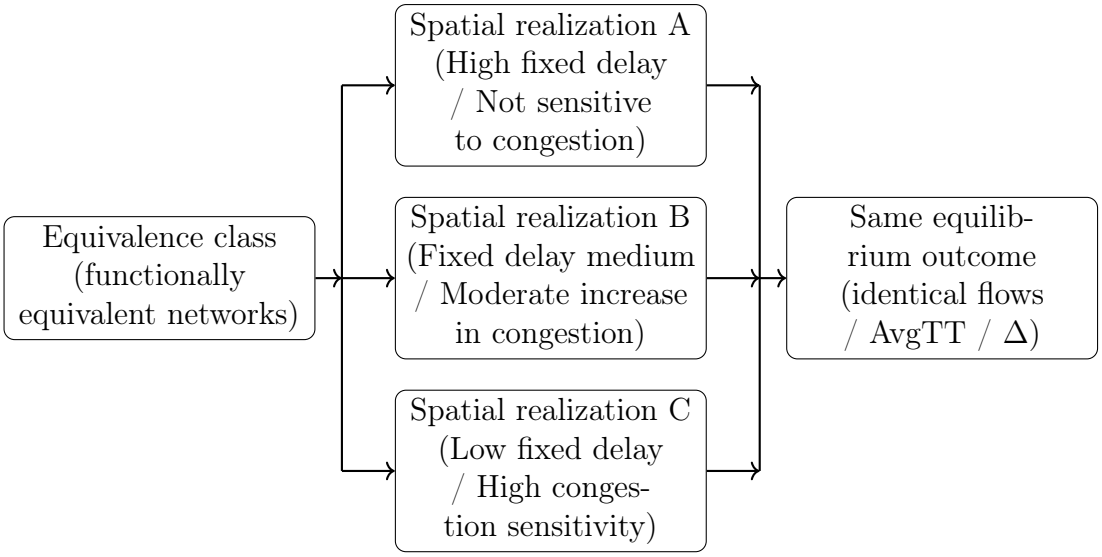


Figure 6.3: Structural representation of spatial non-uniqueness: an equivalence class can generate multiple spatial realizations that nonetheless yield the same equilibrium outcome.

6.4. Translating Cost Parameters into Street Design Elements

The previous subsections have established that equivalent cost families define functional equivalence classes of equilibrium networks and that such equivalence implies spatial

non-uniqueness. The next necessary step is to operationalize this insight by translating abstract cost-function parameters into concrete street-design elements. Without such translation, equivalence remains a mathematical construct; with it, equivalence becomes a practical design instrument.

In transportation modeling, link cost functions encapsulate the relationship between flow and travel time. In urban design, street elements such as geometry, cross-section, control regime, and modal allocation shape that relationship. Therefore, every parameter in a cost function corresponds, directly or indirectly, to a modifiable design attribute. This subsection constructs a systematic mapping between these two domains and demonstrates how manipulating design elements moves a network along an equivalence family in parameter space.

The purpose of this translation is not to claim a one-to-one mapping between model parameters and physical dimensions. Rather, it establishes an interpretive correspondence: cost parameters describe effective behavioral consequences of spatial form, while design elements define the spatial means through which those behavioral consequences are produced. This correspondence is sufficient to support equilibrium-informed design reasoning.

Consider a general link cost function expressed in affine or BPR-type form:

$$t_a(x_a) = b_a + m_a x_a,$$

or

$$t_a(x_a) = t_a^0 \left(1 + \alpha_a \left(\frac{x_a}{c_a}\right)^{\beta_a}\right).$$

In both formulations, parameters control three fundamental aspects of link performance:

(i) **Baseline travel time**: b_a or t_a^0

(ii) **Congestion sensitivity**: m_a or (α_a, β_a)

(iii) **Capacity scale**: c_a

Each of these aspects corresponds to specific categories of street-design interventions.

Baseline travel time represents free-flow conditions. It depends primarily on link length, permitted speed, curvature, and intersection delay. Congestion sensitivity represents how rapidly travel time increases as flow rises; it depends on effective lane capacity, signal control, access frequency, and friction from non-motorized activity. Capacity scale reflects the maximum sustainable throughput; it depends on lane number, lane width, turning provision, and priority assignment.

This decomposition reveals that modifying a cost parameter corresponds to modifying one or more street-design variables. Therefore, moving along an equivalence family in cost-parameter space corresponds to coordinated modifications of geometric and operational design elements in physical space.

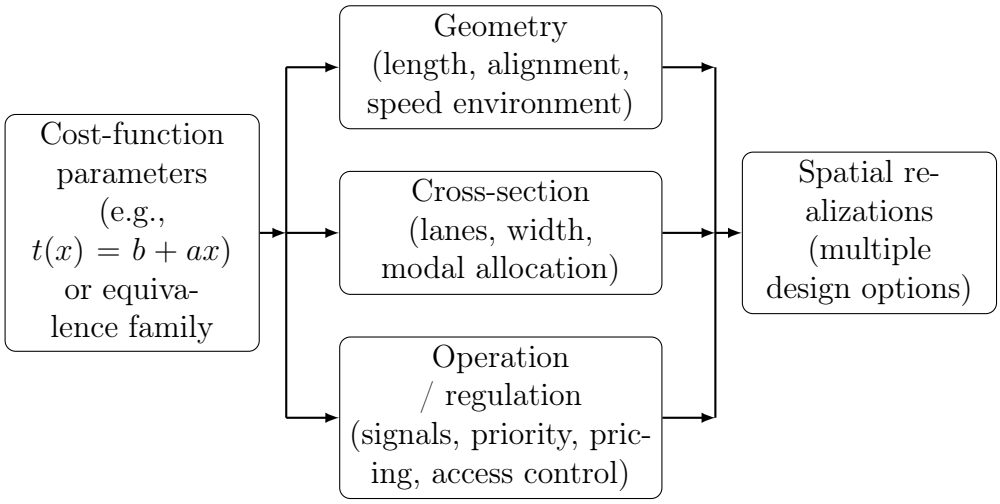


Figure 6.4: Structural translation from cost-function parameters to street-design elements: geometry, cross-section, and operational regulation jointly define multiple spatial realizations under the same equilibrium constraints.

The baseline component of link cost represents free-flow travel conditions. In physical terms, this is controlled by street geometry and regulatory speed. The following spatial design elements directly influence baseline travel time:

(i) Link length and alignment

(ii) Street curvature and gradient

(iii) Posted or enforced speed limits

(iv) Intersection density and spacing

A longer street with higher permitted speed may produce the same baseline travel time as a shorter street with lower speed. Similarly, a straight alignment with traffic-calming measures may yield identical free-flow travel time as a curved alignment with higher operating speed. These different geometric realizations correspond to different spatial experiences but identical baseline cost parameters.

From an equivalence-family perspective, varying baseline parameters while compensating through other parameters (e.g., congestion sensitivity) yields alternative street geometries that preserve equilibrium outcomes. This provides a formal explanation for why multiple street-layout strategies can produce similar network performance despite differing spatial morphology.

(A) Longer alignment + higher speed	(B) Shorter alignment + lower speed
Location: Viale Enrico Forlanini, Milan (IT). Source: Wikimedia Commons (Author: Van Loon) License: CC BY-SA 3.0	Location: Via Cavriana, Milan (IT). Source: Wikimedia Commons (Author: Van Loon) License: CC BY-SA 3.0
v_A (higher speed), e.g., 50 km/h L_A fewer intersections $\Rightarrow d_{\text{int},A}$ small $t_A^0 \approx \frac{L_A}{v_A} + d_{\text{int},A}$	v_B (lower speed), e.g., 30 km/h L_B ($< L_A$) more intersections $\Rightarrow d_{\text{int},B}$ larger $t_B^0 \approx \frac{L_B}{v_B} + d_{\text{int},B}$
Same baseline travel time (free-flow): $t_A^0 = t_B^0$	

Figure 6.5: Two real urban-street examples (Milan) illustrating baseline (free-flow) travel-time equivalence through compensating changes in alignment length, speed environment, and intersection delay.

Congestion sensitivity describes how rapidly travel time increases with flow. This aspect of performance is governed primarily by street cross-section design and intersection control. Key design variables include:

(i) Number and width of lanes

(ii) Presence and design of turning pockets

(iii) Signal phasing and cycle length

(iv) Access management and curbside activity

(v) Degree of modal mixing or segregation

A wide multi-lane arterial with frequent access points may exhibit similar congestion sensitivity as a narrower street with optimized signal coordination and restricted access. Likewise, a street with mixed traffic but generous width may show equivalent congestion response to a narrower but transit-prioritized corridor. These different cross-sectional designs map onto different congestion parameters in cost functions.

Equivalence families demonstrate that altering cross-section and control design can compensate for geometric differences while preserving equilibrium travel time. This insight

supports flexible street-design strategies in which geometric expansion is not the only means of improving mobility performance.

(A) Wide multi-lane arterial with frequent access	(B) Narrower street with optimized control and restricted access
Japan National Route 330, Shimashi, Okinawa (JP) Source: Wikimedia Commons; Author: Benlisquare; License: CC BY-SA 4.0.	**Presidential Boulevard, Lower Merion, Pennsylvania (US)** Source: Wikimedia Commons; Author: Montgomery County Planning Commission; License: CC BY-SA 2.0.
Many lanes + frequent access ⇒ higher intersection / access friction Effective congestion sensitivity: moderate	Fewer lanes + optimized signals + restricted access Effective congestion sensitivity: moderate
Equivalent congestion sensitivity (same slope of cost function)	

Figure 6.6: Real-street analogues for cross-section equivalence: a wide multi-lane arterial with frequent access versus a narrower, more tightly controlled street environment. Despite different spatial configurations and control regimes, both can be interpreted as producing comparable congestion sensitivity (i.e., similar slope of the effective cost function).

Capacity parameters represent the scale at which congestion effects become significant. Physically, capacity is influenced not only by geometry but also by operational regulation. Design variables include:

(i) Lane assignment and dynamic lane control

(ii) Signal priority and adaptive control

(iii) Bus or bicycle priority schemes

(iv) Turn restrictions and access limitation

(v) Pricing and demand-management measures

For example, dynamic lane control can increase effective capacity without changing physical width. Signal priority for transit can redistribute capacity among modes. Congestion pricing can reduce effective demand and thereby modify the capacity-flow relationship experienced by users. All these measures alter capacity parameters in cost functions.

This demonstrates that equivalence families include not only geometric but also regulatory realizations. Operational strategies can thus serve as spatial design instruments that achieve equilibrium effects comparable to physical reconstruction.

Because cost-function parameters jointly determine equilibrium behavior, design translation must consider coordinated modifications across geometry, cross-section, and operation. Equivalence families represent parameter combinations that preserve equilibrium. Therefore, spatial realizations must involve coordinated design adjustments that preserve functional relations among street elements.

For instance, reducing lane width (reducing capacity) may be compensated by improved signal coordination (reducing congestion sensitivity), keeping equilibrium performance constant. Alternatively, lowering speed limits (increasing baseline travel time) may be compensated by reduced intersection delay (lowering congestion sensitivity). These composite adjustments yield new spatial forms that remain within the same equilibrium equivalence class.

This coordinated perspective explains why successful urban street redesigns often involve integrated packages of interventions rather than isolated measures. Equilibrium equivalence provides the analytical justification for such integrated design logic.

To operationalize this translation in design practice, cost-to-design correspondence must be represented visually. The following representational tools are recommended:

(i) Parameter-to-design mapping diagrams

(ii) Comparative street-section drawings

(iii) Network topology variation diagrams

(iv) Flow-distribution overlays on spatial layouts

These representations make equivalence tangible for designers and stakeholders, allowing mathematical relations to be understood as spatial possibilities.

The translation developed above allows equilibrium modeling to be integrated directly into design workflows. Rather than producing model outputs after a design is proposed, equilibrium relations can inform the generation of design alternatives from the outset.

A possible workflow is as follows:

(i) Identify desired equilibrium mobility performance.

(ii) Derive corresponding cost-parameter equivalence family.

(iii) Translate parameter variations into design-element variations.

Table 6.1: Operational regulation measures and their correspondence to capacity (c)-related parameters in link cost functions.

Operational measure	**What changes in practice**	**Parameter implication**	**Typical network-level effect (qualitative)**
Dynamic lane assignment / lane control	Reallocates lanes by time-of-day or real-time conditions (e.g., reversible lanes, hard shoulder running).	$c \uparrow$ (effective capacity) during peak; saturation effects reduced.	Delays onset of congestion; stabilizes throughput without widening.
Adaptive signal control / coordination	Adjusts splits, offsets, cycle length; reduces lost time and spillback.	$c \uparrow$ at nodes; control-delay component decreases.	Improves progression; reduces stop-and-go and variability.
Transit signal priority (TSP)	Allocates green time to transit (early/extended green).	Capacity redistributed by mode: $c_{\text{transit}} \uparrow$, possibly $c_{\text{car}} \downarrow$ locally.	Encourages mode shift; improves transit reliability; may reduce car volumes.
Bus/Bike priority (dedicated lanes)	Reassigns road space; reduces conflicts for prioritized modes.	$c_{\text{bus}} \uparrow$, $c_{\text{bike}} \uparrow$; $c_{\text{car}} \downarrow$ locally.	Safety/reliability improve; local car delay may be offset by reduced demand.
Turn restrictions / access limitation	Reduces conflict points; consolidates access and turning movements.	Effective $c \uparrow$; congestion sensitivity reduced via fewer conflicts.	Less intersection blocking; smoother arterial flow; traffic relocated to access points.
Curbside management	Controls stopping/parking/loading; reduces lane blockages.	Effective $c \uparrow$; fewer random capacity drops.	Improves reliability; mitigates shockwaves on constrained corridors.
Congestion pricing / demand management	Raises generalized cost; shifts/suppresses trips by time/route/mode.	Effective demand x (or Q) $\downarrow$; performance improves without c increase.	Moves system out of high-congestion regime; redistributes flows network-wide.

Table 6.2: Visual translation matrix from cost-function parameters to street-design variables.

Cost parameter	**Geometry**	**Cross-section**	**Operation / regulation**
Baseline travel time (b or t^0)	Length, alignment, speed environment	–	Speed limits, traffic calming
Congestion sensitivity (a or α, β)	–	Lane number, width, modal allocation	Signal phasing, coordination
Capacity scale (c)	–	Lane assignment, turning pockets	Pricing, access control, dynamic lanes

(iv) Generate multiple spatial design alternatives.

(v) Select among alternatives based on broader urban objectives.

This workflow transforms equilibrium analysis from a reactive evaluation tool into a proactive design generator. It also provides a rational basis for comparing physically distinct interventions—new streets, signal upgrades, lane reallocation, pricing schemes—on equal equilibrium-performance terms.

(A) New connector link	(B) Cross-section optimization	(C) Pricing / access control
Ramp-to-ramp connection (new auxiliary lanes), I–205 (US) — CC BY 2.0 Source: Wikimedia Commons (Author: Oregon Department of Transportation).	**Road diet (after), Davis, California (US)** — CC BY-SA 3.0 Source: Wikimedia Commons (Author: TransportObserver).	**Electronic Road Pricing (ERP), Singapore** — CC0 1.0 Source: Wikimedia Commons (Author: ProjectManhattan).
Geometry-led intervention (adds a new link) Main lever: baseline t^0 / intercept	Section + control package (no new links) Main lever: congestion sensitivity (slope / α, β)	Regulation-led package (pricing / restrictions) Main lever: effective Q or c
Same equilibrium regime (identical AvgTT / flows / Δ)		

Figure 6.7: Three alternative street-design realizations can be tuned to satisfy the same equilibrium constraints: (A) adding a new connector link, (B) cross-section and intersection-control optimization, and (C) pricing and access regulation.

6.5. Alternative Spatial Realizations of Equivalent Cost Families

The previous subsections established that equivalent cost families generate functional network equivalence and imply spatial non-uniqueness of equilibrium networks. This subsection reframes that claim explicitly as a design statement: once equilibrium performance is treated as a constraint, multiple spatial interventions can be engineered to satisfy the same equilibrium regime, even though their physical form, operating rules, and street character differ.

The argument is grounded in three illustrative case studies on the classical Braess diamond network, computed through the symbolic-equivalence workflow introduced in Section 4.5 (see the symbolic derivation protocol and numerical cross-check rules). In each case, the added connector $2 \to 3$ (link 5) admits a continuous family of admissible linear cost specifications. Rather than reproducing the numerical coefficients and sampled parameter tables here (reported in Chapter 5 and the analytical appendix), the focus of this section is

the spatial meaning of such families: how moving along an equivalence family corresponds to switching between alternative geometric, cross-sectional, and operational realizations while preserving the same equilibrium outcomes.

In the first configuration, the admissible set of link 5 specifications forms a one-dimensional locus in the two-dimensional parameter space (a_5, b_5). Spatially, moving along this locus can be read as a controlled trade-off between:

(i) baseline delay components (intersection control, access friction, pricing, or fixed operational penalties), and

(ii) congestion sensitivity components (cross-section, effective capacity, and how rapidly delay grows with flow).

A higher baseline-delay / lower congestion-sensitivity realization corresponds to a connector whose performance is dominated by fixed controls (e.g., signal delay, filtering, or pricing), but whose geometric supply is generous enough that marginal congestion grows slowly. Conversely, a lower baseline-delay / higher congestion-sensitivity realization corresponds to a connector that is physically direct and operationally permissive, but geometrically constrained so that marginal delay increases rapidly with usage. Both realizations can be tuned to satisfy the same equilibrium constraint, hence producing the same network-wide equilibrium regime even though their street character differs.

The second configuration demonstrates that equivalence families are relational: the admissible substitutions for the connector depend on the surrounding street environment. When the two main corridors differ in their baseline delay and congestion sensitivity, the connector must compensate accordingly in order to preserve the same equilibrium regime. Design freedom therefore persists, but its direction in the space of feasible substitutions changes with context.

In practical terms, this implies that an "equivalent" connector design cannot be prescribed in isolation (e.g., by specifying a single geometric typology). Instead, it must be calibrated relative to the background network: the same connector geometry may behave as a low-delay shortcut in one context but as a high-friction local link in another.

The third configuration reinforces robustness: even under strongly unbalanced background conditions, the connector still admits multiple admissible realizations that preserve the target equilibrium regime. From a design standpoint, this is the key message: spatial non-uniqueness is not a fragile artifact of a symmetric toy network, but a structural feature of equilibrium constraints.

Across configurations, the case studies support three design-relevant conclusions:

(i) **Equilibrium performance is an underdetermined constraint.** Multiple design degrees of freedom (geometry, cross-section, operations/regulation) can satisfy

the same equilibrium regime.

(ii) **Equivalence families define a menu, not a single solution.** A planner can choose among alternatives that share mobility outcomes but differ in livability, safety, heritage compatibility, political feasibility, and environmental performance.

(iii) **Admissible substitutions are context-dependent.** The background network determines how baseline delay and congestion sensitivity must be traded off to preserve equivalence.

These conclusions justify an equilibrium-informed design workflow in which numerical/symbolic analysis is used to delimit an admissible design set, while urban design criteria select among the admissible realizations.

6.6. Empirical Correspondence with Urban Design Practice

The numerical and analytical results presented in Sections 6.1–6.5 establish that equivalent cost families and spatial non-uniqueness are intrinsic properties of equilibrium-constrained networks. The relevance of this framework for urban planning, however, depends on whether analogous regularities can be recognised in real-world spatial interventions. This section therefore connects the theoretical findings of this thesis to empirical practices in contemporary urban design and transportation planning.

The spatial transfer framework developed in this thesis formalises a bridge from

(i) continuous families of admissible cost functions that yield invariant equilibrium outcomes,

(ii) network equivalence, and

(iii) spatial non-uniqueness and transferable design mechanisms.

Building on this structure, the present section develops a systematic correspondence between the three numerical configurations demonstrated in Section 6.5 and representative interventions commonly employed in planning practice.

Crucially, the three numerical configurations are not treated here as purely abstract constructions. Each configuration specifies how distinct slope–intercept pairs (a_5, b_5) for the connecting link can produce the same equilibrium performance shift under fixed background parameters. In spatial terms, these combinations correspond to alternative realisations of connector design—including geometric allocation, operational control, access regulation, and pricing—that differ in their local attributes while remaining equivalent at the equilibrium, network-wide level. The following subsections illustrate how analogous equivalence relations can be interpreted within empirical urban transformations.

The first numerical configuration demonstrated that a continuous family of cost functions,

$$b_5 = 12 - 2a_5, \tag{6.1}$$

produces an invariant equilibrium performance shift $\Delta = 9$ under fixed background parameters (Section 6.5). Mathematically, (6.1) indicates that the slope and intercept of the connecting link can trade off continuously while preserving the equilibrium outcome. In spatial terms, this implies that different combinations of fixed delay and congestion sensitivity for a new connection can yield identical system-level consequences.

A practical analogue is congestion pricing. Cordon charging and access pricing introduce an approximately flow-independent monetary component into generalised travel cost, effectively increasing the intercept term associated with selected movements or zones. From the standpoint of equilibrium response, such an intercept shift does not act in isolation: it changes route choice incentives, suppresses or re-times marginal trips, and redistributes demand across the remaining alternatives until a new equilibrium is reached.

This mechanism is consistent with Configuration 1 in a direct parametric sense. In the numerical setting, moving along (6.1) increases the fixed component b_5 while reducing the marginal congestion sensitivity a_5, with the aggregate performance indicator remaining invariant by construction. In the spatial setting, pricing raises the fixed component of perceived cost, while the induced traffic reduction and operational adaptation effectively reduce exposure to marginal congestion on the priced movements (an "effective" reduction in slope at the level of experienced conditions). Empirically, this is precisely the outcome that makes pricing politically and operationally legible: despite a substantial modification of the cost decomposition, the post-adaptation regime can exhibit stable (and in many monitored instances improved) network speeds within the charged area, even though the spatial distribution of flows and modal shares is substantially reconfigured. In this sense, pricing can be interpreted as movement along an equivalent cost family: a parametric shift in intercept produces a new spatial–behavioural configuration that remains equilibrium-equivalent at the level of the aggregate performance measure.

Spatially, this interpretation highlights an important design implication. Rather than realising a new connector solely through geometric construction, a city may realise a functionally analogous "connector" through policy instruments that alter effective link costs. Configuration 1 therefore corresponds to a design family in which one realisation is a physical shortcut and another is a priced corridor; both can belong to the same equilibrium equivalence class.

London Congestion Charge zone signage in street context (Central Zone).
Source: Wikimedia Commons (Author: Mariordo).
License: CC BY-SA 3.0.

Congestion tax information sign (Stockholm).
Source: Wikimedia Commons.
License: Public domain (Transportstyrelsen / Swedish Transport Agency marking on file page).

Figure 6.8: Configuration 1 interpreted as an intercept-adjustment mechanism: pricing adds a predominantly flow-independent component to generalised cost, and the subsequent equilibrium adaptation can yield an aggregate regime that is equivalent (at the performance-measure level) to alternative slope–intercept specifications of the connecting link along (6.1).

The second numerical configuration yielded a different equivalent-family constraint,

$$b_5 = \frac{27261}{2090} - \frac{242}{95}\, a_5, \tag{6.2}$$

again producing invariant equilibrium performance under altered background network parameters (Section 6.5). This configuration may be read as demonstrating that the attributes of a connecting link can compensate for asymmetric background corridor conditions while preserving the equilibrium outcome.

In urban design practice, an analogous logic appears in filtered permeability and superblock strategies. Rather than optimising individual links in isolation, these interventions reorganise network permeability by constraining through-movements while maintaining local access. At the modelling level, they operate as a controlled modification of the admissible path set: some movements are removed from the feasible through-route space (or become systematically unattractive), and equilibrium re-forms on the filtered route choice set.

This correspondence aligns with Configuration 2 in two steps. First, the intervention is explicitly topological: it alters which connections are practically available for through movement, analogous to changing bypass availability or admissible path composition. Second, the system-level outcome depends on compensating adjustments outside the filtered area: perimeter operation, signal timing, and the relative attractiveness of alternative modes reshape effective costs on the remaining alternatives. Empirically, the expected

pattern is a large local reduction in through traffic and an upgraded public-space condition inside the filtered area, while wider-area performance does not deteriorate in a one-to-one manner because the equilibrium response reallocates flows, re-times trips, and diverts marginal demand. This is precisely the behavioural signature implied by the equivalence-family reading: a topology change can remain compatible with comparable aggregate performance when accompanied by appropriate parameter shifts in the remaining network.

Thus, Configuration 2 provides an analytical analogue for filtered permeability: multiple connectivity patterns—a traditional grid, a filtered grid, or a superblock-like configuration—can fall within the same equivalence family when accompanied by appropriate operational adjustments.

Superblock intervention in Sant Antoni (Barcelona): filtered permeability and local-access prioritisation.
Source: Wikimedia Commons (Author: User page).

License: CC BY-SA 4.0.

Figure 6.9: Configuration 2 interpreted as filtered permeability: superblock-like interventions restrict through-movements and reconstitute equilibrium on a filtered admissible path set. This provides an urban analogue of equivalence under combined topological change and compensating cost-parameter adjustments (cf. (6.2)).

The third numerical configuration produced the equivalent-family constraint

$$b_5 = \frac{58}{5} - \frac{9}{5}\, a_5, \tag{6.3}$$

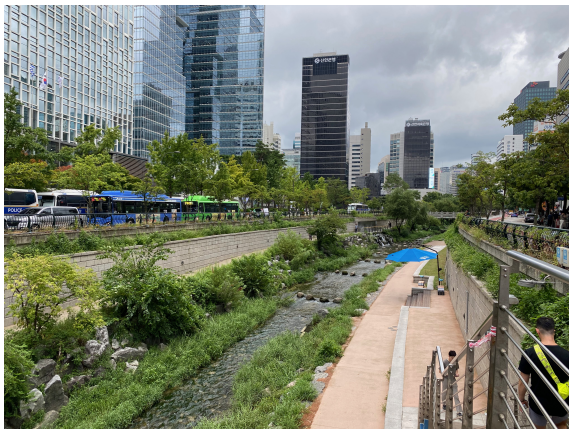
again yielding an invariant equilibrium shift $\Delta = 6$ under modified demand and heterogeneous background link properties (Section 6.5). This configuration shows that equivalence-

family invariance can persist under reduced demand and non-uniform link characteristics.

In empirical practice, close analogues are infrastructure removal and road-diet implementation. Such interventions reduce vehicular capacity or eliminate specific high-throughput connectors, while the longer-run system response is mediated by route reallocation, temporal re-timing, mode substitution, and operational changes. Within the equivalence-family interpretation, a capacity reduction increases effective congestion sensitivity on the remaining movements (a slope-like effect), but it can be offset by changes in fixed components and demand conditions across the system (intercept-like and demand-side effects), allowing the equilibrium regime to remain comparable at the aggregate level.

This is consistent with the observed logic of major removal and reallocation projects. In the Cheonggyecheon case, the removal of an auto-prioritised corridor was accompanied by network-wide redistribution and a durable rebalancing across routes and modes; the post-intervention regime is better characterised as a shift within a feasible envelope than as a proportional worsening of congestion everywhere. Similarly, road-diet reallocation (as in protected cycling corridors) reduces dedicated vehicular space on a corridor while increasing multimodal capacity and reorganising friction and priority at the cross-section level. The empirical expectation—and the condition under which such schemes are considered successful—is that network performance does not deteriorate in a one-to-one manner, because equilibrium adaptation absorbs part of the shock through rerouting, re-timing, and marginal demand suppression. This matches the structural content of (6.3): increases in slope-like congestion exposure can be compensated by changes in fixed components so that the aggregate equilibrium indicator remains stable.

Thus, Configuration 3 provides an analytical counterpart to road removal and road-diet projects. These interventions can be interpreted as moving within an equivalence family in which physical capacity is reduced while operational and behavioural parameters adjust endogenously (or through policy) to reconstitute equilibrium.



Cheonggyecheon restoration corridor (Seoul): replacement of an auto-prioritised facility by linear public space.
Source: Wikimedia Commons (Author: Sgroey).
License: CC BY-SA 4.0.

Protected bike lane on 9th Avenue (New York City): road-diet style space reallocation.
Source: Wikimedia Commons (Author: Jim.henderson).
License: CC0 1.0.

Figure 6.10: Configuration 3 interpreted as capacity reduction within an equivalence family: removing or weakening an auto-prioritised connector can be compensated by changes in effective costs and behaviour, yielding an equilibrium regime that remains comparable at the aggregate level (cf. (6.3)).

Across the three configurations and the empirical examples, a consistent relational structure emerges. Equivalent cost families in abstract networks correspond to families of spatial interventions in real cities. Whether an intervention takes the form of congestion pricing, filtered permeability, or infrastructure removal, the underlying mechanism is the same: movement within a family of equilibrium-equivalent networks, in which local parameters and admissible movements can change while the equilibrium performance indicator remains invariant.

This interpretation reframes planning practice in a design-relevant manner. Rather than evaluating a single proposed intervention as a unique causal perturbation, one can identify an equivalence class of admissible interventions that occupy the same equilibrium envelope, and then select among these spatial realisations based on environmental, social, and political criteria. In this sense, equilibrium modelling becomes not only an evaluative tool but also a generative device for defining feasible design option spaces [4, 5].

The correspondences above suggest a methodological shift in transport planning. Instead of using equilibrium models solely to assess the marginal impact of a single design proposal, planners can identify equivalence families of admissible network configurations and then explore alternative spatial realisations within these families. This supports a design-led approach in which transport modelling defines feasible mobility envelopes, while urban design selects among spatial realisations according to broader objectives—public space quality, safety, climate and health co-benefits, and distributive considerations—without abandoning network performance constraints [17, 23].

This perspective helps dissolve a persistent dichotomy: transport models are no longer external judges of design proposals, but internal instruments for constructing a structured space of feasible alternatives. The framework developed in this thesis makes explicit a logic that planning practice often applies implicitly when combining physical redesign with operational and policy measures.

6.7. Implications for Connectivity-Oriented Urban Design

The preceding sections have established a continuous conceptual chain: equivalent cost families generate network equivalence; network equivalence implies spatial non-uniqueness; numerical experiments confirm the existence and robustness of equivalence families; and empirical practice demonstrates that real-world urban interventions already operate implicitly within such equivalence spaces. This subsection synthesizes these findings into explicit implications for connectivity-oriented urban design.

Connectivity has long been treated as a primary objective in urban design. Traditional planning doctrine associates higher connectivity with improved accessibility, reduced travel time, and enhanced urban vitality. As a result, connectivity strategies often focus on increasing link density, adding new connections, or widening existing corridors. However, the equivalence framework developed in this thesis fundamentally alters this perspective. If multiple network configurations can produce identical equilibrium outcomes, then connectivity is no longer a single-valued design target but a relational property embedded in equivalence families.

This realization introduces a new paradigm: connectivity-oriented urban design should not seek to maximize or minimize connectivity in absolute terms, but to navigate equivalence spaces in which different connectivity patterns are functionally interchangeable in mobility performance yet distinct in spatial quality.

Conventional connectivity metrics—such as intersection density, link-node ratios, or average path length—implicitly assume that a unique spatial configuration corresponds to a given mobility performance. The results of this thesis demonstrate that this assumption is generally false under equilibrium assignment. Distinct topologies may yield identical equilibrium flows and travel times.

Therefore, connectivity metrics must be interpreted as coordinates in an equivalence space rather than absolute performance indicators. Two street networks with different intersection densities or permeability patterns may belong to the same equivalence family and thus perform identically in equilibrium. Conversely, networks with similar metric values may lie in different equivalence families and perform differently.

For design practice, this implies that connectivity evaluation should shift from single-metric optimization to multi-dimensional equivalence exploration. Designers should ask not “is connectivity high or low?” but “which equivalence family does this connectivity pattern belong to, and what alternative spatial realizations exist within the same family?”

The existence of equivalence families implies that equilibrium constraints do not uniquely determine spatial form. Instead, they define feasible manifolds of design solutions. Within these manifolds, multiple spatial layouts, cross-sections, and operational regimes are admissible.

This transforms the role of transport modeling in urban design. Rather than serving as a post-hoc evaluation tool that approves or rejects a proposed layout, equilibrium modeling becomes a generative instrument defining admissible design regions. Within these regions, urban designers can pursue broader objectives—public-space quality, heritage preservation, environmental performance, social equity—without violating mobility constraints.

In practical terms, this means that adding a new connection, filtering permeability, reallocating street space, introducing pricing, or optimizing signals are not separate categories of intervention. They are alternative navigations within the same equivalence landscape. Connectivity-oriented design thus becomes an exercise in selecting desirable spatial realizations from among equivalence-consistent alternatives.

Urban design practice often faces an apparent conflict: increasing connectivity to improve accessibility may degrade livability by introducing traffic intrusion, while reducing vehicular connectivity to improve livability may risk congestion. The equivalence framework resolves this tension.

If multiple connectivity patterns yield identical equilibrium outcomes, then it becomes possible to reduce vehicular permeability in sensitive urban areas while preserving overall network performance through compensatory changes elsewhere in the network or in operational parameters. Filtered permeability, superblocks, and low-traffic neighborhoods exemplify this logic: local connectivity is reduced, yet equilibrium mobility performance is preserved through perimeter reinforcement and behavioral adaptation.

Thus, livability-oriented connectivity reduction and mobility-oriented connectivity enhancement are not inherently contradictory. They are different spatial realizations within equivalence families. This insight provides a rigorous analytical foundation for contemporary urban design strategies that prioritize public space and environmental quality without sacrificing system-level accessibility.

The findings of this chapter suggest a new workflow for connectivity-oriented urban design:

(i) Define target equilibrium performance indicators (average travel time, flow distribution, congestion levels).

(ii) Use equilibrium modeling to identify equivalence families consistent with these targets.

(iii) Translate equivalence-family parameter ranges into admissible spatial design operations (geometry, topology, regulation).

(iv) Generate multiple spatial realizations within these admissible regions.

(v) Select among realizations based on urban design criteria beyond mobility.

This workflow integrates quantitative rigor with qualitative design judgment. It also reverses the conventional sequence: instead of designing first and testing later, planners identify feasible equivalence spaces first and then design within them.

Such an approach supports participatory planning, since multiple spatial alternatives can be presented as functionally equivalent in mobility terms while differing in spatial character. Stakeholders can then debate spatial qualities rather than disputing uncertain traffic forecasts.

Recognizing spatial non-uniqueness has institutional implications. Traditional planning approval processes often depend on demonstrating that a single proposed design meets traffic-performance standards. The equivalence framework suggests that regulators should instead approve ranges of admissible designs lying within equivalence families.

This would enable adaptive and incremental implementation: operational measures may be introduced first, physical construction later, or vice versa, without committing prematurely to a single irreversible spatial form. Planning governance thus becomes more flexible, resilient, and responsive to evolving urban conditions.

7 | Conclusions

This chapter concludes the thesis by synthesizing the main findings, reflecting on the methodological contributions, acknowledging the limitations of the present study, and outlining directions for future research. Rather than reiterating individual results in detail, the chapter positions the outcomes of the thesis within a broader methodological and planning-oriented perspective, revisiting the original research objectives formulated in Chapter 1.

7.1. Summary of findings

The central objective of this thesis was to reposition Braess-type equilibrium reasoning as a design-relevant framework for urban connectivity interventions. Rather than treating equilibrium modelling as a purely ex-post evaluative tool, the thesis used the Braess mechanism to clarify how spatial edits to street networks—such as adding or filtering connections, reshaping corridor hierarchy, and packaging geometric redesign with operational regulation—can reorganise network-wide routing incentives. The intended contribution is therefore not only to characterise paradoxical outcomes, but to support urban-design decision-making under explicit equilibrium constraints.

Several findings emerge from the analysis, following the thesis narrative from the classical benchmark to generalised structures and inverse, equivalence-based interpretation.

First, revisiting the classical diamond network confirms that the Braess' paradox should not be interpreted as a binary attribute of a topology, but as a conditional regime phenomenon embedded in demand and parameter space. Paradoxical deterioration appears within bounded windows, while outside these regimes the intervention may be neutral or beneficial. In design terms, this supports a scenario-based interpretation of connectivity: the same geometric action (e.g., introducing a new connector) may have qualitatively different system consequences depending on the demand regime and prevailing congestion sensitivities, and should therefore be appraised through regime mapping rather than by local intuition alone.

Second, extending the analysis across multiple cost-function families shows that the paradox mechanism persists beyond stylised linear specifications. Under nonlinear congestion formulations, including BPR-type functions that are common in applied assessment, the location and shape of paradox regimes shift, yet paradoxical patterns remain feasible. This is relevant for urban design practice because it indicates that equilibrium-sensitive reasoning about connectivity interventions is not confined to pedagogical examples; it

can be carried into modelling environments that more closely resemble empirical planning appraisal, while still requiring careful attention to how congestion nonlinearity amplifies or attenuates incentives.

Third, the comparison between deterministic user equilibrium (DUE) and stochastic user equilibrium (SUE) demonstrates that behavioural representation materially affects how paradox regimes manifest. Under DUE, route usage can change abruptly, producing sharper regime boundaries. Under SUE, dispersion smooths transitions: shortcut links tend to be deactivated more gradually as congestion rises, and paradox effects may persist over wider but typically less pronounced regions. From a design perspective, this implies that the assessment of connectivity edits is sensitive to assumptions about route-choice variability and information. Treating behavioural dispersion as a robustness dimension can therefore improve the interpretability and transferability of design conclusions.

Fourth, the progression from the diamond benchmark to ladder networks and larger structured systems shows that paradox-like effects can be structurally robust under network scaling. With multiple bypass candidates, shortcut density and placement become decisive, and interactions between repeated motifs can sustain or, in some configurations, intensify equilibrium deterioration. Interpreted spatially, this suggests that paradox risk is not merely a property of a single new link, but can arise from the way cross-connections are inserted within a hierarchical corridor structure. Design implications follow: permeability strategies benefit from being coupled with hierarchy management (where and how cross-links are provided, filtered, or regulated), rather than being framed solely as aggregate increases in connectivity.

Finally, the symbolic analysis of linear-cost settings establishes that the classical Braess example is only one instance within broader equivalence families of cost configurations that generate identical equilibrium outcomes. By constructing parameter atlases and identifying equivalence classes, the thesis reframes the paradox in an inverse, design-relevant form: instead of asking whether a specific link addition causes deterioration, one can ask which combinations of link properties (and their spatial proxies, such as cross-section allocation, access management, or pricing-like measures) are compatible with a target equilibrium effect. This perspective highlights spatial non-uniqueness under equilibrium constraints: multiple intervention packages may be functionally equivalent in mobility performance while remaining distinct in urban form, public-space quality, and multimodal priorities.

Taken together, these findings support a design-centred conclusion: the Braess' paradox operates as a structured diagnostic of how connectivity edits interact with equilibrium incentives in socio-spatial street systems, and as a basis for navigating admissible design freedoms rather than as a categorical argument for or against connectivity. The thesis thus motivates an equilibrium-informed design workflow in which

(i) performance targets and behavioural assumptions are stated explicitly,

(ii) regime boundaries and robustness dimensions are identified through parametric analysis, and

(iii) equivalence-consistent option sets are translated into alternative spatial intervention

packages that can be selected using broader urban criteria such as safety, public-space quality, multimodal integration, and distributional considerations.

7.2. Methodological contributions

Beyond the specific numerical outcomes, the primary contribution of this thesis lies in a methodological integration of equilibrium analysis that is designed to be both reproducible and transferable across modelling assumptions and network settings.

A first contribution is the development of a unified computational workflow for equilibrium-based intervention assessment. The workflow provides a consistent environment to compare before–after networks across multiple network families, cost-function specifications, and equilibrium concepts. By standardising conventions for network representation, path-set construction, demand scanning, and paradox identification, the workflow reduces the risk that observed differences arise from inconsistent implementation choices. This structure enables controlled cross-scenario comparison and supports the interpretation of paradox regimes as consequences of behavioural and cost assumptions rather than artefacts of model setup.

A second contribution is the explicit coupling of numerical exploration with analytical (symbolic) reasoning. Large-scale numerical experiments allow systematic scanning of parameter spaces and the detection of paradox regimes in settings that are difficult to analyse in closed form, including generalised network structures and nonlinear congestion specifications. Symbolic derivations, by contrast, expose the internal structure of equilibrium conditions and make transparent how families of cost parameters can generate equivalent equilibrium outcomes. Treating these two modes as complementary provides a bidirectional validation logic: numerical patterns motivate analytical interpretation, while symbolic results define equivalence relations that can be verified through targeted numerical checks. The combination improves both robustness and explanatory clarity.

A third methodological contribution concerns visualisation as an interpretive device. By constructing paradox maps and parameter atlases, equilibrium outcomes are represented as organised regions in parameter space rather than as isolated equilibrium points. This geometric representation supports comparative reasoning across demand levels, behavioural dispersion, and congestion sensitivity, and it makes regime boundaries and transitions explicit. In addition, the atlas-based approach provides a compact way to communicate how intervention effects vary systematically, thereby facilitating translation from abstract equilibrium findings to planning- and design-relevant questions.

More broadly, the thesis demonstrates that the Braess' paradox can be used as a methodological lens for equilibrium assessment of connectivity interventions. Embedding paradox detection within a structured workflow shifts attention away from singular counterexamples and toward regularities that can be inspected, compared, and reused across contexts. As a result, the thesis contributes an extensible methodological template for analysing equilibrium responses to network edits, with potential applicability to other network topologies, cost formulations, and policy-relevant intervention packages.

7.3. Limitations

Despite its scope, the present study remains subject to several limitations that arise both from the modelling framework adopted and from the specific implementation choices required to enable systematic comparison across scenarios. These limitations should be considered when interpreting the results and when translating the findings into urban-design guidance.

First, the analysis is based on static equilibrium assignment. Both deterministic and stochastic formulations describe steady-state outcomes under fixed demand and fixed network characteristics, and therefore do not represent within-day dynamics such as queue formation, spillback, departure-time choice, or transitional adjustment during staged implementation. At an implementation level, the performance comparison is conducted as a before–after equilibrium contrast under the same exogenous demand level, which is well-suited for identifying equilibrium mechanisms but does not capture the time profile of disruption, short-run learning, or reliability effects that often condition how street reallocation or access filtering is experienced in practice.

Second, the numerical experiments are restricted to a single origin–destination pair with inelastic demand. This choice supports analytical clarity and isolates the mechanism of paradoxical rerouting, yet it abstracts from multi-OD competition for shared capacity and from behavioural or economic demand responses (e.g., destination change, trip suppression, induced demand, or mode shift). At an implementation level, the demand scan therefore functions as a controlled sensitivity analysis rather than a representation of endogenous system response. In multi-market urban contexts, intervention impacts may be distributed unevenly across corridors and user groups, which cannot be inferred directly from the single-market setting.

Third, behavioural representation is necessarily simplified. The deterministic benchmark assumes perfect information and fully rational route choice, while the stochastic formulation represents dispersion through a Logit temperature parameter. Although this provides a tractable way to explore how dispersion alters paradox windows, it does not incorporate learning, habit formation, information dynamics, correlation in perception errors, or heterogeneous sensitivities across users and trip purposes. In particular, the stochastic component is implemented through closed-form choice probabilities governed by a single global dispersion parameter; randomness is therefore represented implicitly through the Logit structure rather than through explicit sampling of individual-level realisations. In the thesis, the dispersion parameter is also treated as a numerically controlled scanning variable to support stability and reproducibility, rather than being calibrated from behavioural data. This implementation choice is appropriate for comparative experiments but limits behavioural interpretability in applied design scenarios where perception, comfort, and safety can change non-uniformly across modes and populations.

Finally, the network families considered, while structurally generalised through ladder constructions, remain stylised representations of urban systems. Geometric constraints, intersection control, signal timing, curb management, multimodal interactions, and land-use feedbacks are not modelled explicitly, and link performance is represented through reduced-form cost functions rather than through detailed operational models. At an im-

plementation level, this implies that the mapping from abstract link parameters to concrete design actions (e.g., cross-section changes, filtering strategies, or pricing/access packages) remains schematic. Consequently, the results are best interpreted as equilibrium-informed diagnostics and regime-screening tools that help structure an option space for connectivity interventions, rather than as direct quantitative predictions for a specific site.

Recognising these limitations is essential for positioning the thesis contribution in urban-design terms. The analysis is most appropriately used to

(i) identify intervention types and demand regimes in which equilibrium incentives may undermine local design intentions,

(ii) support robustness-oriented comparison across behavioural and cost assumptions, and

(iii) motivate the formulation of design bundles that achieve comparable network performance while prioritising spatial quality, safety, and multimodal objectives.

7.4. Future work

The limitations identified in the preceding section motivate a set of concrete research directions that can extend the present thesis from controlled equilibrium diagnostics toward richer behavioural realism and stronger urban-design applicability. Future work is therefore organised around four corresponding avenues.

First, to address the static nature of the current assignment setting, future research should incorporate temporal dynamics and adjustment processes. Dynamic traffic assignment formulations could represent congestion buildup, queue spillback, and time-dependent bottlenecks, while day-to-day learning models could capture how route choices evolve through experience rather than instantaneously reaching equilibrium. Such extensions would enable the study of not only equilibrium end states but also transition paths, implementation phases, and reliability impacts—dimensions that are often central when evaluating street reallocation, access filtering, or staged corridor redesign in urban contexts.

Second, to move beyond the single origin–destination experimental setting, future work should extend the analysis to multi-OD systems with interacting demand markets. Multi-OD models would allow paradoxical effects to be examined at different aggregation levels (corridor, district, and network-wide) and would enable assessment of distributional consequences across user groups, trip purposes, and spatial areas. A related extension is to relax the assumption of inelastic demand by incorporating demand responses such as destination choice, departure-time choice, mode shift, or induced demand mechanisms, thereby improving the interpretability of results for city-scale design interventions where accessibility changes and behavioural adaptation are co-determined.

Third, the behavioural representation can be expanded substantially beyond the current deterministic-versus-dispersion contrast. Future research should incorporate heterogeneity in preferences and constraints, information dynamics, and correlation structures in perception errors. Instead of representing stochasticity only through a single global dispersion parameter, one direction is to develop models in which randomness is represented through

richer behavioural mechanisms (e.g., mixed discrete choice specifications, segmentation by user groups, or explicit simulation of individual-level variability), and where behavioural parameters can be calibrated or stress-tested against empirical evidence. This would be particularly valuable for urban-design applications in which perceived safety, comfort, and attractiveness vary across modes and populations and may change discontinuously with spatial form.

Fourth, to reduce the gap between stylised network families and the complexity of real urban systems, future work should strengthen the representation of spatial and operational constraints. This includes incorporating intersection control and signal timing, geometric feasibility and right-of-way constraints, curb management, and multimodal interactions (public transport priority, cycling and walking networks, and freight access). Integrating land-use feedbacks and accessibility-based behavioural responses would further support the evaluation of design interventions that aim to reshape urban structure rather than merely reallocate traffic. A pragmatic research pathway is to develop a layered modelling approach: use the present equilibrium-based regime mapping to screen candidate interventions and identify fragile regions, then apply higher-fidelity operational and spatial models to a reduced set of design alternatives.

Across these extensions, an important planning- and design-oriented research direction is the development of decision-support tools that operationalise the thesis concepts for ex-ante evaluation. Rather than optimising a single metric, such tools could identify intervention packages that are robust under uncertainty in demand, behavioural response, and operational conditions. The equivalence-class perspective developed in the thesis offers a particularly promising bridge to urban design: future work can formalise how families of equilibrium-equivalent parameterisations translate into alternative spatial configurations (cross-section allocation, permeability filtering, access regulation, and pricing-like measures) that satisfy mobility constraints while allowing designers to prioritise public-space quality, safety, and multimodal objectives.

In summary, future work can extend the thesis from a controlled equilibrium diagnostic of connectivity edits toward an integrated urban-design assessment framework. By incorporating dynamics, multi-market demand, richer behavioural realism, and spatial-operational constraints, subsequent research can strengthen both the explanatory power of Braess-type mechanisms and their practical usefulness for designing connectivity interventions that are robust, context-sensitive, and aligned with broader urban goals.

Bibliography

[1] Michael D. Meyer and Eric J. Miller. *Urban Transportation Planning: A Decision-Oriented Approach*. McGraw-Hill, 2001.

[2] Juan de Dios Ortúzar and Luis G. Willumsen. *Modelling Transport*. Wiley, 4 edition, 2011.

[3] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, 1952.

[4] Yosef Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, 1985.

[5] Ennio Cascetta. *Transportation Systems Analysis: Models and Applications*. Springer, 2009.

[6] Peter R. Stopher and Arnim H. Mayburg. *Urban Transportation Modeling and Planning*. Lexington Books, 1975.

[7] Jane Jacobs. *The Death and Life of Great American Cities*. Random House, 1961.

[8] Bill Hillier. *Space is the Machine: A Configurational Theory of Architecture*. Cambridge University Press, 1996.

[9] Stephen Marshall. *Streets & Patterns*. Spon Press, 2005.

[10] Jacques-François Thisse. The new science of cities by michael batty: The opinion of an economist. *Journal of Regional Science*, 54(3):395–399, 2014.

[11] Sergio Porta, Paolo Crucitti, and Vito Latora. The network analysis of urban streets: A primal approach. *Environment and Planning B: Planning and Design*, 33(5):705–725, 2006.

[12] Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.

[13] Phil B. Goodwin. Empirical evidence on induced traffic: A review and synthesis. *Transportation*, 23(1):35–54, 1996.

[14] Robert Cervero and Mark Hansen. Induced travel demand and induced road investment: A simultaneous equation analysis. *Journal of Transport Economics and Policy*, 36(3):469–490, 2002.

[15] Gilles Duranton and Matthew A. Turner. The fundamental law of road congestion: Evidence from u.s. cities. *American Economic Review*, 101(6):2616–2652, 2011.

[16] Susan Handy. Smart growth and the transportation–land use connection: What does the research tell us? *International Regional Science Review*, 28(2):146–167, 2005.

[17] Reid Ewing and Robert Cervero. Travel and the built environment: A meta-analysis. *Journal of the American Planning Association*, 76(3):265–294, 2010.

[18] Sally Cairns, Stephen Atkins, and Phil Goodwin. Disappearing traffic? the story so far. *Proceedings of the Institution of Civil Engineers: Municipal Engineer*, 151(1): 13–22, 2002. doi: 10.1680/muen.151.1.13.38856.

[19] National Association of City Transportation Officials. *Urban Street Design Guide*. Island Press, 2013.

[20] Jan Gehl. *Cities for People*. Island Press, 2010.

[21] Dietrich Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12(1):258–268, 1968.

[22] J. D. Murchland. Braess's paradox of traffic flow. *Transportation Research*, 4(4): 391–394, 1970. doi: 10.1016/0041-1647(70)90196-6.

[23] David Banister. The sustainable mobility paradigm. *Transport Policy*, 15(2):73–80, 2008.

[24] Peter Newman and Jeffrey Kenworthy. *Sustainability and Cities: Overcoming Automobile Dependence*. Island Press, 1999.

[25] UN-Habitat. Planning and design for sustainable urban mobility: Global report on human settlements 2013. Technical report, United Nations Human Settlements Programme (UN-Habitat), 2013.

[26] Carlos F. Daganzo and Yosef Sheffi. On stochastic models of traffic assignment. *Transportation Science*, 11(3):253–274, 1977.

[27] Caroline Fisk. Some developments in equilibrium traffic assignment. *Transportation Research Part B: Methodological*, 14(3):243–255, 1980.

[28] Chunxue Zhao, Baibai Fu, and Tianming Wang. Braess paradox and robustness of traffic networks under stochastic user equilibrium. *Transportation Research Part E: Logistics and Transportation Review*, 61:135–141, 2014. doi: 10.1016/j.tre.2013.11.008.

[29] Wei Wang, David Z. W. Wang, Huijun Sun, Zengzhe Feng, and Jianjun Wu. Braess paradox of traffic networks with mixed equilibrium behaviors. *Transportation Research Part B: Methodological*, 111:136–154, 2018.

[30] Richard Steinberg and Willard I. Zangwill. The prevalence of braess' paradox. *Transportation Science*, 17(3):301–318, 1983. doi: 10.1287/trsc.17.3.301.

[31] Tim Roughgarden. On the severity of braess's paradox: Designing networks for selfish users is hard. *Journal of Computer and System Sciences*, 72(5):922–953, 2006. doi: 10.1016/j.jcss.2005.05.009.

[32] Igal Milchtaich. Network topology and the efficiency of equilibrium. *Games and Economic Behavior*, 57(2):321–346, 2005.

[33] John Morgan, Henrik Orzen, and Martin Sefton. Network architecture and traffic flows: Experiments on the pigou–knight–downs and braess paradoxes. *Games and Economic Behavior*, 66(1):348–372, 2009.

[34] Petter Holme and Beom Jun Kim. Attack vulnerability of complex networks. *Physical Review E*, 65:056109, 2002.

[35] Anna Nagurney and Qiang Qiang. A network efficiency measure with application to critical infrastructure networks. *Journal of Global Optimization*, 40(1-3):261–275, 2008.

[36] Claude M. Penchina and Leora J. Penchina. The braess paradox in mechanical, traffic, and other networks. *American Journal of Physics*, 71(5):479–482, 2003.

[37] Adilson E. Motter. Improved network performance via antagonism: From synthetic rescues to multi-drug combinations. *BioEssays*, 32(3):236–245, 2010.

[38] Terry L. Friesz. *Dynamic Optimization and Differential Games*. Springer, 2010.

[39] Giulio E. Cantarella. Dynamic processes and equilibrium in transportation networks: Towards a unifying theory. *Transportation Science*, 31(4):275–291, 1997.

[40] Dingyi Zhuang, Yuzhu Huang, Vindula Jayawardana, Jinhua Zhao, Dajiang Suo, and Cathy Wu. The braess's paradox in dynamic traffic. *Transportation Research Part B: Methodological*, 170:1–25, 2023.

[41] Stefan Bittihn and Andreas Schadschneider. Braess paradox in a network with stochastic dynamics and fixed strategies. *Physica A: Statistical Mechanics and its Applications*, 417:190–200, 2015.

[42] Hani S. Mahmassani and Gang-Len Chang. On boundedly rational user equilibrium in transportation systems. *Transportation Science*, 21(2):89–99, 1987.

[43] Jiayang Li, Zhaoran Wang, and Yu (Marco) Nie. Wardrop equilibrium can be boundedly rational: A new behavioral theory of route choice. *Transportation Research Part B: Methodological*, 132:218–237, 2020.

[44] Hai Yang and Michael G. H. Bell. Models and algorithms for road network design: A review and some new developments. *Transport Reviews*, 18(3):257–278, 1998.

[45] Dietrich Braess, Anna Nagurney, and Tina Wakolbinger. On a paradox of traffic planning. *Transportation Science*, 39(4):446–450, 2005.

[46] William S. Vickrey. Congestion theory and transport investment. *American Economic Review*, 59(2):251–260, 1969.

[47] Eric I Pas and Shari L Principio. Braess' paradox: Some new insights. *Transportation Research Part B: Methodological*, 31(3):265–276, 1997.

[48] Tim Roughgarden and Éva Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.

[49] John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[50] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1563 of *Lecture Notes in Computer Science*, pages 404–413. Springer, 1999.

[51] Henry C. Lin, Tim Roughgarden, Éva Tardos, and Asher Walkover. Stronger bounds on braess's paradox and the maximum latency of selfish routing. *SIAM Journal on Discrete Mathematics*, 25(4):1667–1686, 2011.

[52] Tim Roughgarden. *Selfish Routing and the Price of Anarchy*. The MIT Press, Cambridge, MA, 2005. ISBN 9780262182430.

[53] Hyejin Youn, Michael T. Gastner, and Hawoong Jeong. Price of anarchy in transportation networks: Efficiency and optimality control. *Physical Review Letters*, 101 (12):128701, 2008. doi: 10.1103/PhysRevLett.101.128701.

[54] Michael Patriksson. Sensitivity analysis of traffic equilibria. *Transportation Science*, 38(3):258–281, 2004. doi: 10.1287/trsc.1030.0043.

[55] M. J. Smith. The existence, uniqueness and stability of traffic equilibria. *Transportation Research Part B: Methodological*, 13(4):295–304, 1979.

[56] Martin J. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, New Haven, 1956.

[57] Robert B. Dial. A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research*, 5(2):83–111, 1971.

[58] Jia Yao, Zhanhong Cheng, Jingtong Dai, Anthony Chen, and Shi An. Traffic assignment paradox incorporating congestion and stochastic perceived error simultaneously. *Transportation Research Part B: Methodological*, 132:1–22, 2020.

[59] Joel E. Cohen and Paul Horowitz. Paradoxical behaviour of mechanical and electrical networks. *Nature*, 352:699–701, 1991.

[60] Dirk Witthaut and Marc Timme. Braess's paradox in oscillator networks, desynchronization and power outage. *New Journal of Physics*, 14:083036, 2012.

[61] Rahmi Akçelik. Travel time functions for transport planning purposes: Davidson's function, its time-dependent form and an alternative travel time function. *Australian Road Research*, 21(3):49–59, 1991.

[62] Norihito Toyota and Fumiho Ogura. Braess like paradox on ladder network: Cases with two way bypasses, 2015. arXiv preprint arXiv:1501.02097v1.

[63] ITDP and EPC. Better streets, better cities: A guide to street design in urban india. Technical report, Institute for Transportation and Development Policy (ITDP) and Environmental Planning Collaborative (EPC), 2011.

[64] Michael Florian and Donald Hearn. Network equilibrium models and algorithms. In *Handbook of Transportation Science*, pages 485–550. Springer, 1995.

[65] Markos Papageorgiou. Some remarks on macroscopic traffic flow modelling. *Transportation Research Part A: Policy and Practice*, 32(5):323–329, 1998.

[66] Anna Nagurney. Congested urban transportation networks and emission paradoxes. *Transportation Research Part D: Transport and Environment*, 5(2):145–151, 2000.

[67] David P. Watling. Asymmetric problems and stochastic process models of traffic assignment. *Transportation Research Part B: Methodological*, 30(5):339–357, 1996. doi: 10.1016/0191-2615(96)00006-9.

[68] José R. Correa, Andreas S. Schulz, and Nicolás E. Stier-Moses. A geometric approach to the price of anarchy in nonatomic congestion games. *Games and Economic Behavior*, 50(1):1–24, 2004.

[69] Mogens Fosgerau, Mads Paulsen, and Thomas Kjær Rasmussen. A perturbed utility route choice model. *Transportation Research Part C: Emerging Technologies*, 136: 103514, 2022. doi: 10.1016/j.trc.2021.103514.

[70] Daron Acemoglu, Ali Makhdoumi, Azarakhsh Malekian, and Asuman Ozdaglar. Informational braess' paradox: The effect of information on traffic congestion. *Operations Research*, 66(2):427–444, 2018.

[71] Yannis A. Korilis, Aurel A. Lazar, and Ariel Orda. Avoiding the braess paradox in non-cooperative networks. *Journal of Applied Probability*, 36(1):211–222, 1999.

[72] Larry J. LeBlanc, Edward K. Morlok, and William P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, 9(5):309–318, 1975. doi: 10.1016/0041-1647(75)90030-1.

[73] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. doi: 10.1002/nav.3800030109.

[74] Anna Nagurney. *Network Economics: A Variational Inequality Approach*. Springer, Berlin, 1993.

[75] Stella C. Dafermos. The traffic assignment problem for multiclass-user transportation networks. *Transportation Science*, 6(1):73–87, 1972. doi: 10.1287/trsc.6.1.73.

[76] Michiel C. J. Bliemer and Piet H. L. Bovy. Quasi-variational inequality formulation of the multiclass dynamic traffic assignment problem. *Transportation Research Part B: Methodological*, 37(6):501–519, 2003. doi: 10.1016/S0191-2615(02)00025-5.

Acknowledgements

This Master's thesis was completed under the attentive supervision of Prof. Lorenzo Mussone and Prof. Roberto Notari. I am deeply grateful for their continuous guidance and support throughout the entire process—from the selection of the research topic and the development of the theoretical framework to the refinement of the methodology and the writing and revision of the manuscript. Their rigorous academic standards, insightful advice, and patient encouragement have greatly strengthened my research training and academic writing. I would like to express my sincere gratitude and highest respect to both supervisors.

I would also like to thank all the teachers and fellow students who offered support during the completion of this thesis. Their helpful discussions, suggestions, and assistance in accessing references and advancing the research have been invaluable. Any remaining limitations or errors in this thesis are solely my own responsibility.

Finally, I am sincerely grateful to my family and friends for their understanding, tolerance, and encouragement throughout my studies. Their constant support and companionship have provided essential motivation for me to complete this thesis.

A | Tables and Supplementary Results

Ordering of LinkFlow in Appendix A

In Appendix A, the vectors LinkFlow(B) and LinkFlow(A) are directed link-flow vectors. The j-th entry reports the equilibrium flow on the j-th directed link in the corresponding network. LinkFlow(B) follows the ordering of the before link list, and LinkFlow(A) follows the ordering of the after link list. If a bypass is bidirectional, the two directions appear as two separate directed links.

Diamond network (nodes 1–4, $r = 1$, $s = 4$)

The BEFORE network contains the four directed links

$$(1 \to 2),\ (2 \to 4),\ (1 \to 3),\ (3 \to 4).$$

The AFTER network appends the added middle link $(2 \to 3)$ (and only if a two-way option is used, its reverse $(3 \to 2)$).

Table A.1: Diamond network: index-to-link mapping for LinkFlow(B) and LinkFlow(A).

Index	Directed link $u \to v$	Remark
1	$1 \to 2$	upper-left branch
2	$2 \to 4$	upper-right branch
3	$1 \to 3$	lower-left branch
4	$3 \to 4$	lower-right branch
5	$2 \to 3$	added link (after network)
6	$3 \to 2$	only if the added link is two-way

For the standard one-way diamond used in Appendix A tables,

$$\text{LinkFlow(B)} = [x_{1\to2}, x_{2\to4}, x_{1\to3}, x_{3\to4}], \quad \text{LinkFlow(A)} = [x_{1\to2}, x_{2\to4}, x_{1\to3}, x_{3\to4}, x_{2\to3}].$$

Ladder network (unified numbering, $n \geq 1$)

Node numbering is

$$U(i) = i, \quad L(i) = i + (n+1), \quad r = 2n+3, \quad s = 2n+4, \quad i = 1, \ldots, n+1.$$

The BEFORE ordering (thus LinkFlow(B)) is always:

$$\begin{aligned}
&(r \to U(1)),\ (r \to L(1)),\\
&(U(1) \to U(2)), \ldots, (U(n) \to U(n+1)),\\
&(L(1) \to L(2)), \ldots, (L(n) \to L(n+1)),\\
&(U(n+1) \to s),\ (L(n+1) \to s).
\end{aligned}$$

Hence LinkFlow(B) has length $2n+4$.

For the AFTER network, the same base links appear first, and then each activated bypass position listed in the table as BypassPos is appended in that listed order. A bypass at position i connects $U(i)$ and $L(i)$. If the bypass is bidirectional, the appended directed links are

$$(U(i) \to L(i)) \quad \text{and} \quad (L(i) \to U(i)).$$

Therefore, if $k = \text{NumBypassPos}$ bypass positions are activated and bypasses are bidirectional, LinkFlow(A) has length $2n + 4 + 2k$.

$n = 2$ explicit mapping

For $n = 2$: $U = \{1, 2, 3\}$, $L = \{4, 5, 6\}$, $(r, s) = (7, 8)$. Candidate bypass positions are $i \in \{1, 2, 3\}$, corresponding to the bidirectional pairs $1 \leftrightarrow 4$, $2 \leftrightarrow 5$, and $3 \leftrightarrow 6$.

The BEFORE ordering (length 8) is

$$\begin{aligned}
&(7 \to 1),\ (7 \to 4),\ (1 \to 2),\ (2 \to 3),\\
&(4 \to 5),\ (5 \to 6),\ (3 \to 8),\ (6 \to 8).
\end{aligned}$$

If BypassPos=[1] is activated (bidirectional), the appended AFTER links are

$$(1 \to 4),\ (4 \to 1).$$

If BypassPos=[2] is activated, the appended AFTER links are

$$(2 \to 5),\ (5 \to 2).$$

If BypassPos=[3] is activated, the appended AFTER links are

$$(3 \to 6),\ (6 \to 3).$$

If multiple positions are activated, the appended links follow the order printed in BypassPos.

$n = 3$ explicit mapping

For $n = 3$: $U = \{1, 2, 3, 4\}$, $L = \{5, 6, 7, 8\}$, $(r, s) = (9, 10)$. Candidate bypass positions are $i \in \{1, 2, 3, 4\}$, corresponding to the bidirectional pairs $1 \leftrightarrow 5$, $2 \leftrightarrow 6$, $3 \leftrightarrow 7$, and $4 \leftrightarrow 8$.

The BEFORE ordering (length 10) is

$$(9 \to 1),\ (9 \to 5),\ (1 \to 2),\ (2 \to 3),\ (3 \to 4),$$
$$(5 \to 6),\ (6 \to 7),\ (7 \to 8),\ (4 \to 10),\ (8 \to 10).$$

If BypassPos=[1 2 4] is activated with bidirectional bypasses, the appended AFTER links are, in order,

$$(1 \to 5),\ (5 \to 1),\ (2 \to 6),\ (6 \to 2),$$
$$(4 \to 8),\ (8 \to 4).$$

Special transition case: ladder $n = 0$

For $n = 0$: $U(1) = 1$, $L(1) = 2$, $(r, s) = (3, 4)$. The BEFORE ordering (length 4) is

$$(3 \to 1),\ (3 \to 2),\ (1 \to 4),\ (2 \to 4).$$

Any added bypass between nodes (u, v) is appended after these base links. If the bypass is bidirectional, both $(u \to v)$ and $(v \to u)$ appear as separate entries in LinkFlow(A).

A.1. Supplementary table for Section 5.1

Table A.2: Classical benchmark scan results for $Q = 1, \ldots, 20$ (Keeping at most three decimal places).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	55.5	31	−24.5	false	[0.5 0.5 0.5 0.5]	[1 0 0 1 1]
2	61	52	−9	false	[1 1 1 1]	[2 0 0 2 2]
3	66.5	73	6.5	true	[1.5 1.5 1.5 1.5]	[3 0 0 3 3]
4	72	87.197	15.197	true	[2 2 2 2]	[3.691 0.309 0.309 3.691 3.381]
5	77.5	89.614	12.114	true	[2.5 2.5 2.5 2.5]	[3.846 1.154 1.154 3.846 2.692]
6	83	92	9	true	[3 3 3 3]	[4 2 2 4 2]
7	88.5	94.384	5.884	true	[3.5 3.5 3.5 3.5]	[4.154 2.846 2.846 4.154 1.308]
8	94	96.769	2.769	true	[4 4 4 4]	[4.308 3.692 3.692 4.308 0.615]
9	99.5	99.5	0	false	[4.5 4.5 4.5 4.5]	[4.5 4.5 4.5 4.5 0]
10	105	105	0	false	[5 5 5 5]	[5 5 5 5 0]
11	110.5	110.5	0	false	[5.5 5.5 5.5 5.5]	[5.5 5.5 5.5 5.5 0]
12	116	116	0	false	[6 6 6 6]	[6 6 6 6 0]
13	121.5	121.5	0	false	[6.5 6.5 6.5 6.5]	[6.5 6.5 6.5 6.5 0]
14	127	127	0	false	[7 7 7 7]	[7 7 7 7 0]
15	132.5	132.5	0	false	[7.5 7.5 7.5 7.5]	[7.5 7.5 7.5 7.5 0]
16	138	138	0	false	[8 8 8 8]	[8 8 8 8 0]
17	143.5	143.5	0	false	[8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 0]
18	149	149	0	false	[9 9 9 9]	[9 9 9 9 0]
19	154.5	154.5	0	false	[9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 0]
20	160	160	0	false	[10 10 10 10]	[10 10 10 10 0]

A.2. Supplementary table for Section 5.2

Table A.3: Case 1: linear+const: scan results for $Q = 1, \ldots, 50$ (deterministic user equilibrium).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	105	30	−75	false	[0.5 0.5 0.5 0.5]	[1 0 0 1 1]
2	110	60	−50	false	[1 1 1 1]	[2 0 0 2 2]
3	115	90	−25	false	[1.5 1.5 1.5 1.5]	[3 0 0 3 3]
4	120	120	0	false	[2 2 2 2]	[4 0 0 4 4]
5	125	150	25	true	[2.5 2.5 2.5 2.5]	[5 0 0 5 5]
6	130	153.333	23.333	true	[3 3 3 3]	[5.333 0.667 0.667 5.333 4.667]
7	135	156.667	21.667	true	[3.5 3.5 3.5 3.5]	[5.667 1.333 1.333 5.667 4.333]
8	140	160	20	true	[4 4 4 4]	[6 2 2 6 4]
9	145	163.333	18.333	true	[4.5 4.5 4.5 4.5]	[6.333 2.667 2.667 6.333 3.667]
10	150	166.667	16.667	true	[5 5 5 5]	[6.667 3.333 3.333 6.667 3.333]
11	155	170	15	true	[5.5 5.5 5.5 5.5]	[7 4 4 7 3]
12	160	173.333	13.333	true	[6 6 6 6]	[7.333 4.667 4.667 7.333 2.667]
13	165	176.667	11.667	true	[6.5 6.5 6.5 6.5]	[7.667 5.333 5.333 7.667 2.333]
14	170	180	10	true	[7 7 7 7]	[8 6 6 8 2]
15	175	183.333	8.333	true	[7.5 7.5 7.5 7.5]	[8.333 6.667 6.667 8.333 1.667]

Continued on next page

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
16	180	186.667	6.667	true	[8 8 8 8]	[8.667 7.333 7.333 8.667 1.333]
17	185	190	5	true	[8.5 8.5 8.5 8.5]	[9 8 8 9 1]
18	190	193.333	3.333	true	[9 9 9 9]	[9.333 8.667 8.667 9.333 0.667]
19	195	196.667	1.667	true	[9.5 9.5 9.5 9.5]	[9.667 9.333 9.333 9.667 0.333]
20	200	200	0	false	[10 10 10 10]	[10 10 10 10 0]
21	205	205	0	false	[10.5 10.5 10.5 10.5]	[10.5 10.5 10.5 10.5 0]
22	210	210	0	false	[11 11 11 11]	[11 11 11 11 0]
23	215	215	0	false	[11.5 11.5 11.5 11.5]	[11.5 11.5 11.5 11.5 0]
24	220	220	0	false	[12 12 12 12]	[12 12 12 12 0]
25	225	225	0	false	[12.5 12.5 12.5 12.5]	[12.5 12.5 12.5 12.5 0]
26	230	230	0	false	[13 13 13 13]	[13 13 13 13 0]
27	235	235	0	false	[13.5 13.5 13.5 13.5]	[13.5 13.5 13.5 13.5 0]
28	240	240	0	false	[14 14 14 14]	[14 14 14 14 0]
29	245	245	0	false	[14.5 14.5 14.5 14.5]	[14.5 14.5 14.5 14.5 0]
30	250	250	0	false	[15 15 15 15]	[15 15 15 15 0]
31	255	255	0	false	[15.5 15.5 15.5 15.5]	[15.5 15.5 15.5 15.5 0]
32	260	260	0	false	[16 16 16 16]	[16 16 16 16 0]
33	265	265	0	false	[16.5 16.5 16.5 16.5]	[16.5 16.5 16.5 16.5 0]
34	270	270	0	false	[17 17 17 17]	[17 17 17 17 0]
35	275	275	0	false	[17.5 17.5 17.5 17.5]	[17.5 17.5 17.5 17.5 0]
36	280	280	0	false	[18 18 18 18]	[18 18 18 18 0]
37	285	285	0	false	[18.5 18.5 18.5 18.5]	[18.5 18.5 18.5 18.5 0]
38	290	290	0	false	[19 19 19 19]	[19 19 19 19 0]
39	295	295	0	false	[19.5 19.5 19.5 19.5]	[19.5 19.5 19.5 19.5 0]
40	300	300	0	false	[20 20 20 20]	[20 20 20 20 0]
41	305	305	0	false	[20.5 20.5 20.5 20.5]	[20.5 20.5 20.5 20.5 0]
42	310	310	0	false	[21 21 21 21]	[21 21 21 21 0]
43	315	315	0	false	[21.5 21.5 21.5 21.5]	[21.5 21.5 21.5 21.5 0]
44	320	320	0	false	[22 22 22 22]	[22 22 22 22 0]
45	325	325	0	false	[22.5 22.5 22.5 22.5]	[22.5 22.5 22.5 22.5 0]
46	330	330	0	false	[23 23 23 23]	[23 23 23 23 0]
47	335	335	0	false	[23.5 23.5 23.5 23.5]	[23.5 23.5 23.5 23.5 0]
48	340	340	0	false	[24 24 24 24]	[24 24 24 24 0]
49	345	345	0	false	[24.5 24.5 24.5 24.5]	[24.5 24.5 24.5 24.5 0]
50	350	350	0	false	[25 25 25 25]	[25 25 25 25 0]

Table A.4: Case 2: quadratic+const: scan results for $Q = 1, \ldots, 50$ (deterministic user equilibrium).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	85	12.5	−72.5	false	[0.5 0.5 0.5 0.5]	[1 0 0 1 1]
2	100	90	−10	false	[1 1 1 1]	[2 0 0 2 2]
3	122.5	175.167	52.667	true	[1.5 1.5 1.5 1.5]	[2.483 0.517 0.517 2.483 1.967]
4	160	202.543	42.543	true	[2 2 2 2]	[3.113 0.887 0.887 3.113 2.226]
5	212.5	240	27.497	true	[2.5 2.5 2.5 2.5]	[3.618 1.382 1.382 3.618 2.237]
6	280	287.547	7.547	true	[3 3 3 3]	[3.981 2.019 2.019 3.981 1.962]
7	362.5	362.5	0	false	[3.5 3.5 3.5 3.5]	[3.5 3.5 3.5 3.5 0]
8	460	460	0	false	[4 4 4 4]	[4 4 4 4 0]
9	572.5	572.5	0	false	[4.5 4.5 4.5 4.5]	[4.5 4.5 4.5 4.5 0]
10	700	700	0	false	[5 5 5 5]	[5 5 5 5 0]
11	842.5	842.5	0	false	[5.5 5.5 5.5 5.5]	[5.5 5.5 5.5 5.5 0]
12	1000	1000	0	false	[6 6 6 6]	[6 6 6 6 0]
13	1172.5	1172.5	0	false	[6.5 6.5 6.5 6.5]	[6.5 6.5 6.5 6.5 0]
14	1360	1360	0	false	[7 7 7 7]	[7 7 7 7 0]
15	1562.5	1562.5	0	false	[7.5 7.5 7.5 7.5]	[7.5 7.5 7.5 7.5 0]
16	1780	1780	0	false	[8 8 8 8]	[8 8 8 8 0]
17	2012.5	2012.5	0	false	[8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 0]
18	2260	2260	0	false	[9 9 9 9]	[9 9 9 9 0]
19	2522.5	2522.5	0	false	[9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 0]

Continued on next page

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
20	2800	2800	0	false	[10 10 10 10]	[10 10 10 10 0]
21	3092.5	3092.5	0	false	[10.5 10.5 10.5 10.5]	[10.5 10.5 10.5 10.5 0]
22	3400	3400	0	false	[11 11 11 11]	[11 11 11 11 0]
23	3722.5	3722.5	0	false	[11.5 11.5 11.5 11.5]	[11.5 11.5 11.5 11.5 0]
24	4060	4060	0	false	[12 12 12 12]	[12 12 12 12 0]
25	4412.5	4412.5	0	false	[12.5 12.5 12.5 12.5]	[12.5 12.5 12.5 12.5 0]
26	4780	4780	0	false	[13 13 13 13]	[13 13 13 13 0]
27	5162.5	5162.5	0	false	[13.5 13.5 13.5 13.5]	[13.5 13.5 13.5 13.5 0]
28	5560	5560	0	false	[14 14 14 14]	[14 14 14 14 0]
29	5972.5	5972.5	0	false	[14.5 14.5 14.5 14.5]	[14.5 14.5 14.5 14.5 0]
30	6400	6400	0	false	[15 15 15 15]	[15 15 15 15 0]
31	6842.5	6842.5	0	false	[15.5 15.5 15.5 15.5]	[15.5 15.5 15.5 15.5 0]
32	7300	7300	0	false	[16 16 16 16]	[16 16 16 16 0]
33	7772.5	7772.5	0	false	[16.5 16.5 16.5 16.5]	[16.5 16.5 16.5 16.5 0]
34	8260	8260	0	false	[17 17 17 17]	[17 17 17 17 0]
35	8762.5	8762.5	0	false	[17.5 17.5 17.5 17.5]	[17.5 17.5 17.5 17.5 0]
36	9280	9280	0	false	[18 18 18 18]	[18 18 18 18 0]
37	9812.5	9812.5	0	false	[18.5 18.5 18.5 18.5]	[18.5 18.5 18.5 18.5 0]
38	10 360	10 360	0	false	[19 19 19 19]	[19 19 19 19 0]
39	10 922.5	10 922.5	0	false	[19.5 19.5 19.5 19.5]	[19.5 19.5 19.5 19.5 0]
40	11 500	11 500	0	false	[20 20 20 20]	[20 20 20 20 0]
41	12 092.5	12 092.5	0	false	[20.5 20.5 20.5 20.5]	[20.5 20.5 20.5 20.5 0]
42	12 700	12 700	0	false	[21 21 21 21]	[21 21 21 21 0]
43	13 322.5	13 322.5	0	false	[21.5 21.5 21.5 21.5]	[21.5 21.5 21.5 21.5 0]
44	13 960	13 960	0	false	[22 22 22 22]	[22 22 22 22 0]
45	14 612.5	14 612.5	0	false	[22.5 22.5 22.5 22.5]	[22.5 22.5 22.5 22.5 0]
46	15 280	15 280	0	false	[23 23 23 23]	[23 23 23 23 0]
47	15 962.5	15 962.5	0	false	[23.5 23.5 23.5 23.5]	[23.5 23.5 23.5 23.5 0]
48	16 660	16 660	0	false	[24 24 24 24]	[24 24 24 24 0]
49	17 372.5	17 372.5	0	false	[24.5 24.5 24.5 24.5]	[24.5 24.5 24.5 24.5 0]
50	18 100	18 100	0	false	[25 25 25 25]	[25 25 25 25 0]

Table A.5: Case 3: all quadratic: scan results for $Q = 1, \ldots, 50$ (deterministic user equilibrium).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	2.5	2.5	0	false	[0.5 0.5 0.5 0.5]	[0.5 0.5 0.5 0.5 0]
2	10	10	0	false	[1 1 1 1]	[1 1 1 1 0]
3	22.5	22.5	0	false	[1.5 1.5 1.5 1.5]	[1.5 1.5 1.5 1.5 0]
4	40	40	0	false	[2 2 2 2]	[2 2 2 2 0]
5	62.5	62.5	0	false	[2.5 2.5 2.5 2.5]	[2.5 2.5 2.5 2.5 0]
6	90	90	0	false	[3 3 3 3]	[3 3 3 3 0]
7	122.5	122.5	0	false	[3.5 3.5 3.5 3.5]	[3.5 3.5 3.5 3.5 0]
8	160	160	0	false	[4 4 4 4]	[4 4 4 4 0]
9	202.5	202.5	0	false	[4.5 4.5 4.5 4.5]	[4.5 4.5 4.5 4.5 0]
10	250	250	0	false	[5 5 5 5]	[5 5 5 5 0]
11	302.5	302.5	0	false	[5.5 5.5 5.5 5.5]	[5.5 5.5 5.5 5.5 0]
12	360	360	0	false	[6 6 6 6]	[6 6 6 6 0]
13	422.5	422.5	0	false	[6.5 6.5 6.5 6.5]	[6.5 6.5 6.5 6.5 0]
14	490	490	0	false	[7 7 7 7]	[7 7 7 7 0]
15	562.5	562.5	0	false	[7.5 7.5 7.5 7.5]	[7.5 7.5 7.5 7.5 0]
16	640	640	0	false	[8 8 8 8]	[8 8 8 8 0]
17	722.5	722.5	0	false	[8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 0]
18	810	810	0	false	[9 9 9 9]	[9 9 9 9 0]
19	902.5	902.5	0	false	[9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 0]
20	1000	1000	0	false	[10 10 10 10]	[10 10 10 10 0]
21	1102.5	1102.5	0	false	[10.5 10.5 10.5 10.5]	[10.5 10.5 10.5 10.5 0]
22	1210	1210	0	false	[11 11 11 11]	[11 11 11 11 0]
23	1322.5	1322.5	0	false	[11.5 11.5 11.5 11.5]	[11.5 11.5 11.5 11.5 0]

Continued on next page

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
24	1440	1440	0	false	[12 12 12 12]	[12 12 12 12 0]
25	1562.5	1562.5	0	false	[12.5 12.5 12.5 12.5]	[12.5 12.5 12.5 12.5 0]
26	1690	1690	0	false	[13 13 13 13]	[13 13 13 13 0]
27	1822.5	1822.5	0	false	[13.5 13.5 13.5 13.5]	[13.5 13.5 13.5 13.5 0]
28	1960	1960	0	false	[14 14 14 14]	[14 14 14 14 0]
29	2102.5	2102.5	0	false	[14.5 14.5 14.5 14.5]	[14.5 14.5 14.5 14.5 0]
30	2250	2250	0	false	[15 15 15 15]	[15 15 15 15 0]
31	2402.5	2402.5	0	false	[15.5 15.5 15.5 15.5]	[15.5 15.5 15.5 15.5 0]
32	2560	2560	0	false	[16 16 16 16]	[16 16 16 16 0]
33	2722.5	2722.5	0	false	[16.5 16.5 16.5 16.5]	[16.5 16.5 16.5 16.5 0]
34	2890	2890	0	false	[17 17 17 17]	[17 17 17 17 0]
35	3062.5	3062.5	0	false	[17.5 17.5 17.5 17.5]	[17.5 17.5 17.5 17.5 0]
36	3240	3240	0	false	[18 18 18 18]	[18 18 18 18 0]
37	3422.5	3422.5	0	false	[18.5 18.5 18.5 18.5]	[18.5 18.5 18.5 18.5 0]
38	3610	3610	0	false	[19 19 19 19]	[19 19 19 19 0]
39	3802.5	3802.5	0	false	[19.5 19.5 19.5 19.5]	[19.5 19.5 19.5 19.5 0]
40	4000	4000	0	false	[20 20 20 20]	[20 20 20 20 0]
41	4202.5	4202.5	0	false	[20.5 20.5 20.5 20.5]	[20.5 20.5 20.5 20.5 0]
42	4410	4410	0	false	[21 21 21 21]	[21 21 21 21 0]
43	4622.5	4622.5	0	false	[21.5 21.5 21.5 21.5]	[21.5 21.5 21.5 21.5 0]
44	4840	4840	0	false	[22 22 22 22]	[22 22 22 22 0]
45	5062.5	5062.5	0	false	[22.5 22.5 22.5 22.5]	[22.5 22.5 22.5 22.5 0]
46	5290	5290	0	false	[23 23 23 23]	[23 23 23 23 0]
47	5522.5	5522.5	0	false	[23.5 23.5 23.5 23.5]	[23.5 23.5 23.5 23.5 0]
48	5760	5760	0	false	[24 24 24 24]	[24 24 24 24 0]
49	6002.5	6002.5	0	false	[24.5 24.5 24.5 24.5]	[24.5 24.5 24.5 24.5 0]
50	6250	6250	0	false	[25 25 25 25]	[25 25 25 25 0]

Table A.6: Case 4: linear+quadratic mix: scan results for $Q = 1, \ldots, 50$ (deterministic user equilibrium).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	55.5	55.5	0	false	[0.5 0.5 0.5 0.5]	[0.5 0.5 0.5 0.5 0]
2	61	61	0	false	[1 1 1 1]	[1 1 1 1 0]
3	66.5	66.5	0	false	[1.5 1.5 1.5 1.5]	[1.5 1.5 1.5 1.5 0]
4	72	72	0	false	[2 2 2 2]	[2 2 2 2 0]
5	77.5	77.5	0	false	[2.5 2.5 2.5 2.5]	[2.5 2.5 2.5 2.5 0]
6	83	83	0	false	[3 3 3 3]	[3 3 3 3 0]
7	88.5	88.5	0	false	[3.5 3.5 3.5 3.5]	[3.5 3.5 3.5 3.5 0]
8	94	90.925	−3.075	false	[4 4 4 4]	[4.308 3.692 3.692 4.308 0.615]
9	99.5	91.964	−7.536	false	[4.5 4.5 4.5 4.5]	[4.615 4.385 4.385 4.615 0.231]
10	105	92.5	−12.5	false	[5 5 5 5]	[5 5 5 5 0]
11	110.5	93.049	−17.451	false	[5.5 5.5 5.5 5.5]	[5.333 5.667 5.667 5.333 0.333]
12	116	93.611	−22.389	false	[6 6 6 6]	[5.667 6.333 6.333 5.667 0.667]
13	121.5	94.185	−27.315	false	[6.5 6.5 6.5 6.5]	[6 7 7 6 1]
14	127	94.771	−32.229	false	[7 7 7 7]	[6.333 7.667 7.667 6.333 1.333]
15	132.5	95.37	−37.130	false	[7.5 7.5 7.5 7.5]	[6.667 8.333 8.333 6.667 1.667]
16	138	95.982	−42.019	false	[8 8 8 8]	[7 9 9 7 2]
17	143.5	96.606	−46.894	false	[8.5 8.5 8.5 8.5]	[7.333 9.667 9.667 7.333 2.333]
18	149	97.242	−51.758	false	[9 9 9 9]	[7.667 10.333 10.333 7.667 2.667]
19	154.5	97.891	−56.609	false	[9.5 9.5 9.5 9.5]	[8 11 11 8 3]
20	160	98.552	−61.448	false	[10 10 10 10]	[8.333 11.667 11.667 8.333 3.333]
21	165.5	99.226	−66.274	false	[10.5 10.5 10.5 10.5]	[8.667 12.333 12.333 8.667 3.667]
22	171	99.912	−71.088	false	[11 11 11 11]	[9 13 13 9 4]
23	176.5	100.611	−75.889	false	[11.5 11.5 11.5 11.5]	[9.333 13.667 13.667 9.333 4.333]
24	182	101.322	−80.679	false	[12 12 12 12]	[9.667 14.333 14.333 9.667 4.667]
25	187.5	102.046	−85.456	false	[12.5 12.5 12.5 12.5]	[10 15 15 10 5]
26	193	102.782	−90.222	false	[13 13 13 13]	[10.333 15.667 15.667 10.333 5.333]
27	198.5	103.531	−94.975	false	[13.5 13.5 13.5 13.5]	[10.667 16.333 16.333 10.667 5.667]

Continued on next page

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
28	204	104.292	−99.716	false	[14 14 14 14]	[11 17 17 11 6]
29	209.5	105.066	−104.444	false	[14.5 14.5 14.5 14.5]	[11.333 17.667 17.667 11.333 6.333]
30	215	105.852	−109.160	false	[15 15 15 15]	[11.667 18.333 18.333 11.667 6.667]
31	220.5	106.651	−113.863	false	[15.5 15.5 15.5 15.5]	[12 19 19 12 7]
32	226	107.462	−118.554	false	[16 16 16 16]	[12.333 19.667 19.667 12.333 7.333]
33	231.5	108.286	−123.232	false	[16.5 16.5 16.5 16.5]	[12.667 20.333 20.333 12.667 7.667]
34	237	109.122	−127.897	false	[17 17 17 17]	[13 21 21 13 8]
35	242.5	109.971	−132.548	false	[17.5 17.5 17.5 17.5]	[13.333 21.667 21.667 13.333 8.333]
36	248	110.833	−137.187	false	[18 18 18 18]	[13.667 22.333 22.333 13.667 8.667]
37	253.5	111.707	−141.812	false	[18.5 18.5 18.5 18.5]	[14 23 23 14 9]
38	259	112.594	−146.425	false	[19 19 19 19]	[14.333 23.667 23.667 14.333 9.333]
39	264.5	113.494	−151.023	false	[19.5 19.5 19.5 19.5]	[14.667 24.333 24.333 14.667 9.667]
40	270	114.406	−155.608	false	[20 20 20 20]	[15 25 25 15 10]
41	275.5	115.331	−160.179	false	[20.5 20.5 20.5 20.5]	[15.333 25.667 25.667 15.333 10.333]
42	281	116.269	−164.737	false	[21 21 21 21]	[15.667 26.333 26.333 15.667 10.667]
43	286.5	117.219	−169.280	false	[21.5 21.5 21.5 21.5]	[16 27 27 16 11]
44	292	118.182	−173.809	false	[22 22 22 22]	[16.333 27.667 27.667 16.333 11.333]
45	297.5	119.158	−178.323	false	[22.5 22.5 22.5 22.5]	[16.667 28.333 28.333 16.667 11.667]
46	303	120.147	−182.823	false	[23 23 23 23]	[17 29 29 17 12]
47	308.5	121.148	−187.309	false	[23.5 23.5 23.5 23.5]	[17.333 29.667 29.667 17.333 12.333]
48	314	122.162	−191.779	false	[24 24 24 24]	[17.667 30.333 30.333 17.667 12.667]
49	319.5	123.188	−196.234	false	[24.5 24.5 24.5 24.5]	[18 31 31 18 13]
50	325	124.227	−200.440	false	[25 25 25 25]	[18.333 31.667 31.667 18.333 13.333]

Table A.7: Case 5: constant+linear+quadratic: scan results for $Q = 1, \ldots, 50$ (deterministic user equilibrium).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	20	20	0	false	[0 0 1 1]	[0 0 1 1 0]
2	60	60	0	false	[0 0 2 2]	[0 0 2 2 0]
3	120	120	0	false	[0 0 3 3]	[0 0 3 3 0]
4	200	200	0	false	[0 0 4 4]	[0 0 4 4 0]
5	199.981	199.981	0	false	[1.000 1.000 4.000 4.000]	[1.000 1.000 4.000 4.000 0]
6	199.994	199.994	0	false	[2.000 2.000 4.000 4.000]	[2.000 2.000 4.000 4.000 0]
7	199.987	199.987	0	false	[3.000 3.000 4.000 4.000]	[3.000 3.000 4.000 4.000 0]
8	199.988	199.988	0	false	[4.000 4.000 4.000 4.000]	[4.000 4.000 4.000 4.000 0]
9	200.007	200.007	0	false	[5.000 5.000 4.000 4.000]	[5.000 5.000 4.000 4.000 0]
10	200.004	200.004	0	false	[6.000 6.000 4.000 4.000]	[6.000 6.000 4.000 4.000 0]
11	200.003	200.003	0	false	[7.000 7.000 4.000 4.000]	[7.000 7.000 4.000 4.000 0]
12	200.001	200.001	0	false	[8.000 8.000 4.000 4.000]	[8.000 8.000 4.000 4.000 0]
13	200.001	200.001	0	false	[9.000 9.000 4.000 4.000]	[9.000 9.000 4.000 4.000 0]
14	200	200	0	false	[10.000 10.000 4.000 4.000]	[10.000 10.000 4.000 4.000 0]
15	200	200	0	false	[11.000 11.000 4.000 4.000]	[11.000 11.000 4.000 4.000 0]
16	200	200	0	false	[12.000 12.000 4.000 4.000]	[12.000 12.000 4.000 4.000 0]
17	200	200	0	false	[13.000 13.000 4.000 4.000]	[13.000 13.000 4.000 4.000 0]
18	200	200	0	false	[14.000 14.000 4.000 4.000]	[14.000 14.000 4.000 4.000 0]
19	200	200	0	false	[15.000 15.000 4.000 4.000]	[15.000 15.000 4.000 4.000 0]
20	200	200	0	false	[16.000 16.000 4.000 4.000]	[16.000 16.000 4.000 4.000 0]
21	200	200	0	false	[17.000 17.000 4.000 4.000]	[17.000 17.000 4.000 4.000 0]
22	200	200	0	false	[18.000 18.000 4.000 4.000]	[18.000 18.000 4.000 4.000 0]
23	200	200	0	false	[19.000 19.000 4.000 4.000]	[19.000 19.000 4.000 4.000 0]
24	200	200	0	false	[20.000 20.000 4.000 4.000]	[20.000 20.000 4.000 4.000 0]
25	200	200	0	false	[21.000 21.000 4.000 4.000]	[21.000 21.000 4.000 4.000 0]
26	200	200	0	false	[22.000 22.000 4.000 4.000]	[22.000 22.000 4.000 4.000 0]
27	200	200	0	false	[23.000 23.000 4.000 4.000]	[23.000 23.000 4.000 4.000 0]
28	200	200	0	false	[24.000 24.000 4.000 4.000]	[24.000 24.000 4.000 4.000 0]
29	200	200	0	false	[25.000 25.000 4.000 4.000]	[25.000 25.000 4.000 4.000 0]
30	200	200	0	false	[26.000 26.000 4.000 4.000]	[26.000 26.000 4.000 4.000 0]
31	200	200	0	false	[27.000 27.000 4.000 4.000]	[27.000 27.000 4.000 4.000 0]

Continued on next page

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
32	200	200	0	false	[28.000 28.000 4.000 4.000]	[28.000 28.000 4.000 4.000 0]
33	200	200	0	false	[29.000 29.000 4.000 4.000]	[29.000 29.000 4.000 4.000 0]
34	200	200	0	false	[30.000 30.000 4.000 4.000]	[30.000 30.000 4.000 4.000 0]
35	200	200	0	false	[31.000 31.000 4.000 4.000]	[31.000 31.000 4.000 4.000 0]
36	200	200	0	false	[32.000 32.000 4.000 4.000]	[32.000 32.000 4.000 4.000 0]
37	200	200	0	false	[33.000 33.000 4.000 4.000]	[33.000 33.000 4.000 4.000 0]
38	200	200	0	false	[34.000 34.000 4.000 4.000]	[34.000 34.000 4.000 4.000 0]
39	200	200	0	false	[35.000 35.000 4.000 4.000]	[35.000 35.000 4.000 4.000 0]
40	200	200	0	false	[36.000 36.000 4.000 4.000]	[36.000 36.000 4.000 4.000 0]
41	200	200	0	false	[37.000 37.000 4.000 4.000]	[37.000 37.000 4.000 4.000 0]
42	200	200	0	false	[38.000 38.000 4.000 4.000]	[38.000 38.000 4.000 4.000 0]
43	200	200	0	false	[39.000 39.000 4.000 4.000]	[39.000 39.000 4.000 4.000 0]
44	200	200	0	false	[40.000 40.000 4.000 4.000]	[40.000 40.000 4.000 4.000 0]
45	200	200	0	false	[41.000 41.000 4.000 4.000]	[41.000 41.000 4.000 4.000 0]
46	200	200	0	false	[42.000 42.000 4.000 4.000]	[42.000 42.000 4.000 4.000 0]
47	200	200	0	false	[43.000 43.000 4.000 4.000]	[43.000 43.000 4.000 4.000 0]
48	200	200	0	false	[44.000 44.000 4.000 4.000]	[44.000 44.000 4.000 4.000 0]
49	200	200	0	false	[45.000 45.000 4.000 4.000]	[45.000 45.000 4.000 4.000 0]
50	200	200	0	false	[46.000 46.000 4.000 4.000]	[46.000 46.000 4.000 4.000 0]

Table A.8: Case 6: constant-shortcut + quadratic: scan results for $Q = 1, \ldots, 50$ (deterministic user equilibrium).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	15	30	15	true	[0.5 0.5 0.5 0.5]	[0.763 0.763 0.237 0.237 0]
2	30	90	60	true	[1 1 1 1]	[1.526 1.526 0.474 0.474 0]
3	45	105.023	60.023	true	[1.5 1.5 1.5 1.5]	[2.292 2.292 0.708 0.708 0]
4	60	103.084	43.084	true	[2 2 2 2]	[3.057 3.057 0.943 0.943 0]
5	75	100.811	25.812	true	[2.5 2.5 2.5 2.5]	[3.822 3.822 1.178 1.178 0]
6	90	96.712	6.712	true	[3 3 3 3]	[4.587 4.587 1.413 1.413 0]
7	105	82.993	−22.007	false	[3.5 3.5 3.5 3.5]	[3.493 3.178 3.507 3.822 0.315]
8	120	62.631	−57.369	false	[4 4 4 4]	[3.945 3.221 4.055 4.779 0.723]
9	135	38.476	−96.524	false	[4.5 4.5 4.5 4.5]	[4.408 3.318 4.592 5.682 1.090]
10	150	10.307	−139.693	false	[5 5 5 5]	[4.898 3.470 5.102 6.530 1.428]
11	165	−22.025	−187.025	false	[5.5 5.5 5.5 5.5]	[5.411 3.663 5.589 7.337 1.748]
12	180	−58.602	−238.602	false	[6 6 6 6]	[5.943 3.890 6.057 8.110 2.054]
13	195	−114.35	−309.35	false	[6.5 6.5 6.5 6.5]	[6.490 4.145 6.510 8.855 2.344]
14	210	−178.291	−388.291	false	[7 7 7 7]	[7.047 4.420 6.953 9.580 2.627]
15	225	−249.478	−474.478	false	[7.5 7.5 7.5 7.5]	[7.613 4.717 7.387 10.283 2.896]
16	240	−328.898	−568.898	false	[8 8 8 8]	[8.186 5.025 7.814 10.975 3.162]
17	255	−416.552	−671.552	false	[8.5 8.5 8.5 8.5]	[8.762 5.349 8.238 11.651 3.413]
18	270	−512.44	−782.440	false	[9 9 9 9]	[9.343 5.686 8.657 12.314 3.657]
19	285	−616.57	−901.570	false	[9.5 9.5 9.5 9.5]	[9.927 6.034 9.073 12.966 3.893]
20	300	−728.94	−1028.941	false	[10 10 10 10]	[10.514 6.390 9.486 13.610 4.124]
21	315	−849.556	−1164.556	false	[10.5 10.5 10.5 10.5]	[11.104 6.753 9.896 14.247 4.350]
22	330	−978.42	−1308.421	false	[11 11 11 11]	[11.696 7.123 10.304 14.877 4.573]
23	345	−1115.53	−1460.530	false	[11.5 11.5 11.5 11.5]	[12.290 7.496 10.710 15.504 4.793]
24	360	−1260.89	−1620.887	false	[12 12 12 12]	[12.886 7.874 11.114 16.126 5.012]
25	375	−1414.5	−1789.498	false	[12.5 12.5 12.5 12.5]	[13.484 8.254 11.516 16.746 5.230]
26	390	−1576.37	−1966.372	false	[13 13 13 13]	[14.084 8.636 11.916 17.364 5.448]
27	405	−1746.5	−2151.503	false	[13.5 13.5 13.5 13.5]	[14.685 9.017 12.315 17.983 5.668]
28	420	−1924.89	−2344.891	false	[14 14 14 14]	[15.288 9.399 12.712 18.601 5.889]
29	435	−2111.55	−2546.548	false	[14.5 14.5 14.5 14.5]	[15.892 9.781 13.108 19.219 6.111]
30	450	−2306.48	−2756.481	false	[15 15 15 15]	[16.498 10.163 13.502 19.837 6.335]
31	465	−2509.69	−2974.692	false	[15.5 15.5 15.5 15.5]	[17.104 10.545 13.896 20.455 6.560]
32	480	−2721.18	−3201.181	false	[16 16 16 16]	[17.712 10.926 14.288 21.074 6.786]
33	495	−2940.96	−3435.961	false	[16.5 16.5 16.5 16.5]	[18.321 11.308 14.679 21.692 7.013]
34	510	−3169.03	−3679.033	false	[17 17 17 17]	[18.931 11.689 15.069 22.311 7.241]
35	525	−3405.4	−3930.399	false	[17.5 17.5 17.5 17.5]	[19.541 12.070 15.459 22.930 7.471]

Continued on next page

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
36	540	−3650.07	−4189.062	false	[18 18 18 18]	[20.153 12.451 15.847 23.549 7.702]
37	555	−3903.04	−4455.021	false	[18.5 18.5 18.5 18.5]	[20.765 12.832 16.235 24.168 7.934]
38	570	−4164.32	−4728.275	false	[19 19 19 19]	[21.378 13.211 16.622 24.789 8.167]
39	585	−4433.91	−5008.823	false	[19.5 19.5 19.5 19.5]	[21.992 13.591 17.008 25.409 8.401]
40	600	−4711.81	−5296.661	false	[20 20 20 20]	[22.606 13.969 17.394 26.031 8.637]
41	615	−4998.03	−5591.785	false	[20.5 20.5 20.5 20.5]	[23.221 14.347 17.779 26.653 8.874]
42	630	−5292.57	−5894.193	false	[21 21 21 21]	[23.836 14.724 18.164 27.276 9.112]
43	645	−5595.43	−6203.880	false	[21.5 21.5 21.5 21.5]	[24.451 15.100 18.549 27.900 9.351]
44	660	−5906.61	−6520.843	false	[22 22 22 22]	[25.067 15.475 18.933 28.525 9.592]
45	675	−6226.12	−6845.269	false	[22.5 22.5 22.5 22.5]	[25.683 15.849 19.317 29.151 9.834]
46	690	−6553.97	−7177.139	false	[23 23 23 23]	[26.300 16.222 19.700 29.778 10.077]
47	705	−6890.17	−7516.549	false	[23.5 23.5 23.5 23.5]	[26.916 16.594 20.084 30.406 10.322]
48	720	−7234.73	−7863.693	false	[24 24 24 24]	[27.533 16.965 20.467 31.035 10.568]
49	735	−7587.65	−8218.623	false	[24.5 24.5 24.5 24.5]	[28.150 17.334 20.850 31.666 10.816]
50	750	−7948.92	−8581.363	false	[25 25 25 25]	[28.767 17.702 21.233 32.298 11.065]

Table A.9: Case A (mixed specification): full scan results for $Q = 1, \ldots, 20$ under DUE. $ATT^{(B)}$ and $ATT^{(A)}$ denote the equilibrium average travel time before and after adding the bypass. LinkFlow is reported as $[x_1\ x_2\ x_3\ x_4\ x_5]$, with $x_5 = 0$ in the before-network by convention.

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	55.5	30.000	−25.500	false	[0.5 0.5 0.5 0.5]	[1 0 0 1 1]
2	61.0	50.000	−11.000	false	[1 1 1 1]	[2 0 0 2 2]
3	66.5	70.000	3.500	true	[1.5 1.5 1.5 1.5]	[3 0 0 3 3]
4	72.0	89.998	17.998	true	[2 2 2 2]	[4.000 0.000 0.000 4.000 4.000]
5	77.5	91.820	14.320	true	[2.5 2.5 2.5 2.5]	[4.091 0.909 0.909 4.091 3.182]
6	83.0	93.641	10.641	true	[3 3 3 3]	[4.182 1.818 1.818 4.182 2.364]
7	88.5	95.454	6.955	true	[3.5 3.5 3.5 3.5]	[4.273 2.727 2.727 4.273 1.545]
8	94.0	97.274	3.274	true	[4 4 4 4]	[4.364 3.636 3.636 4.364 0.727]
9	99.5	99.5	0	false	[4.5 4.5 4.5 4.5]	[4.5 4.5 4.5 4.5 0]
10	105.0	105.0	0	false	[5 5 5 5]	[5 5 5 5 0]
11	110.5	110.5	0	false	[5.5 5.5 5.5 5.5]	[5.5 5.5 5.5 5.5 0]
12	116.0	116.0	0	false	[6 6 6 6]	[6 6 6 6 0]
13	121.5	121.5	0	false	[6.5 6.5 6.5 6.5]	[6.5 6.5 6.5 6.5 0]
14	127.0	127.0	0	false	[7 7 7 7]	[7 7 7 7 0]
15	132.5	132.5	0	false	[7.5 7.5 7.5 7.5]	[7.5 7.5 7.5 7.5 0]
16	138.0	138.0	0	false	[8 8 8 8]	[8 8 8 8 0]
17	143.5	143.5	0	false	[8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 0]
18	149.0	149.0	0	false	[9 9 9 9]	[9 9 9 9 0]
19	154.5	154.5	0	false	[9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 0]
20	160.0	160.0	0	false	[10 10 10 10]	[10 10 10 10 0]

Table A.10: Case B (fully nonlinear specification): full scan results for $Q = 200, 400, \ldots, 6000$ under DUE. LinkFlow is reported as $[x_1\ x_2\ x_3\ x_4\ x_5]$, with $x_5 = 0$ throughout, indicating that the added bypass is not on the equilibrium support.

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
200	90.001	90.001	0	false	[200 200 0 0]	[200 200 0 0 0]
400	90.022	90.022	0	false	[400 400 0 0]	[400 400 0 0 0]
600	90.109	90.109	0	false	[600 600 0 0]	[600 600 0 0 0]
800	90.346	90.346	0	false	[800 800 0 0]	[800 800 0 0 0]
1000	90.844	90.844	0	false	[1000 1000 0 0]	[1000 1000 0 0 0]
1200	91.750	91.750	0	false	[1200 1200 0 0]	[1200 1200 0 0 0]
1400	93.241	93.241	0	false	[1400 1400 0 0]	[1400 1400 0 0 0]
1600	95.530	95.530	0	false	[1600 1600 0 0]	[1600 1600 0 0 0]
1800	98.857	98.857	0	false	[1800 1800 0 0]	[1800 1800 0 0 0]
2000	99.999	99.999	0	false	[1855.41 1855.41 144.591 144.591]	[1855.41 1855.41 144.591 144.591 0]
2200	100.01	100.01	0	false	[1855.83 1855.83 344.172 344.172]	[1855.83 1855.83 344.172 344.172 0]
2400	100.08	100.08	0	false	[1858.97 1858.97 541.03 541.03]	[1858.97 1858.97 541.03 541.03 0]
2600	100.27	100.27	0	false	[1867.62 1867.62 732.378 732.378]	[1867.62 1867.62 732.378 732.378 0]
2800	100.65	100.65	0	false	[1884.39 1884.39 915.607 915.607]	[1884.39 1884.39 915.607 915.607 0]
3000	101.31	101.31	0	false	[1913.30 1913.30 1086.70 1086.70]	[1913.30 1913.30 1086.70 1086.70 0]
3200	102.27	102.27	0	false	[1952.67 1952.67 1247.33 1247.33]	[1952.67 1952.67 1247.33 1247.33 0]
3400	103.57	103.57	0	false	[2002.77 2002.77 1397.23 1397.23]	[2002.77 2002.77 1397.23 1397.23 0]
3600	105.25	105.25	0	false	[2061.80 2061.80 1538.20 1538.20]	[2061.80 2061.80 1538.20 1538.20 0]
3800	107.32	107.32	0	false	[2128.54 2128.54 1671.46 1671.46]	[2128.54 2128.54 1671.46 1671.46 0]
4000	109.81	109.81	0	false	[2201.45 2201.45 1798.55 1798.55]	[2201.45 2201.45 1798.55 1798.55 0]
4200	112.76	112.76	0	false	[2279.01 2279.01 1920.99 1920.99]	[2279.01 2279.01 1920.99 1920.99 0]
4400	116.21	116.21	0	false	[2360.87 2360.87 2039.13 2039.13]	[2360.87 2360.87 2039.13 2039.13 0]
4600	120.19	120.19	0	false	[2445.82 2445.82 2154.18 2154.18]	[2445.82 2445.82 2154.18 2154.18 0]
4800	124.75	124.75	0	false	[2533.28 2533.28 2266.72 2266.72]	[2533.28 2533.28 2266.72 2266.72 0]
5000	129.93	129.93	0	false	[2622.98 2622.98 2377.02 2377.02]	[2622.98 2622.98 2377.02 2377.02 0]
5200	135.79	135.79	0	false	[2714.34 2714.34 2485.66 2485.66]	[2714.34 2714.34 2485.66 2485.66 0]
5400	142.38	142.38	0	false	[2807.01 2807.01 2592.99 2592.99]	[2807.01 2807.01 2592.99 2592.99 0]
5600	149.75	149.75	0	false	[2900.74 2900.74 2699.26 2699.26]	[2900.74 2900.74 2699.26 2699.26 0]
5800	157.97	157.97	0	false	[2996.08 2996.08 2803.92 2803.92]	[2996.08 2996.08 2803.92 2803.92 0]
6000	167.08	167.08	0	false	[3091.49 3091.49 2908.51 2908.51]	[3091.49 3091.49 2908.51 2908.51 0]

A.3. Supplementary table for Section 5.3

Table A.11: SUE scan results ($\theta = 0.2$): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	55.5	31.000	-24.5	false	[0.5 0.5 0.5 0.5]	[1 0 0 1 1]
2	61.0	52.000	-9.0	false	[1 1 1 1]	[2 0 0 2 2]
3	66.5	73.000	6.5	true	[1.5 1.5 1.5 1.5]	[3 0 0 3 3]
4	72.0	86.540	14.54	true	[2 2 2 2]	[3.657 0.343 0.343 3.657 3.315]
5	77.5	89.413	11.913	true	[2.5 2.5 2.5 2.5]	[3.833 1.167 1.167 3.833 2.667]
6	83.0	92.000	9.0	true	[3 3 3 3]	[4 2 2 4 2]
7	88.5	94.518	6.018	true	[3.5 3.5 3.5 3.5]	[4.165 2.835 2.835 4.165 1.331]
8	94.0	97.033	3.0334	true	[4 4 4 4]	[4.334 3.666 3.666 4.334 0.668]
9	99.5	99.763	0.26293	true	[4.5 4.5 4.5 4.5]	[4.529 4.471 4.471 4.529 0.057]
10	105.0	105.000	0	false	[5 5 5 5]	[5 5 5 5 0]
11	110.5	110.500	0	false	[5.5 5.5 5.5 5.5]	[5.5 5.5 5.5 5.5 0]
12	116.0	116.000	0	false	[6 6 6 6]	[6 6 6 6 0]
13	121.5	121.500	0	false	[6.5 6.5 6.5 6.5]	[6.5 6.5 6.5 6.5 0]
14	127.0	127.000	0	false	[7 7 7 7]	[7 7 7 7 0]
15	132.5	132.500	0	false	[7.5 7.5 7.5 7.5]	[7.5 7.5 7.5 7.5 0]
16	138.0	138.000	0	false	[8 8 8 8]	[8 8 8 8 0]
17	143.5	143.500	0	false	[8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 0]
18	149.0	149.000	0	false	[9 9 9 9]	[9 9 9 9 0]
19	154.5	154.500	0	false	[9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 0]
20	160.0	160.000	0	false	[10 10 10 10]	[10 10 10 10 0]

Table A.12: SUE scan results ($\theta = 0.8$): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	55.5	31.000	-24.5	false	[0.5 0.5 0.5 0.5]	[1 0 0 1 1]
2	61.0	52.000	-9.0	false	[1 1 1 1]	[2 0 0 2 2]
3	66.5	72.992	6.4918	true	[1.5 1.5 1.5 1.5]	[3.000 0.000 0.000 3.000 2.999]
4	72.0	84.881	12.881	true	[2 2 2 2]	[3.570 0.430 0.430 3.570 3.140]
5	77.5	88.868	11.368	true	[2.5 2.5 2.5 2.5]	[3.799 1.201 1.201 3.799 2.597]
6	83.0	92.000	9.0	true	[3 3 3 3]	[4 2 2 4 2]
7	88.5	94.883	6.3829	true	[3.5 3.5 3.5 3.5]	[4.197 2.803 2.803 4.197 1.394]
8	94.0	97.722	3.7223	true	[4 4 4 4]	[4.400 3.600 3.600 4.400 0.800]
9	99.5	100.77	1.2679	true	[4.5 4.5 4.5 4.5]	[4.634 4.367 4.367 4.634 0.267]
10	105.0	105.04	0.044887	true	[5 5 5 5]	[5.004 4.996 4.996 5.004 0.009]
11	110.5	110.5	0.00020533	false	[5.5 5.5 5.5 5.5]	[5.500 5.500 5.500 5.500 0]
12	116.0	116.0	0	false	[6 6 6 6 0]	[6 6 6 6 0]
13	121.5	121.5	0	false	[6.5 6.5 6.5 6.5]	[6.5 6.5 6.5 6.5 0]
14	127.0	127.0	0	false	[7 7 7 7 0]	[7 7 7 7 0]
15	132.5	132.5	0	false	[7.5 7.5 7.5 7.5]	[7.5 7.5 7.5 7.5 0]
16	138.0	138.0	0	false	[8 8 8 8]	[8 8 8 8 0]
17	143.5	143.5	0	false	[8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 0]
18	149.0	149.0	0	false	[9 9 9 9]	[9 9 9 9 0]
19	154.5	154.5	0	false	[9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 0]
20	160.0	160.0	0	false	[10 10 10 10]	[10 10 10 10 0]

Table A.13: SUE scan results ($\theta = 5$): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	55.5	31.108	-24.392	false	[0.5 0.5 0.5 0.5 0]	[0.997 0.003 0.003 0.997 0.994]
2	61.0	51.843	-9.1571	false	[1 1 1 1 0]	[1.954 0.046 0.046 1.954 1.908]
3	66.5	68.760	2.2601	true	[1.5 1.5 1.5 1.5 0]	[2.715 0.285 0.285 2.715 2.429]
4	72.0	79.494	7.4944	true	[2 2 2 2 0]	[3.239 0.761 0.761 3.239 2.477]
5	77.5	86.602	9.1016	true	[2.5 2.5 2.5 2.5 0]	[3.645 1.355 1.355 3.645 2.289]
6	83.0	92.000	9.0	true	[3 3 3 3 0]	[4 2 2 4 2]
7	88.5	96.566	8.0658	true	[3.5 3.5 3.5 3.5 0]	[4.334 2.666 2.666 4.334 1.668]
8	94.0	100.73	6.7301	true	[4 4 4 4 0]	[4.663 3.337 3.337 4.663 1.326]
9	99.5	104.74	5.2439	true	[4.5 4.5 4.5 4.5 0]	[4.997 4.003 4.003 4.997 0.994]
10	105.0	108.78	3.7843	true	[5 5 5 5 0]	[5.347 4.653 4.653 5.347 0.694]
11	110.5	113.00	2.4952	true	[5.5 5.5 5.5 5.5 0]	[5.722 5.278 5.278 5.722 0.444]
12	116.0	117.49	1.4861	true	[6 6 6 6 0]	[6.128 5.872 5.872 6.128 0.256]
13	121.5	122.30	0.80073	true	[6.5 6.5 6.5 6.5 0]	[6.567 6.433 6.433 6.567 0.134]
14	127.0	127.40	0.39761	true	[7 7 7 7 0]	[7.032 6.968 6.968 7.032 0.064]
15	132.5	132.69	0.18685	true	[7.5 7.5 7.5 7.5 0]	[7.515 7.485 7.485 7.515 0.029]
16	138.0	138.08	0.084949	true	[8 8 8 8 0]	[8.007 7.993 7.993 8.007 0.013]
17	143.5	143.54	0.037887	true	[8.5 8.5 8.5 8.5 0]	[8.503 8.497 8.497 8.503 0.006]
18	149.0	149.02	0.016701	true	[9 9 9 9 0]	[9.001 8.999 8.999 9.001 0.002]
19	154.5	154.51	0.0073041	false	[9.5 9.5 9.5 9.5 0]	[9.501 9.499 9.499 9.501 0.001]
20	160.0	160.00	0.0031761	false	[10 10 10 10 0]	[10.000 10.000 10.000 10.000 0.000]

A.4. Supplementary table for Section 5.4

Table A.14: DUE scan results: classical diamond network with bidirectional bypass ($b \leftrightarrow c$): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	55.500	31.000	-24.500	false	[0.5 0.5 0.5 0.5]	[1 0 0 1 1 0]
2	61.000	52.000	-9.000	false	[1 1 1 1]	[2 0 0 2 2 0]
3	66.500	73.000	6.500	true	[1.5 1.5 1.5 1.5]	[3 0 0 3 3 0]
4	72.000	87.197	15.197	true	[2 2 2 2]	[3.691 0.309 0.309 3.691 3.381 0]
5	77.500	89.614	12.114	true	[2.5 2.5 2.5 2.5]	[3.846 1.154 1.154 3.846 2.692 0]
6	83.000	92.001	9.001	true	[3 3 3 3]	[4 2 2 4 2.000 0.000]
7	88.500	94.390	5.890	true	[3.5 3.5 3.5 3.5]	[4.154 2.846 2.846 4.154 1.309 0.001]
8	94.000	96.772	2.772	true	[4 4 4 4]	[4.308 3.692 3.692 4.308 0.616 0.001]
9	99.500	99.500	0.000	false	[4.5 4.5 4.5 4.5]	[4.5 4.5 4.5 4.5 0 0]
10	105.000	105.000	0.000	false	[5 5 5 5]	[5 5 5 5 0 0]
11	110.500	110.500	0.000	false	[5.5 5.5 5.5 5.5]	[5.5 5.5 5.5 5.5 0 0]
12	116.000	116.000	0.000	false	[6 6 6 6]	[6 6 6 6 0 0]
13	121.500	121.500	0.000	false	[6.5 6.5 6.5 6.5]	[6.5 6.5 6.5 6.5 0 0]
14	127.000	124.923	-2.077	false	[7 7 7 7]	[6.769 7.231 7.231 6.769 0.001 0.462]
15	132.500	127.307	-5.193	false	[7.5 7.5 7.5 7.5]	[6.923 8.077 8.077 6.923 0.001 1.156]
16	138.000	129.697	-8.303	false	[8 8 8 8]	[7.077 8.923 8.923 7.077 0.002 1.848]
17	143.500	132.079	-11.421	false	[8.5 8.5 8.5 8.5]	[7.229 9.771 9.771 7.229 0.008 2.549]
18	149.000	134.461	-14.539	false	[9 9 9 9]	[7.382 10.618 10.618 7.382 0.010 3.245]
19	154.500	136.852	-17.648	false	[9.5 9.5 9.5 9.5]	[7.539 11.462 11.462 7.539 0.004 3.927]
20	160.000	139.237	-20.763	false	[10 10 10 10]	[7.691 12.309 12.309 7.691 0.009 4.627]

Table A.15: DUE scan results: ladder $n = 0$, configuration Cfg 1 ($k = 1$, pos=1): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	10.500	4.000	-6.500	false	[0.5 0.5 0.5 0.5]	[1 0 0 1 1 0]
2	11.000	6.000	-5.000	false	[1 1 1 1]	[2 0 0 2 2 0]
3	11.500	8.000	-3.500	false	[1.5 1.5 1.5 1.5]	[3 0 0 3 3 0]
4	12.000	10.000	-2.000	false	[2 2 2 2]	[4 0 0 4 4 0]
5	12.500	12.000	-0.500	false	[2.5 2.5 2.5 2.5]	[5 0 0 5 5 0]
6	13.000	14.000	1.000	true	[3 3 3 3]	[6 0 0 6 6 0]
7	13.500	16.000	2.500	true	[3.5 3.5 3.5 3.5]	[7 0 0 7 7 0]
8	14.000	18.000	4.000	true	[4 4 4 4]	[8.000 0.000 0.000 8.000 8.000 0]
9	14.500	18.000	3.500	true	[4.5 4.5 4.5 4.5]	[8 1 1 8 7 0]
10	15.000	18.000	3.000	true	[5 5 5 5]	[8.000 2.000 2.000 8.000 6.000 0]
11	15.500	18.000	2.500	true	[5.5 5.5 5.5 5.5]	[8.000 3.000 3.000 8.000 5.000 0]
12	16.000	18.000	2.000	true	[6 6 6 6]	[8 4 4 8 4 0]
13	16.500	18.001	1.501	true	[6.5 6.5 6.5 6.5]	[7.999 5.001 5.001 7.999 3.005 0.006]
14	17.000	18.005	1.005	true	[7 7 7 7]	[8.002 5.998 5.998 8.002 2.011 0.006]
15	17.500	18.000	0.500	true	[7.5 7.5 7.5 7.5]	[8 7 7 8 1.001 0.001]
16	18.000	18.000	0.000	false	[8 8 8 8]	[8.000 8.000 8.000 8.000 0.001 0]
17	18.500	18.500	0.000	false	[8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 0 0]
18	19.000	19.000	0.000	false	[9 9 9 9]	[9 9 9 9 0 0]
19	19.500	19.500	0.000	false	[9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 0 0]
20	20.000	20.000	0.000	false	[10 10 10 10]	[10 10 10 10 0 0]

Table A.16: DUE scan results: ladder $n = 2$, configuration Cfg 1 ($k = 1$, pos=1): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	21.000	15.000	-6.000	false	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	[1 0 0 0 1 1 0 1 1 0]
2	22.000	18.000	-4.000	false	[1 1 1 1 1 1 1 1]	[2 0 0 0 2 2 0 2 2 0]
3	23.000	21.000	-2.000	false	[1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5]	[3 0 0 0 3 3 0 3 3 0]
4	24.000	24.000	-0.000	false	[2 2 2 2 2 2 2 2]	[4 0 0.000 0.000 4.000 4.000 0.000 4.000 4.000 0]
5	25.000	25.666	0.666	true	[2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5]	[5 0 0.667 0.667 4.333 4.333 0.667 4.333 4.333 0]
6	26.000	27.333	1.333	true	[3 3 3 3 3 3 3 3]	[6 0 1.333 1.333 4.667 4.667 1.333 4.667 4.667 0]
7	27.000	29.000	2.000	true	[3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5]	[7 0 2.000 2.000 5.000 5.000 2.000 5.000 5.000 0]
8	28.000	30.666	2.666	true	[4 4 4 4 4 4 4 4]	[7.999 0.001 2.666 2.666 5.334 5.334 2.666 5.334 5.333 0]
9	29.000	31.333	2.333	true	[4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5]	[7.999 1.001 3.333 3.333 5.667 5.667 3.333 5.667 4.666 0]
10	30.000	32.000	2.000	true	[5 5 5 5 5 5 5 5]	[8.000 2.000 4.000 4.000 6.000 6.000 4.000 6.000 4.000 0]
11	31.000	32.666	1.666	true	[5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5]	[7.998 3.002 4.666 4.666 6.334 6.334 4.666 6.334 3.332 0]
12	32.000	33.333	1.333	true	[6 6 6 6 6 6 6 6]	[8.000 4.000 5.333 5.333 6.667 6.667 5.333 6.667 2.667 0]
13	33.000	34.000	1.000	true	[6.5 6.5 6.5 6.5 6.5 6.5 6.5 6.5]	[8.000 5.000 6.000 6.000 7.000 7.000 6.000 7.000 2.001 0.001]
14	34.000	34.667	0.667	true	[7 7 7 7 7 7 7 7]	[8.000 6.000 6.667 6.667 7.333 7.333 6.667 7.333 1.334 0.000]
15	35.000	35.333	0.333	true	[7.5 7.5 7.5 7.5 7.5 7.5 7.5 7.5]	[8.000 7.000 7.333 7.333 7.667 7.667 7.333 7.667 0.667 0]
16	36.000	36.000	0.000	false	[8 8 8 8 8 8 8 8]	[8.000 8.000 8.000 8.000 8.000 8.000 8.000 8.000 0.000 0]
17	37.000	37.000	0.000	false	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 0 0]
18	38.000	38.000	0.000	false	[9 9 9 9 9 9 9 9]	[9 9 9 9 9 9 9 0 0]
19	39.000	39.000	0.000	false	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 0 0]
20	40.000	40.000	0.000	false	[10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 0 0]

Table A.17: DUE scan results: ladder $n = 2$, configuration Cfg 2 ($k = 1$, pos=2): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	21.000	6.000	-15.000	false	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	[1 0 1 0 0 1 0 1 1 0]
2	22.000	10.000	-12.000	false	[1 1 1 1 1 1 1 1]	[2 0 2 0 0 2 0 2 2 0]
3	23.000	14.000	-9.000	false	[1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5]	[3 0 3 0 0 3 0 3 3 0]
4	24.000	18.000	-6.000	false	[2 2 2 2 2 2 2 2]	[4 0 4 0 0 4 0 4 4 0]
5	25.000	22.000	-3.000	false	[2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5]	[5 0 5 0 0 5 0 5 5 0]
6	26.000	26.000	0.000	false	[3 3 3 3 3 3 3 3]	[6 0 6 0 0 6 0 6 6 0]
7	27.000	30.000	3.000	true	[3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5]	[7 0 7 0 0 7 0 7 7 0]
8	28.000	34.000	6.000	true	[4 4 4 4 4 4 4 4]	[8 0 8 0 0 8 0 8 8 0]
9	29.000	37.999	8.999	true	[4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5]	[9.000 0.000 9.000 0.000 0.000 9.000 0.000 9.000 9.000 0]
10	30.000	38.000	8.000	true	[5 5 5 5 5 5 5 5]	[9.000 1.000 9.000 1.000 1.000 9.000 1.000 9.000 8.000 0]
11	31.000	38.001	7.001	true	[5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5]	[9.000 2.000 9.000 2.000 2.000 9.000 2.000 9.000 7.001 0.000]
12	32.000	38.002	6.002	true	[6 6 6 6 6 6 6 6]	[9.000 3.000 9.000 3.000 3.000 9.000 3.000 9.000 6.003 0.002]
13	33.000	38.000	5.000	true	[6.5 6.5 6.5 6.5 6.5 6.5 6.5 6.5]	[9.000 4.000 9.000 4.000 4.000 9.000 4.000 9.000 5.001 0.001]
14	34.000	38.000	4.000	true	[7 7 7 7 7 7 7 7]	[9.000 5.000 9.000 5.000 5.000 9.000 5.000 9.000 4.001 0.001]
15	35.000	38.001	3.001	true	[7.5 7.5 7.5 7.5 7.5 7.5 7.5 7.5]	[9.000 6.000 9.000 6.000 6.000 9.000 6.000 9.000 3.003 0.002]
16	36.000	38.004	2.004	true	[8 8 8 8 8 8 8 8]	[9.001 6.999 9.001 6.999 6.999 9.001 6.999 9.001 2.007 0.004]
17	37.000	38.002	1.002	true	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5]	[9.001 7.999 9.001 7.999 7.999 9.001 7.999 9.001 1.004 0.003]
18	38.000	38.001	0.001	false	[9 9 9 9 9 9 9 9]	[9.000 9.000 9.000 9.000 9.000 9.000 9.000 9.000 0.001 0.001]
19	39.000	39.000	0.000	false	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 0 0]
20	40.000	40.000	0.000	false	[10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 0 0]

Table A.18: DUE scan results: ladder $n = 2$, configuration Cfg 3 ($k = 1$, pos=3): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	21.000	15.000	-6.000	false	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	[1 0 1 1 0 0 0 1 1 0]
2	22.000	18.000	-4.000	false	[1 1 1 1 1 1 1 1]	[2 0 2 2 0 0 0 2 2 0]
3	23.000	21.000	-2.000	false	[1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5]	[3 0 3 3 0 0 0 3 3 0]
4	24.000	24.000	-0.000	false	[2 2 2 2 2 2 2 2]	[4.000 0.000 4.000 4.000 0.000 0.000 0 4 4.000 0]
5	25.000	25.666	0.666	true	[2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5]	[4.333 0.667 4.333 4.333 0.667 0.667 0 5 4.333 0]
6	26.000	27.333	1.333	true	[3 3 3 3 3 3 3 3]	[4.667 1.333 4.667 4.667 1.333 1.333 0 6 4.667 0]
7	27.000	29.000	2.000	true	[3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5]	[5.000 2.000 5.000 5.000 2.000 2.000 0 7 5.000 0]
8	28.000	30.666	2.666	true	[4 4 4 4 4 4 4 4]	[5.334 2.666 5.334 5.334 2.666 2.666 0.001 7.999 5.333 0]
9	29.000	31.333	2.333	true	[4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5]	[5.667 3.333 5.667 5.667 3.333 3.333 1.001 7.999 4.666 0]
10	30.000	32.000	2.000	true	[5 5 5 5 5 5 5 5]	[6.000 4.000 6.000 6.000 4.000 4.000 2.000 8.000 4.000 0]
11	31.000	32.666	1.666	true	[5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5]	[6.334 4.666 6.334 6.334 4.666 4.666 3.002 7.998 3.332 0]
12	32.000	33.333	1.333	true	[6 6 6 6 6 6 6 6]	[6.667 5.333 6.667 6.667 5.333 5.333 4.000 8.000 2.667 0]
13	33.000	34.000	1.000	true	[6.5 6.5 6.5 6.5 6.5 6.5 6.5 6.5]	[7.000 6.000 7.000 7.000 6.000 6.000 5.000 8.000 2.001 0.001]
14	34.000	34.667	0.667	true	[7 7 7 7 7 7 7 7]	[7.333 6.667 7.333 7.333 6.667 6.667 6.000 8.000 1.334 0.000]
15	35.000	35.333	0.333	true	[7.5 7.5 7.5 7.5 7.5 7.5 7.5 7.5]	[7.667 7.333 7.667 7.667 7.333 7.333 7.000 8.000 0.667 0]
16	36.000	36.000	0.000	false	[8 8 8 8 8 8 8 8]	[8.000 8.000 8.000 8.000 8.000 8.000 8.000 8.000 0.000 0]
17	37.000	37.000	0.000	false	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 0 0]
18	38.000	38.000	0.000	false	[9 9 9 9 9 9 9 9]	[9 9 9 9 9 9 9 0 0]
19	39.000	39.000	0.000	false	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 0 0]
20	40.000	40.000	0.000	false	[10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 0 0]

Table A.19: DUE scan results: ladder $n = 2$, configuration Cfg 4 ($k = 2$, pos=`[1 2]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	21.000	6.000	-15.000	false	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	[1 0 1 0 0 1 0 1 0 0 1 0]
2	22.000	10.000	-12.000	false	[1 1 1 1 1 1 1 1]	[2 0 2 0 0 2 0 2 0 0 2 0]
3	23.000	14.000	-9.000	false	[1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5]	[3 0 3 0 0 3 0 3 0 0 3 0]
4	24.000	18.000	-6.000	false	[2 2 2 2 2 2 2 2]	[4 0 4 0 0 4 0 4 0 0 4 0]
5	25.000	22.000	-3.000	false	[2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5]	[5 0 5 0 0 5 0 5 0 0 5 0]
6	26.000	26.000	0.000	false	[3 3 3 3 3 3 3 3]	[6 0 6 0 0 6 0 6 0 0 6 0]
7	27.000	30.000	3.000	true	[3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5]	[7 0 7 0 0 7 0 7 0 0 7 0]
8	28.000	34.000	6.000	true	[4 4 4 4 4 4 4 4]	[8 0 8 0 0 8 0 8 0 0 8 0]
9	29.000	37.999	8.999	true	[4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5]	[9.000 0.000 9.000 0.000 0.000 9.000 0.000 9.000 0 0 9.000 0]
10	30.000	38.000	8.000	true	[5 5 5 5 5 5 5 5]	[9.000 1.000 9.000 1.000 1.000 9.000 1.000 9.000 0 0 8.000 0]
11	31.000	38.001	7.001	true	[5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5]	[9.000 2.000 9.000 2.000 2.000 9.000 2.000 9.000 0 0 7.001 0.000]
12	32.000	38.002	6.002	true	[6 6 6 6 6 6 6 6]	[9.000 3.000 9.000 3.000 3.000 9.000 3.000 9.000 0 0 6.003 0.002]
13	33.000	38.000	5.000	true	[6.5 6.5 6.5 6.5 6.5 6.5 6.5 6.5]	[9.000 4.000 9.000 4.000 4.000 9.000 4.000 9.000 0 0 5.001 0.001]
14	34.000	38.000	4.000	true	[7 7 7 7 7 7 7 7]	[9.000 5.000 9.000 5.000 5.000 9.000 5.000 9.000 0 0 4.001 0.001]
15	35.000	38.001	3.001	true	[7.5 7.5 7.5 7.5 7.5 7.5 7.5 7.5]	[9.000 6.000 9.000 6.000 6.000 9.000 6.000 9.000 0 0 3.003 0.002]
16	36.000	38.004	2.004	true	[8 8 8 8 8 8 8 8]	[9.001 6.999 9.001 6.999 6.999 9.001 6.999 9.001 0 0 2.007 0.004]
17	37.000	38.002	1.002	true	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5]	[9.001 7.999 9.001 7.999 7.999 9.001 7.999 9.001 0 0 1.004 0.003]
18	38.000	38.001	0.001	false	[9 9 9 9 9 9 9 9]	[9.000 9.000 9.000 9.000 9.000 9.000 9.000 9.000 0 0 0.001 0.001]
19	39.000	39.000	0.000	false	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 0 0 0 0]
20	40.000	40.000	0.000	false	[10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 0 0 0 0]

Table A.20: DUE scan results: ladder $n = 2$, configuration Cfg 5 ($k = 2$, pos=`[1 3]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	21.000	14.500	-6.500	false	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	[1 0 0.5 0.5 0.5 0.5 0 1 0.5 0 0.5 0]
2	22.000	17.000	-5.000	false	[1 1 1 1 1 1 1 1]	[2 0 1 1 1 1 0 2 1 0 1 0]
3	23.000	19.500	-3.500	false	[1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5]	[3 0 1.5 1.5 1.5 1.5 0 3 1.5 0 1.5 0]
4	24.000	22.000	-2.000	false	[2 2 2 2 2 2 2 2]	[4 0 2 2 2 2 0 4 2 0 2 0]
5	25.000	24.500	-0.500	false	[2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5]	[5 0 2.5 2.5 2.5 2.5 0 5 2.5 0 2.5 0]
6	26.000	27.000	1.000	true	[3 3 3 3 3 3 3 3]	[6 0 3 3 3 3 0 6 3 0 3 0]
7	27.000	29.500	2.500	true	[3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5]	[7 0 3.5 3.5 3.5 3.5 0 7 3.5 0 3.5 0]
8	28.000	32.000	4.000	true	[4 4 4 4 4 4 4 4]	[8.000 0.000 4 4 4 4 0.000 8.000 4.000 0 4.000 0]
9	29.000	32.499	3.499	true	[4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5]	[8.000 1.001 4.5 4.5 4.5 4.5 1.001 8.000 3.500 0 3.500 0]
10	30.000	33.001	3.001	true	[5 5 5 5 5 5 5 5]	[8.000 2.000 5 5 5 5 2.000 8.000 3.000 0 3.000 0]
11	31.000	33.500	2.500	true	[5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5]	[8.000 3.000 5.5 5.5 5.5 5.5 3.000 8.000 2.500 0 2.500 0]
12	32.000	34.000	2.000	true	[6 6 6 6 6 6 6 6]	[8.000 4.000 6 6 6 6 4.000 8.000 2.000 0.000 2.000 0.000]
13	33.000	34.501	1.501	true	[6.5 6.5 6.5 6.5 6.5 6.5 6.5 6.5]	[7.999 5.001 6.5 6.5 6.5 6.5 5.001 7.999 1.502 0.003 1.502 0.003]
14	34.000	35.001	1.001	true	[7 7 7 7 7 7 7 7]	[8.000 6.000 7 7 7 7 6.000 8.000 1.002 0.002 1.002 0.002]
15	35.000	35.500	0.500	true	[7.5 7.5 7.5 7.5 7.5 7.5 7.5 7.5]	[8.000 7.000 7.5 7.5 7.5 7.5 7.000 8.000 0.501 0.000 0.501 0.000]
16	36.000	36.000	0.000	false	[8 8 8 8 8 8 8 8]	[8.000 8.000 8 8 8 8 8.000 8.000 0.001 0.000 0.001 0.000]
17	37.000	37.000	0.000	false	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5]	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 0 0 0 0]
18	38.000	38.000	0.000	false	[9 9 9 9 9 9 9 9]	[9 9 9 9 9 9 9 0 0 0 0]
19	39.000	39.000	0.000	false	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 0 0 0 0]
20	40.000	40.000	0.000	false	[10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 0 0 0 0]

Table A.21: DUE scan results: ladder $n = 2$, configuration Cfg 6 ($k = 2$, pos=`[2 3]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	21.000	6.000	-15.000	false	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	[1 0 1 0 0 1 0 1 1 0 0 0]
2	22.000	10.000	-12.000	false	[1 1 1 1 1 1 1 1]	[2 0 2 0 0 2 0 2 2 0 0 0]
3	23.000	14.000	-9.000	false	[1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5]	[3 0 3 0 0 3 0 3 3 0 0 0]
4	24.000	18.000	-6.000	false	[2 2 2 2 2 2 2 2]	[4 0 4 0 0 4 0 4 4 0 0 0]
5	25.000	22.000	-3.000	false	[2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5]	[5 0 5 0 0 5 0 5 5 0 0 0]
6	26.000	26.000	0.000	false	[3 3 3 3 3 3 3 3]	[6 0 6 0 0 6 0 6 6 0 0 0]
7	27.000	30.000	3.000	true	[3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5]	[7 0 7 0 0 7 0 7 7 0 0 0]
8	28.000	34.000	6.000	true	[4 4 4 4 4 4 4 4]	[8 0 8 0 0 8 0 8 8 0 0 0]
9	29.000	37.999	8.999	true	[4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5]	[9.000 0.000 9.000 0.000 0.000 9.000 0.000 9.000 9.000 0 0 0]
10	30.000	38.000	8.000	true	[5 5 5 5 5 5 5 5]	[9.000 1.000 9.000 1.000 1.000 9.000 1.000 9.000 8.000 0 0 0]
11	31.000	38.001	7.001	true	[5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5]	[9.000 2.000 9.000 2.000 2.000 9.000 2.000 9.000 7.001 0.000 0 0]
12	32.000	38.002	6.002	true	[6 6 6 6 6 6 6 6]	[9.000 3.000 9.000 3.000 3.000 9.000 3.000 9.000 6.003 0.002 0 0]
13	33.000	38.000	5.000	true	[6.5 6.5 6.5 6.5 6.5 6.5 6.5 6.5]	[9.000 4.000 9.000 4.000 4.000 9.000 4.000 9.000 5.001 0.001 0 0]
14	34.000	38.000	4.000	true	[7 7 7 7 7 7 7 7]	[9.000 5.000 9.000 5.000 5.000 9.000 5.000 9.000 4.001 0.001 0 0]
15	35.000	38.001	3.001	true	[7.5 7.5 7.5 7.5 7.5 7.5 7.5 7.5]	[9.000 6.000 9.000 6.000 6.000 9.000 6.000 9.000 3.003 0.002 0 0]
16	36.000	38.004	2.004	true	[8 8 8 8 8 8 8 8]	[9.001 6.999 9.001 6.999 6.999 9.001 6.999 9.001 2.007 0.004 0 0]
17	37.000	38.002	1.002	true	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5]	[9.001 7.999 9.001 7.999 7.999 9.001 7.999 9.001 1.004 0.003 0 0]
18	38.000	38.001	0.001	false	[9 9 9 9 9 9 9 9]	[9.000 9.000 9.000 9.000 9.000 9.000 9.000 9.000 0.001 0.001 0 0]
19	39.000	39.000	0.000	false	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 0 0 0 0]
20	40.000	40.000	0.000	false	[10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 0 0 0 0]

Table A.22: DUE scan results: ladder $n = 2$, configuration Cfg 7 ($k = 3$, pos=`[1 2 3]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	21.000	6.000	-15.000	false	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	[1 0 1 0 0 1 0 1 0 0 1 0 0 0]
2	22.000	10.000	-12.000	false	[1 1 1 1 1 1 1 1]	[2 0 2 0 0 2 0 2 0 0 2 0 0 0]
3	23.000	14.000	-9.000	false	[1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5]	[3 0 3 0 0 3 0 3 0 0 3 0 0 0]
4	24.000	18.000	-6.000	false	[2 2 2 2 2 2 2 2]	[4 0 4 0 0 4 0 4 0 0 4 0 0 0]
5	25.000	22.000	-3.000	false	[2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5]	[5 0 5 0 0 5 0 5 0 0 5 0 0 0]
6	26.000	26.000	0.000	false	[3 3 3 3 3 3 3 3]	[6 0 6 0 0 6 0 6 0 0 6 0 0 0]
7	27.000	30.000	3.000	true	[3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5]	[7 0 7 0 0 7 0 7 0 0 7 0 0 0]
8	28.000	34.000	6.000	true	[4 4 4 4 4 4 4 4]	[8 0 8 0 0 8 0 8 0 0 8 0 0 0]
9	29.000	37.999	8.999	true	[4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.5]	[9.000 0.000 9.000 0.000 0.000 9.000 0.000 9.000 0 0 9.000 0 0 0]
10	30.000	38.000	8.000	true	[5 5 5 5 5 5 5 5]	[9.000 1.000 9.000 1.000 1.000 9.000 1.000 9.000 0 0 8.000 0 0 0]
11	31.000	38.001	7.001	true	[5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5]	[9.000 2.000 9.000 2.000 2.000 9.000 2.000 9.000 0 0 7.001 0.000 0 0]
12	32.000	38.002	6.002	true	[6 6 6 6 6 6 6 6]	[9.000 3.000 9.000 3.000 3.000 9.000 3.000 9.000 0 0 6.003 0.002 0 0]
13	33.000	38.000	5.000	true	[6.5 6.5 6.5 6.5 6.5 6.5 6.5 6.5]	[9.000 4.000 9.000 4.000 4.000 9.000 4.000 9.000 0 0 5.001 0.001 0 0]
14	34.000	38.000	4.000	true	[7 7 7 7 7 7 7 7]	[9.000 5.000 9.000 5.000 5.000 9.000 5.000 9.000 0 0 4.001 0.001 0 0]
15	35.000	38.001	3.001	true	[7.5 7.5 7.5 7.5 7.5 7.5 7.5 7.5]	[9.000 6.000 9.000 6.000 6.000 9.000 6.000 9.000 0 0 3.003 0.002 0 0]
16	36.000	38.004	2.004	true	[8 8 8 8 8 8 8 8]	[9.001 6.999 9.001 6.999 6.999 9.001 6.999 9.001 0 0 2.007 0.004 0 0]
17	37.000	38.002	1.002	true	[8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5]	[9.001 7.999 9.001 7.999 7.999 9.001 7.999 9.001 0 0 1.004 0.003 0 0]
18	38.000	38.001	0.001	false	[9 9 9 9 9 9 9 9]	[9.000 9.000 9.000 9.000 9.000 9.000 9.000 9.000 0 0 0.001 0.001 0 0]
19	39.000	39.000	0.000	false	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5]	[9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 0 0 0 0 0 0]
20	40.000	40.000	0.000	false	[10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 0 0 0 0 0 0]

Table A.23: DUE scan results: ladder $n = 3$, configuration Cfg 1 ($k = 1$, pos=1): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	23.000	-0.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1.000 1.000 1.000 0 0 0 1.000 0 0 0]
2	26.000	25.000	-1.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 1.5 1.5 1.5 0.5 0.5 0.5 1.5 0.5 0.5 0]
3	29.000	27.001	-1.999	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 2.001 2.001 2.001 0.999 0.999 0.999 2.001 0.999 0.999 0]
4	30.800	29.000	-1.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 2.500 2.500 2.500 1.500 1.500 1.500 2.500 1.500 1.500 0]
5	31.999	31.000	-0.999	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 3.000 3.000 3.000 2.000 2.000 2.000 3.000 2.000 2.000 0]
6	33.199	33.001	-0.198	false	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 3.504 3.504 3.504 2.496 2.496 2.496 3.504 2.496 2.496 0]
7	34.400	35.000	0.600	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 4.000 4.000 4.000 3.000 3.000 3.000 4.000 3.000 3.000 0]
8	35.600	37.000	1.399	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[7.999 0.001 4.500 4.500 4.500 3.500 3.500 3.500 4.500 3.500 3.499 0]
9	36.800	38.000	1.200	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[8.000 1.000 5.000 5.000 5.000 4.000 4.000 4.000 5.000 4.000 3.000 0]
10	38.000	39.000	1.000	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[8.000 2.000 5.499 5.499 5.499 4.501 4.501 4.501 5.499 4.501 2.500 0]
11	39.200	40.000	0.800	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[8.000 3.000 6.000 6.000 6.000 5.000 5.000 5.000 6.000 5.000 2.000 0]
12	40.401	41.000	0.599	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[8.000 4.000 6.500 6.500 6.500 5.500 5.500 5.500 6.500 5.500 1.500 0]
13	41.600	42.000	0.400	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[8.000 5.000 7.000 7.000 7.000 6.000 6.000 6.000 7.000 6.000 1.000 0]
14	42.800	43.000	0.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[8.000 6.000 7.500 7.500 7.500 6.500 6.500 6.500 7.500 6.500 0.500 0]
15	44.000	44.000	0.000	false	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000 0.000 0]
16	45.200	45.200	0.000	false	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600 0 0]
17	46.400	46.400	0.000	false	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0]

Table A.24: DUE scan results: ladder $n = 3$, configuration Cfg 2 ($k = 1$, pos=2): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	16.000	-7.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 0 0 0 1 1 0 1 1 0]
2	26.000	20.000	-6.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 0 0 0 2 2 0 2 2 0]
3	29.000	24.000	-5.000	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 0 0 0 3 3 0 3 3 0]
4	30.800	28.000	-2.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 0.000 0.000 0 4.000 4.000 0.000 4.000 4.000 0]
5	31.999	30.666	-1.333	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 0.667 0.667 0 4.333 4.333 0.667 4.333 4.333 0]
6	33.199	33.333	0.134	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 6 1.333 1.333 0 4.667 4.667 1.333 4.667 4.667 0]
7	34.400	36.000	1.600	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 7 2.000 2.000 0 5.000 5.000 2.000 5.000 5.000 0]
8	35.600	38.666	3.066	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8 0 8 2.667 2.667 0 5.333 5.333 2.667 5.333 5.333 0]
9	36.800	41.332	4.532	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[8.999 0.001 8.999 3.333 3.333 0.001 5.667 5.667 3.333 5.667 5.666 0]
10	38.000	42.000	4.000	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[9.000 1.000 9.000 4.000 4.000 1.000 6.000 6.000 4.000 6.000 5.000 0]
11	39.200	42.667	3.466	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[9.000 2.000 9.000 4.667 4.667 2.000 6.333 6.333 4.667 6.333 4.333 0]
12	40.401	43.333	2.932	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[9.000 3.000 9.000 5.333 5.333 3.000 6.667 6.667 5.333 6.667 3.667 0]
13	41.600	44.000	2.400	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[9.000 4.000 9.000 6.000 6.000 4.000 7.000 7.000 6.000 7.000 3.001 0.001]
14	42.800	44.666	1.866	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[9.000 5.000 9.000 6.667 6.667 5.000 7.333 7.333 6.667 7.333 2.333 0]
15	44.000	45.334	1.334	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[9.000 6.000 9.000 7.333 7.333 6.000 7.667 7.667 7.333 7.667 1.667 0]
16	45.200	46.000	0.800	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[9.000 7.000 9.000 8.000 8.000 7.000 8.000 8.000 8.000 8.000 1.000 0]
17	46.400	46.667	0.266	true	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[9.000 8.000 9.000 8.666 8.666 8.000 8.334 8.334 8.666 8.334 0.333 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0]

Table A.25: DUE scan results: ladder $n = 3$, configuration Cfg 3 ($k = 1$, pos=3): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	23.000	0.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 1 1 0 0 0 1 0 0 0]
2	26.000	26.000	-0.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 2 2.000 0 0 0 2.000 0 0 0]
3	29.000	28.500	-0.500	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 3 2.500 0 0 0.500 2.500 0.500 0.500 0]
4	30.800	30.998	0.198	true	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[3.999 0.001 3.999 3.999 3 0.001 0.001 1 3 1 0.999 0]
5	31.999	32.166	0.167	true	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[4.333 0.667 4.333 4.333 3.500 0.667 0.667 1.500 3.500 1.500 0.833 0]
6	33.199	33.333	0.134	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[4.666 1.334 4.666 4.666 4.000 1.334 1.334 2.000 4.000 2.000 0.666 0]
7	34.400	34.500	0.100	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[5.000 2.000 5.000 5.000 4.500 2.000 2.000 2.500 4.500 2.500 0.500 0]
8	35.600	35.664	0.064	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[5.331 2.669 5.331 5.331 4.997 2.669 2.669 3.003 4.997 3.003 0.334 0]
9	36.800	36.833	0.033	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[5.667 3.333 5.667 5.667 5.500 3.333 3.333 3.500 5.500 3.500 0.167 0]
10	38.000	38.000	0.000	false	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000 0.000 0]
11	39.200	39.200	0.000	false	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600 0 0]
12	40.401	40.401	0.000	false	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199 0 0]
13	41.600	41.600	0.000	false	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800 0 0]
14	42.800	42.800	0.000	false	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400 0 0]
15	44.000	44.000	0.000	false	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000 0 0]
16	45.200	45.200	0.000	false	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600 0 0]
17	46.400	46.400	0.000	false	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0]

Table A.26: DUE scan results: ladder $n = 3$, configuration Cfg 4 ($k = 1$, pos=4): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	16.000	-7.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 1 1 0 0 0 0 1 1 0]
2	26.000	20.000	-6.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 2 2 0 0 0 0 2 2 0]
3	29.000	24.000	-5.000	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 3 3 0 0 0 0 3 3 0]
4	30.800	28.000	-2.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 4 4 0 0 0 0 4 4 0]
5	31.999	32.000	0.001	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 5 5 0 0 0 0 5 5 0]
6	33.199	36.000	2.800	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6.000 0.000 6.000 6.000 6.000 0.000 0.000 0.000 0 6 6.000 0]
7	34.400	37.750	3.350	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[6.25 0.75 6.25 6.25 6.25 0.75 0.75 0.75 0 7 6.25 0]
8	35.600	39.499	3.898	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[6.500 1.500 6.500 6.500 6.500 1.500 1.500 1.500 0.000 8.000 6.499 0]
9	36.800	40.249	3.449	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[6.75 2.25 6.75 6.75 6.75 2.25 2.25 2.25 1.001 7.999 5.749 0]
10	38.000	40.999	2.999	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[6.999 3.001 6.999 6.999 6.999 3.001 3.001 3.001 2.000 8.000 4.999 0]
11	39.200	41.750	2.550	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[7.250 3.750 7.250 7.250 7.250 3.750 3.750 3.750 3.000 8.000 4.250 0]
12	40.401	42.500	2.099	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[7.500 4.500 7.500 7.500 7.500 4.500 4.500 4.500 4.001 7.999 3.500 0]
13	41.600	43.250	1.650	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[7.750 5.250 7.750 7.750 7.750 5.250 5.250 5.250 5.000 8.000 2.750 0]
14	42.800	44.000	1.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[8.000 6.000 8.000 8.000 8.000 6.000 6.000 6.000 6.000 8.000 2.000 0]
15	44.000	44.750	0.750	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[8.250 6.750 8.250 8.250 8.250 6.750 6.750 6.750 7.000 8.000 1.250 0]
16	45.200	45.500	0.300	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[8.500 7.500 8.500 8.500 8.500 7.500 7.500 7.500 8.000 8.000 0.500 0]
17	46.400	46.400	0.000	false	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0]

Table A.27: DUE scan results: ladder $n = 3$, configuration Cfg 5 ($k = 2$, pos=`[1 2]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	16.000	-7.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 0 0 0 1 1 0 1 0 0 1 0]
2	26.000	20.000	-6.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 0 0 0 2 2 0 2 0 0 2 0]
3	29.000	24.000	-5.000	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 0 0 0 3 3 0 3 0 0 3 0]
4	30.800	28.000	-2.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 0.000 0.000 0 4.000 4.000 0.000 4.000 0 0 4.000 0]
5	31.999	30.666	-1.333	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 0.667 0.667 0 4.333 4.333 0.667 4.333 0 0 4.333 0]
6	33.199	33.333	0.134	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 6 1.333 1.333 0 4.667 4.667 1.333 4.667 0 0 4.667 0]
7	34.400	36.000	1.600	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 7 2.000 2.000 0 5.000 5.000 2.000 5.000 0 0 5.000 0]
8	35.600	38.666	3.066	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8 0 8 2.667 2.667 0 5.333 5.333 2.667 5.333 0 0 5.333 0]
9	36.800	41.332	4.532	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[8.999 0.001 8.999 3.333 3.333 0.001 5.667 5.667 3.333 5.667 0 0 5.666 0]
10	38.000	42.000	4.000	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[9.000 1.000 9.000 4.000 4.000 1.000 6.000 6.000 4.000 6.000 0 0 5.000 0]
11	39.200	42.667	3.466	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[9.000 2.000 9.000 4.667 4.667 2.000 6.333 6.333 4.667 6.333 0 0 4.333 0]
12	40.401	43.333	2.932	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[9.000 3.000 9.000 5.333 5.333 3.000 6.667 6.667 5.333 6.667 0 0 3.667 0]
13	41.600	44.000	2.400	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[9.000 4.000 9.000 6.000 6.000 4.000 7.000 7.000 6.000 7.000 0 0 3.001 0.001]
14	42.800	44.666	1.866	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[9.000 5.000 9.000 6.667 6.667 5.000 7.333 7.333 6.667 7.333 0 0 2.333 0]
15	44.000	45.334	1.334	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[9.000 6.000 9.000 7.333 7.333 6.000 7.667 7.667 7.333 7.667 0 0 1.667 0]
16	45.200	46.000	0.800	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[9.000 7.000 9.000 8.000 8.000 7.000 8.000 8.000 8.000 8.000 0 0 1.000 0]
17	46.400	46.667	0.266	true	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[9.000 8.000 9.000 8.666 8.666 8.000 8.334 8.334 8.666 8.334 0 0 0.333 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0 0 0]

Table A.28: DUE scan results: ladder $n = 3$, configuration Cfg 6 ($k = 2$, pos=`[1 3]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	23.000	-0.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1.000 1.000 1.000 0 0 0 1.000 0 0 0 0 0]
2	26.000	25.000	-1.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 1.5 1.5 1.5 0.5 0.5 0.5 1.5 0.5 0.5 0 0 0]
3	29.000	27.001	-1.999	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 2.001 2.001 2.001 0.999 0.999 0.999 2.001 0.999 0.999 0 0 0]
4	30.800	29.000	-1.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 2.500 2.500 2.500 1.500 1.500 1.500 2.500 1.500 1.500 0 0 0]
5	31.999	31.000	-0.999	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 3.000 3.000 3.000 2.000 2.000 2.000 3.000 2.000 2.000 0 0 0]
6	33.199	33.001	-0.198	false	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 3.504 3.504 3.504 2.496 2.496 2.496 3.504 2.496 2.496 0 0 0]
7	34.400	35.000	0.600	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 4.000 4.000 4.000 3.000 3.000 3.000 4.000 3.000 3.000 0 0 0]
8	35.600	37.000	1.399	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[7.999 0.001 4.500 4.500 4.500 3.500 3.500 3.500 4.500 3.500 3.499 0 0 0]
9	36.800	38.000	1.200	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[8.000 1.000 5.000 5.000 5.000 4.000 4.000 4.000 5.000 4.000 3.000 0 0 0]
10	38.000	39.000	1.000	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[8.000 2.000 5.499 5.499 5.499 4.501 4.501 4.501 5.499 4.501 2.500 0 0 0]
11	39.200	40.000	0.800	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[8.000 3.000 6.000 6.000 6.000 5.000 5.000 5.000 6.000 5.000 2.000 0 0 0]
12	40.401	41.000	0.599	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[8.000 4.000 6.500 6.500 6.500 5.500 5.500 5.500 6.500 5.500 1.500 0 0 0]
13	41.600	42.000	0.400	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[8.000 5.000 7.000 7.000 7.000 6.000 6.000 6.000 7.000 6.000 1.000 0 0 0]
14	42.800	43.000	0.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[8.000 6.000 7.500 7.500 7.500 6.500 6.500 6.500 7.500 6.500 0.500 0 0 0]
15	44.000	44.000	0.000	false	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000 0.000 0 0 0]
16	45.200	45.200	0.000	false	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600 0 0 0 0]
17	46.400	46.400	0.000	false	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200 0 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0 0 0]

Table A.29: DUE scan results: ladder $n = 3$, configuration Cfg 7 ($k = 2$, pos=`[1 4]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	16.000	-7.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 1 1 0 0 0 0 1 0 0 1 0]
2	26.000	20.000	-6.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 2 2 0 0 0 0 2 0 0 2 0]
3	29.000	24.000	-5.000	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 3 3 0 0 0 0 3 0 0 3 0]
4	30.800	28.000	-2.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 4 4 0 0 0 0 4 0 0 4 0]
5	31.999	32.000	0.000	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5.000 5.000 5.000 0.000 0.000 0.000 0 5 0.000 0 5.000 0]
6	33.199	34.666	1.467	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 5.333 5.333 5.333 0.667 0.667 0.667 0 6 0.667 0 5.333 0]
7	34.400	37.333	2.933	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 5.667 5.667 5.667 1.333 1.333 1.333 0 7 1.333 0 5.667 0]
8	35.600	39.999	4.399	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8.000 0.000 6 6 6 2 2 2 0.000 8.000 2.000 0 6.000 0]
9	36.800	40.664	3.865	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[7.998 1.002 6.333 6.333 6.333 2.667 2.667 2.667 1.000 8.000 1.666 0 5.333 0]
10	38.000	41.333	3.333	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[8.000 2.000 6.667 6.667 6.667 3.333 3.333 3.333 2.000 8.000 1.334 0 4.666 0]
11	39.200	42.000	2.800	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[8.000 3.000 7.000 7.000 7.000 4.000 4.000 4.000 3.000 8.000 1.000 0 4.000 0]
12	40.401	42.670	2.269	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[7.994 4.006 7.337 7.337 7.337 4.663 4.663 4.663 3.994 8.006 0.657 0 3.343 0]
13	41.600	43.333	1.733	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[8.000 5.000 7.666 7.666 7.666 5.334 5.334 5.334 5.000 8.000 0.334 0 2.666 0]
14	42.800	44.000	1.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[8.000 6.000 8.000 8.000 8.000 6.000 6.000 6.000 5.999 8.001 0.000 0 2.001 0.001]
15	44.000	44.750	0.750	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[8.250 6.750 8.250 8.250 8.250 6.750 6.750 6.750 7.000 8.000 0 0 1.250 0]
16	45.200	45.500	0.300	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[8.500 7.500 8.500 8.500 8.500 7.500 7.500 7.500 8.000 8.000 0 0 0.500 0]
17	46.400	46.400	0.000	false	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200 0 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0 0 0]

Table A.30: DUE scan results: ladder $n = 3$, configuration Cfg 8 ($k = 2$, pos=`[2 3]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	16.000	-7.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 0 0 0 1 1 0 1 1 0 0 0 0]
2	26.000	20.000	-6.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 0 0 0 2 2.000 0 2.000 2 0 0 0]
3	29.000	23.500	-5.500	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 0 0.500 0 3 2.500 0.500 2.500 3 0 0 0.500]
4	30.800	27.000	-3.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 0 1.000 0 4 3.000 1.000 3.000 4 0 0 1.000]
5	31.999	30.500	-1.499	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 0 1.500 0 5 3.500 1.500 3.500 5 0 0 1.500]
6	33.199	34.000	0.801	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 6 0.000 2.000 0 6.000 4.000 2.000 4.000 6.000 0 0 2]
7	34.400	36.500	2.100	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 7 1.000 2.500 0 6.000 4.500 2.500 4.500 6.000 0 0 1.500]
8	35.600	39.000	3.399	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8 0 8 2.000 3.000 0 6.000 5.000 3.000 5.000 6.000 0 0 1.000]
9	36.800	41.497	4.697	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[8.999 0.001 8.999 3.000 3.501 0.001 6.000 5.499 3.501 5.499 5.998 0 0 0.501]
10	38.000	42.000	4.000	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[9.000 1.000 9.000 4.000 4.000 1.000 6.000 6.000 4.000 6.000 5.000 0 0 0.000]
11	39.200	42.667	3.466	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[9.000 2.000 9.000 4.667 4.667 2.000 6.333 6.333 4.667 6.333 4.333 0 0 0]
12	40.401	43.333	2.932	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[9.000 3.000 9.000 5.333 5.333 3.000 6.667 6.667 5.333 6.667 3.667 0 0 0]
13	41.600	44.000	2.400	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[9.000 4.000 9.000 6.000 6.000 4.000 7.000 7.000 6.000 7.000 3.001 0.001 0 0]
14	42.800	44.666	1.866	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[9.000 5.000 9.000 6.667 6.667 5.000 7.333 7.333 6.667 7.333 2.333 0 0 0]
15	44.000	45.334	1.334	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[9.000 6.000 9.000 7.333 7.333 6.000 7.667 7.667 7.333 7.667 1.667 0 0 0]
16	45.200	46.000	0.800	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[9.000 7.000 9.000 8.000 8.000 7.000 8.000 8.000 8.000 8.000 1.000 0 0 0]
17	46.400	46.667	0.266	true	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[9.000 8.000 9.000 8.666 8.666 8.000 8.334 8.334 8.666 8.334 0.333 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0 0 0]

Table A.31: DUE scan results: ladder $n = 3$, configuration Cfg 9 ($k = 2$, pos=`[2 4]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	15.500	-7.500	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 0.5 0.5 0 0.5 0.5 0 1 0.5 0 0.5 0]
2	26.000	19.000	-7.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 1 1 0 1 1 0 2 1 0 1 0]
3	29.000	22.500	-6.500	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 1.5 1.5 0 1.5 1.5 0 3 1.5 0 1.5 0]
4	30.800	26.000	-4.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 2 2 0 2 2 0 4 2 0 2 0]
5	31.999	29.500	-2.499	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 2.5 2.5 0 2.5 2.5 0 5 2.5 0 2.5 0]
6	33.199	33.000	-0.199	false	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 6 3 3 0 3 3 0 6 3 0 3 0]
7	34.400	36.500	2.100	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 7 3.5 3.5 0 3.5 3.5 0 7 3.5 0 3.5 0]
8	35.600	40.000	4.400	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8 0 8 4.000 4.000 0 4.000 4.000 0.000 8.000 4.000 0 4.000 0]
9	36.800	42.500	5.700	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[9.000 0.000 9.000 4.500 4.500 0.000 4.500 4.500 1.000 8.000 4.500 0 3.500 0]
10	38.000	42.999	4.999	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[9.000 1.000 9.000 5.000 5.000 1.000 5.000 5.000 2.000 8.000 4.000 0 3.000 0]
11	39.200	43.499	4.299	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[9.000 2.000 9.000 5.500 5.500 2.000 5.500 5.500 3.000 8.000 3.499 0 2.500 0]
12	40.401	44.000	3.599	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[9 3 9 6.000 6.000 3 6.000 6.000 4.000 8.000 3.001 0.001 2.000 0]
13	41.600	44.498	2.898	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[8.998 4.002 8.998 6.498 6.498 4.002 6.502 6.502 4.999 8.001 2.500 0 1.499 0]
14	42.800	45.000	2.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[9.000 5.000 9.000 7.000 7.000 5.000 7.000 7.000 6.000 8.000 2.000 0 1.000 0]
15	44.000	45.500	1.500	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[9.000 6.000 9.000 7.500 7.500 6.000 7.500 7.500 7.000 8.000 1.500 0 0.500 0]
16	45.200	46.001	0.801	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[9.001 6.999 9.001 8.000 8.000 6.999 8.000 8.000 8.000 8.000 1.001 0 0.000 0]
17	46.400	46.667	0.266	true	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[9.000 8.000 9.000 8.666 8.666 8.000 8.334 8.334 8.666 8.334 0.333 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0 0 0]

Table A.32: DUE scan results: ladder $n = 3$, configuration Cfg 10 ($k = 2$, pos=`[3 4]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	16.000	-7.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 1 1 0 0 0 0 1 0 0 1 0]
2	26.000	20.000	-6.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 2 2 0 0 0 0 2 0 0 2 0]
3	29.000	24.000	-5.000	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 3 3 0 0 0 0 3 0 0 3 0]
4	30.800	28.000	-2.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 4 4 0 0 0 0 4 0 0 4 0]
5	31.999	32.000	0.001	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 5 5 0 0 0 0 5 0 0 5 0]
6	33.199	36.000	2.800	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6.000 0.000 6.000 6.000 6.000 0.000 0.000 0 0 6 0 0 6.000 0]
7	34.400	37.750	3.350	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[6.25 0.75 6.25 6.25 6.25 0.75 0.75 0.75 0 7 0 0 6.25 0]
8	35.600	39.499	3.898	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[6.500 1.500 6.500 6.500 6.500 1.500 1.500 1.500 0.000 8.000 0 0 6.499 0]
9	36.800	40.249	3.449	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[6.75 2.25 6.75 6.75 6.75 2.25 2.25 2.25 1.001 7.999 0 0 5.749 0]
10	38.000	40.999	2.999	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[6.999 3.001 6.999 6.999 6.999 3.001 3.001 3.001 2.000 8.000 0 0 4.999 0]
11	39.200	41.750	2.550	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[7.250 3.750 7.250 7.250 7.250 3.750 3.750 3.750 3.000 8.000 0 0 4.250 0]
12	40.401	42.500	2.099	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[7.500 4.500 7.500 7.500 7.500 4.500 4.500 4.500 4.001 7.999 0 0 3.500 0]
13	41.600	43.250	1.650	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[7.750 5.250 7.750 7.750 7.750 5.250 5.250 5.250 5.000 8.000 0 0 2.750 0]
14	42.800	44.000	1.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[8.000 6.000 8.000 8.000 8.000 6.000 6.000 6.000 6.000 8.000 0 0 2.000 0]
15	44.000	44.750	0.750	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[8.250 6.750 8.250 8.250 8.250 6.750 6.750 6.750 7.000 8.000 0 0 1.250 0]
16	45.200	45.500	0.300	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[8.500 7.500 8.500 8.500 8.500 7.500 7.500 7.500 8.000 8.000 0 0 0.500 0]
17	46.400	46.400	0.000	false	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200 0 0 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 10 0 0 0 0 0]

Table A.33: DUE scan results: ladder $n = 3$, configuration Cfg 11 ($k = 3$, pos=`[1 2 3]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	16.000	-7.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 0 0 0 1 1 0 1 0 0 1 0 0 0]
2	26.000	20.000	-6.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 0 6.0 0 2 2.000 0 2.000 0 0 2 0 0 0]
3	29.000	23.500	-5.500	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 0 0.500 0 3 2.500 0.500 2.500 0 0 3 0 0 0.500]
4	30.800	27.000	-3.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 0 1.000 0 4 3.000 1.000 3.000 0 0 4 0 0 1.000]
5	31.999	30.500	-1.499	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 0 1.500 0 5 3.500 1.500 3.500 0 0 5 0 0 1.500]
6	33.199	34.000	0.801	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 6 0.000 2.000 0 6.000 4.000 2.000 4.000 0 0 6.000 0 0 2]
7	34.400	36.500	2.100	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 7 1.000 2.500 0 6.000 4.500 2.500 4.500 0 0 6.000 0 0 1.500]
8	35.600	39.000	3.399	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8 0 8 2.000 3.000 0 6.000 5.000 3.000 5.000 0 0 6.000 0 0 1.000]
9	36.800	41.497	4.697	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[8.999 0.001 8.999 3.000 3.501 0.001 6.000 5.499 3.501 5.499 0 0 5.998 0 0 0.501]
10	38.000	42.000	4.000	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[9.000 1.000 9.000 4.000 4.000 1.000 6.000 6.000 4.000 6.000 0 0 5.000 0 0 0.000]
11	39.200	42.667	3.466	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[9.000 2.000 9.000 4.667 4.667 2.000 6.333 6.333 4.667 6.333 0 0 4.333 0 0 0]
12	40.401	43.333	2.932	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[9.000 3.000 9.000 5.333 5.333 3.000 6.667 6.667 5.333 6.667 0 0 3.667 0 0 0]
13	41.600	44.000	2.400	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[9.000 4.000 9.000 6.000 6.000 4.000 7.000 7.000 6.000 7.000 0 0 3.001 0.001 0 0]
14	42.800	44.666	1.866	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[9.000 5.000 9.000 6.667 6.667 5.000 7.333 7.333 6.667 7.333 0 0 2.333 0 0 0]
15	44.000	45.334	1.334	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[9.000 6.000 9.000 7.333 7.333 6.000 7.667 7.667 7.333 7.667 0 0 1.667 0 0 0]
16	45.200	46.000	0.800	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[9.000 7.000 9.000 8.000 8.000 7.000 8.000 8.000 8.000 8.000 0 0 1.000 0 0 0]
17	46.400	46.667	0.266	true	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[9.000 8.000 9.000 8.666 8.666 8.000 8.334 8.334 8.666 8.334 0 0 0.333 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0 0 0 0 0 0]

Table A.34: DUE scan results: ladder $n = 3$, configuration Cfg 12 ($k = 3$, pos=`[1 2 4]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	15.500	-7.500	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 0.5 0 0.5 0 0 0.5 0 0.5 0 1 0 0 0.5 0 0 0.5 0]
2	26.000	19.000	-7.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 1 1 0 1 1 0 2 0 0 1 0 1 0]
3	29.000	22.500	-6.500	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 1.5 1.5 0 1.5 1.5 0 3 0 0 1.5 0 1.5 0]
4	30.800	26.000	-4.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 2 2 0 2 2 0 4 0 0 2 0 2 0]
5	31.999	29.500	-2.499	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 2.5 2.5 0 2.5 2.5 0 5 0 0 2.5 0 2.5 0]
6	33.199	33.000	-0.199	false	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 6 3 3 0 3 3 0 6 0 0 3 0 3 0]
7	34.400	36.500	2.100	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 7 3.5 3.5 0 3.5 3.5 0 7 0 0 3.5 0 3.5 0]
8	35.600	40.000	4.400	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8 0 8 4.000 4.000 0 4.000 4.000 0.000 8.000 0 0 4.000 0 4.000 0]
9	36.800	42.500	5.700	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[9.000 0.000 9.000 4.500 4.500 0.000 4.500 4.500 1.000 8.000 0 0 4.500 0 3.500 0]
10	38.000	42.999	4.999	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[9.000 1.000 9.000 5.000 5.000 1.000 5.000 5.000 2.000 8.000 0 0 4.000 0 3.000 0]
11	39.200	43.499	4.299	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[9.000 2.000 9.000 5.500 5.500 2.000 5.500 5.500 3.000 8.000 0 0 3.499 0 2.500 0]
12	40.401	44.000	3.599	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[9 3 9 6.000 6.000 3 6.000 6.000 4.000 8.000 0 0 3.001 0.001 2.000 0]
13	41.600	44.498	2.898	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[8.998 4.002 8.998 6.498 6.498 4.002 6.502 6.502 4.999 8.001 0 0 2.500 0 1.499 0]
14	42.800	45.000	2.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[9.000 5.000 9.000 7.000 7.000 5.000 7.000 7.000 6.000 8.000 0 0 2.000 0 1.000 0]
15	44.000	45.500	1.500	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[9.000 6.000 9.000 7.500 7.500 6.000 7.500 7.500 7.000 8.000 0 0 1.500 0 0.500 0]
16	45.200	46.001	0.801	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[9.001 6.999 9.001 8.000 8.000 6.999 8.000 8.000 8.000 8.000 0 0 1.001 0 0.000 0]
17	46.400	46.667	0.266	true	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[9.000 8.000 9.000 8.666 8.666 8.000 8.334 8.334 8.666 8.334 0 0 0.333 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 10 0 0 0 0 0 0 0]

Table A.35: DUE scan results: ladder $n = 3$, configuration Cfg 13 ($k = 3$, pos=`[1 3 4]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	16.000	-7.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 1 1 0 0 0 0 1 0 0 0 0 1 0]
2	26.000	20.000	-6.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 2 2 0 0 0 0 2 0 0 0 0 2 0]
3	29.000	24.000	-5.000	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 3 3 0 0 0 0 3 0 0 0 0 3 0]
4	30.800	28.000	-2.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4.000 4.000 4 0.000 0.000 0 0 4 0.000 0 0 0.000 4 0]
5	31.999	31.500	-0.499	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 4.500 4.500 5 0.500 0.500 0 0 5 0.500 0 0 0.500 5 0]
6	33.199	35.000	1.800	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 5.000 5.000 6.000 1.000 1.000 0.000 0 6 1.000 0 0 1.000 6.000 0]
7	34.400	37.500	3.100	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 5.500 5.500 6.000 1.500 1.500 1.000 0 7 1.500 0 0 0.500 6.000 0]
8	35.600	39.999	4.399	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8.000 0.000 6.000 6.000 6.000 2.000 2.000 2.000 0.000 2.000 0.000 8.000 2.000 0 0 0.000 6.000 0]
9	36.800	40.664	3.865	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[7.998 1.002 6.333 6.333 6.333 2.667 2.667 2.667 1.000 8.000 1.666 0 0 0 5.333 0]
10	38.000	41.333	3.333	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[8.000 2.000 6.667 6.667 6.667 3.333 3.333 3.333 2.000 8.000 1.334 0 0 0 4.666 0]
11	39.200	42.000	2.800	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[8.000 3.000 7.000 7.000 7.000 4.000 4.000 4.000 3.000 8.000 1.000 0 0 0 4.000 0]
12	40.401	42.670	2.269	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[7.994 4.006 7.337 7.337 7.337 4.663 4.663 4.663 3.994 8.006 0.657 0 0 0 3.343 0]
13	41.600	43.333	1.733	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[8.000 5.000 7.666 7.666 7.666 5.334 5.334 5.334 5.000 8.000 0.334 0 0 0 2.666 0]
14	42.800	44.000	1.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[8.000 6.000 8.000 8.000 8.000 6.000 6.000 6.000 5.999 8.001 0.000 0 0 0 2.001 0.001]
15	44.000	44.750	0.750	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[8.250 6.750 8.250 8.250 8.250 6.750 6.750 6.750 7.000 8.000 0 0 0 0 1.250 0]
16	45.200	45.500	0.300	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[8.500 7.500 8.500 8.500 8.500 7.500 7.500 7.500 8.000 8.000 0 0 0 0 0.500 0]
17	46.400	46.400	0.000	false	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200 0 0 0 0 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0 0 0 0 0 0]

Table A.36: DUE scan results: ladder $n = 3$, configuration Cfg 14 ($k = 3$, pos=`[2 3 4]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	11.000	-12.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 0]
2	26.000	16.000	-10.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 0 2 0 2 0 0 2 2 0 0 2 2 0]
3	29.000	21.000	-8.000	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 0 3 0 3 0 0 3 3 0 0 3 3 0]
4	30.800	26.000	-4.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 0 4 0 4 0 0 4 4 0 0 4 4 0]
5	31.999	31.000	-0.999	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 0 5 0 5 0 0 5 5 0 0 5 5 0]
6	33.199	36.000	2.801	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 6 0.000 6.000 0 6.000 0.000 0 6 6.000 0 0 6.000 6.000 0]
7	34.400	39.000	4.600	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 7 1.000 6.000 0 6.000 1.000 0 7 6.000 0 0 5.000 6.000 0]
8	35.600	42.000	6.400	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8 0 8 2.000 6.000 0 6.000 2.000 0.000 8.000 6.000 0 0 4 6.000 0]
9	36.800	43.979	7.180	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[8.994 0.006 8.994 2.997 5.993 0.006 6.003 3.007 1.006 7.994 5.997 0 0 2.997 4.987 0]
10	38.000	44.000	6.000	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[9.000 1.000 9.000 4.000 6.000 1.000 6.000 4.000 2.000 8.000 5.000 0 0 2.000 4.000 0]
11	39.200	44.000	4.800	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[9.000 2.000 9.000 5.000 6.000 2.000 6.000 5.000 3.000 8.000 4.000 0 0 1.000 3.000 0]
12	40.401	44.000	3.600	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[9 3 9 6 6.000 3 6 6.000 4.000 8.000 3.001 0.001 0 0.000 2.000 0]
13	41.600	44.498	2.898	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[8.998 4.002 8.998 6.5 6.5 4.002 6.5 6.5 5.000 8.000 2.498 0 0 0 1.500 0]
14	42.800	45.000	2.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[9.000 5.000 9.000 7.000 7.000 5.000 7.000 7.000 6.000 8.000 2.000 0 0 0 1.000 0]
15	44.000	45.500	1.500	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[9.000 6.000 9.000 7.500 7.501 6.000 7.500 7.499 7.000 8.000 1.500 0 0 0.001 0.500 0]
16	45.200	46.001	0.801	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[9.001 6.999 9.001 8.000 8.000 6.999 8.000 8.000 8.000 8.000 1.001 0 0 0 0.000 0]
17	46.400	46.667	0.267	true	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[9.000 8.000 9.000 8.666 8.666 8.000 8.334 8.334 8.666 8.334 0.334 0 0 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0 0 0 0 0]

Table A.37: DUE scan results: ladder $n = 3$, configuration Cfg 15 ($k = 4$, pos=`[1 2 3 4]`): full scan for $Q = 1, \ldots, 20$ (same columns as Table 5.1).

Q	$ATT^{(B)}$	$ATT^{(A)}$	ΔATT	Paradox	LinkFlow$^{(B)}$	LinkFlow$^{(A)}$
1	23.000	11.000	-12.000	false	[1 0 1 1 1 0 0 0 1 0]	[1 0 1 0 1 0 1 0 0 1 0 0 1 0 0 1 1 0]
2	26.000	16.000	-10.000	false	[2 0 2 2 2 0 0 0 2 0]	[2 0 2 0 2 0 2 0 0 2 0 0 2 0 0 2 2 0]
3	29.000	21.000	-8.000	false	[3 0 3 3 3 0 0 0 3 0]	[3 0 3 0 3 0 3 0 0 3 0 0 3 0 0 3 3 0]
4	30.800	26.000	-4.800	false	[3.6 0.4 3.6 3.6 3.6 0.4 0.4 0.4 3.6 0.4]	[4 0 4 0 4 0 4 0 0 4 0 0 4 0 0 4 4 0]
5	31.999	31.000	-0.999	false	[4.000 1.000 4.000 4.000 4.000 1.000 1.000 1.000 4.000 1.000]	[5 0 5 0 5 0 5 0 0 5 0 0 5 0 0 5 5 0]
6	33.199	36.000	2.801	true	[4.399 1.601 4.399 4.399 4.399 1.601 1.601 1.601 4.399 1.601]	[6 0 6 0.000 6.000 0 6.000 0.000 0 6 0 0 6.000 0 0 6.000 6.000 0]
7	34.400	39.000	4.600	true	[4.800 2.200 4.800 4.800 4.800 2.200 2.200 2.200 4.800 2.200]	[7 0 7 1.000 6.000 0 6.000 1.000 0 7 0 0 6.000 0 0 5.000 6.000 0]
8	35.600	42.000	6.400	true	[5.200 2.800 5.200 5.200 5.200 2.800 2.800 2.800 5.200 2.800]	[8 0 8 2.000 6.000 0 6.000 2.000 0.000 8.000 0 0 6.000 0 0 4 6.000 0]
9	36.800	43.979	7.180	true	[5.600 3.400 5.600 5.600 5.600 3.400 3.400 3.400 5.600 3.400]	[8.994 0.006 8.994 2.997 5.993 0.006 6.003 3.007 1.006 7.994 0 0 5.997 0 0 2.997 4.987 0]
10	38.000	44.000	6.000	true	[6.000 4.000 6.000 6.000 6.000 4.000 4.000 4.000 6.000 4.000]	[9.000 1.000 9.000 4.000 6.000 1.000 6.000 4.000 2.000 8.000 0 0 5.000 0 0 2.000 4.000 0]
11	39.200	44.000	4.800	true	[6.400 4.600 6.400 6.400 6.400 4.600 4.600 4.600 6.400 4.600]	[9.000 2.000 9.000 5.000 6.000 2.000 6.000 5.000 3.000 8.000 0 0 4.000 0 0 1.000 3.000 0]
12	40.401	44.000	3.600	true	[6.801 5.199 6.801 6.801 6.801 5.199 5.199 5.199 6.801 5.199]	[9 3 9 6 6.000 3 6 6.000 4.000 8.000 0 0 3.001 0.001 0 0.000 2.000 0]
13	41.600	44.498	2.898	true	[7.200 5.800 7.200 7.200 7.200 5.800 5.800 5.800 7.200 5.800]	[8.998 4.002 8.998 6.5 6.5 4.002 6.5 6.5 5.000 8.000 0 0 2.498 0 0 0 1.500 0]
14	42.800	45.000	2.200	true	[7.600 6.400 7.600 7.600 7.600 6.400 6.400 6.400 7.600 6.400]	[9.000 5.000 9.000 7.000 7.000 5.000 7.000 7.000 6.000 8.000 0 0 2.000 0 0 0 1.000 0]
15	44.000	45.500	1.500	true	[8.000 7.000 8.000 8.000 8.000 7.000 7.000 7.000 8.000 7.000]	[9.000 6.000 9.000 7.500 7.501 6.000 7.500 7.499 7.000 8.000 0 0 1.500 0 0 0.001 0.500 0]
16	45.200	46.001	0.801	true	[8.400 7.600 8.400 8.400 8.400 7.600 7.600 7.600 8.400 7.600]	[9.001 6.999 9.001 8.000 8.000 6.999 8.000 8.000 8.000 8.000 0 0 1.001 0 0 0 0.000 0]
17	46.400	46.667	0.267	true	[8.800 8.200 8.800 8.800 8.800 8.200 8.200 8.200 8.800 8.200]	[9.000 8.000 9.000 8.666 8.666 8.000 8.334 8.334 8.666 8.334 0 0 0.334 0 0 0 0 0]
18	47.600	47.600	0.000	false	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800]	[9.200 8.800 9.200 9.200 9.200 8.800 8.800 8.800 9.200 8.800 0 0 0 0 0 0 0 0 0 0]
19	48.800	48.800	0.000	false	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400]	[9.600 9.400 9.600 9.600 9.600 9.400 9.400 9.400 9.600 9.400 0 0 0 0 0 0 0 0 0 0]
20	50.000	50.000	0.000	false	[10 10 10 10 10 10 10 10 10 10]	[10 10 10 10 10 10 10 10 10 10 0 0 0 0 0 0 0 0 0 0]

B | Matlab code used in this thesis

B.1. Package overview and verification logic

This appendix provides the computational code used to produce and verify the numerical results reported in this thesis. The materials are organised by script/module: each section documents (i) what is computed, (ii) which parameters are user-editable, (iii) what outputs are generated, and (iv) the configuration adopted in the present submission.

For each script/module:

- run the script with the default configuration used in the thesis submission;
- verify that link legends, path enumeration, and convergence diagnostics are consistent;
- reproduce the scan tables and $\Delta ATT(Q)$ curves and compare against the corresponding items in the main text.

B.2. Script 1: `goal2_main.m`

B.2.1. What this script computes

The script `goal2_main.m` is a self-contained runner for equilibrium-based before/after comparisons under structural interventions (added directed links). For each demand level Q in a user-defined scan grid, it computes:

- the equilibrium link-flow vector in the *before* network and the average travel time $ATT^{(B)}$;
- the equilibrium link-flow vector in the *after* network and the average travel time $ATT^{(A)}$;
- the performance shift $\Delta ATT = ATT^{(A)} - ATT^{(B)}$ and a paradox indicator.

Two assignment models are supported: deterministic user equilibrium (DUE) and stochastic user equilibrium (SUE, Logit with dispersion parameter θ). Two network families are supported: diamond and ladder (including the transition case $n = 0$).

B.2.2. How to run and verify

Step 1: execute. Run `Code/goal2_main.m`. The script prints the BEFORE/AFTER link legends (index and directed edge `u->v`) and reports the number of enumerated simple r–s paths.

Step 2: check link ordering. All per-link cost arrays must follow the *AFTER link ordering* printed by the script. This design makes manual verification possible: the reported link-flow vectors can be checked entry-by-entry against the printed legend.

Step 3: reproduce scan outputs. For each Q, the script outputs a scan table containing $ATT^{(B)}$, $ATT^{(A)}$, ΔATT, paradox flag, and link-flow vectors (before/after). A $\Delta ATT(Q)$ plot is produced for visual comparison across cases/configurations.

B.2.3. User-editable parameters (what to edit)

All parameters are collected under the structure `CFG` at the top of the script.

Network.

- `CFG.net.kind='diamond'|'ladder'`
- (diamond) `CFG.net.dir='oneway'|'twoway'`
- (ladder) `CFG.net.n` (with `n=0` as transition case), `CFG.net.dir_ladder='oneway'|'twoway'`
- intervention edges: `CFG.net.added_edges_nodes=[uv;...]`

Demand scan and assignment.

- demand grid: `CFG.Q=q_min:q_step:q_max`
- assignment: `CFG.assign.kind='DUE'|'SUE'`
- SUE dispersion: `CFG.assign.theta` (only if SUE)

Per-link costs.

- per-link type: `CFG.cost.link_type\{i\}='linear'|'bpr'`
- linear parameters: `CFG.cost.b(i)`, `CFG.cost.m(i)`
- BPR parameters: `CFG.cost.t0(i)`, `CFG.cost.cap(i)`, with global `CFG.cost.alpha`, `CFG.cost.beta`
- optional ladder autofill: `CFG.cost.auto_ladder=true|false`

Thresholds and solver settings.

- paradox tolerance: `CFG.opts.paradox_eps`
- numerical zero band: `CFG.opts.zero_tol`
- convergence and iteration limits: `CFG.opts.tol`, `CFG.opts.max_iter`, and restart controls

B.2.4. Outputs (what is produced)

For each configuration/case, the script produces a scan table with columns:

- `Q`, `AvgTT_before`, `AvgTT_after`, `Delta_AvgTT`, `IsParadox`
- `theta` (NaN for DUE)
- `LinkFlow_before`, `LinkFlow_after` (vectors formatted as strings following the printed legend)

It also produces debug plots (network diagram) and a $\Delta ATT(Q)$ curve plot.

B.2.5. Configuration used in the present submission

The default submission configuration is encoded in the `CFG_list` block at the top of the script. Readers can reproduce the thesis results by running the script without modifying any parameters, and then matching the printed scan tables and figures with those reported in the main text.

B.2.6. Source code listing

Listing B.1: Script 1: Unified runner and helper functions (`goal2_main.m`)

```
function goal2_main()
% Goal-2 unified runner (Diamond / ladder-unified + per-link costs + DUE/SUE)
% - Keeps your current multi-config overlay workflow.
% - Ladder settings (dir/workflow) match the uploaded ladder runner.
%
% Notes:
%   * For LADDER runs, set CFG.net.kind='ladder'. Ladder direction is read from
%     CFG.net.dir_ladder (defaults to 'twoway') so diamond can stay 'oneway'.
%   * For LADDER runs, costs can be auto-filled using the Toyota-style T-model
%     (CFG.cost.auto_ladder=true), matching the uploaded file behaviour.

clc; clear; close all; clearvars;

% ======================
% CONFIG
% ======================

% NETWORK PARAMETERS
CFG.net.kind = 'diamond';     % 'diamond' | 'ladder'

% Shared direction (diamond)
CFG.net.dir = 'oneway';       % 'oneway' | 'twoway'

% Ladder-only params (ignored for diamond)
CFG.net.n         = 3;        % ladder segments
```

```
CFG.net.dir_ladder= 'twoway'; % match uploaded ladder settings
CFG.net.added_edges_nodes = [];   % workflow (below) will populate for ladder

% Ladder-only: workflow that generates added edges (matches uploaded file)
CFG.workflow = struct( ...
    'mode', 'single', ...        % 'single'|'cumulative'|'subsets'
    'include_empty', true, ...   % include "no added edges" config
    'positions', [] ...          % if empty => all bypass positions
);

% COST FUNCTION PARAMETERS
CFG.cost.type = 'per-link';

% Per-edge cost setup (diamond: 5 edges; ladder: auto-filled if enabled)
CFG.cost.link_type = {'linear','linear','linear','linear','linear'};

CFG.cost.b = [0, 50, 50, 0, 10];
CFG.cost.m = [10, 1, 1,10, 1];

CFG.cost.t0  = [NaN, NaN, NaN, NaN, NaN];
CFG.cost.cap = [NaN, NaN, NaN, NaN, NaN];

CFG.cost.alpha = 0.15;
CFG.cost.beta  = 4.0;


% Added edges override (kept from your code)
CFG.cost.added_override = struct( ...
  'link_type', {{}}, ...
  't0',        [],  ...
  'cap',       [],  ...
  'b',         [],  ...
  'm',         []   ...
);

% ASSIGNMENT PARAMETERS
CFG.assign.kind  = 'DUE';
CFG.assign.theta = 5;

% DEMAND SCAN
CFG.Q = 1:1:20;

% SOLVER OPTIONS
CFG.opts = struct( ...
  'paradox_eps', 1e-3, ...
  'zero_tol',    1e-2, ...
  'tol',         1e-8, ...
  'max_iter',    5e4, ...
  'restart_k',   200, ...
  'restart_tol', 1e-6, ...
  'restart_max', 5, ...
  'refine_enable', false, ...
  'save', false);


% ======================
% RUN (multi-config + one figure overlay)
% ======================

CFG_list = {};

% ---- Case2: current CFG ----
CFG_list{end+1} = CFG;

CFG_2 = CFG;
CFG_2.net.kind       = 'ladder';
CFG_2.net.n          = 0;
CFG_2.net.dir_ladder = 'twoway';

```

```
CFG_2.net.added_edges_nodes = [1 2];
CFG_2.cost.auto_ladder = false;

CFG_2.cost.link_type = {'linear','linear','linear','linear','linear','linear'};
CFG_2.cost.b         = [0, 50, 50, 0, 10, 10];
CFG_2.cost.m         = [10, 1, 1, 10, 1, 1];
CFG_2.cost.t0        = [NaN, NaN, NaN, NaN, NaN, NaN];
CFG_2.cost.cap       = [NaN, NaN, NaN, NaN, NaN, NaN];

CFG_list{end+1} = CFG_2;


labels_custom = { ...
    'CLASSIC BRAESS ()', ...
    'LADDER n=0 (BIDRECTIONAL BYPASS)' ...
};


% Case 2: added-link-only (kept)
%CFG2 = CFG;
%CFG2.cost.added_override = struct('link_type',{{'bpr'}},'t0',[10],'cap',[2000],'b',NaN,'m',NaN);
%CFG_list{end+1} = CFG2;

% Case 3: linear + bpr (kept)
%CFG3 = CFG;
%bpr_idx = [1 2 3 4 5];
%for k = bpr_idx, CFG3.cost.link_type{k} = 'bpr'; end
%CFG3.cost.t0(bpr_idx)  = [40 50 50 40 10];
%CFG3.cost.cap(bpr_idx) = [2000 2000 2000 2000 2000];
%CFG3.cost.alpha = 0.15; CFG3.cost.beta = 4;
%CFG3.cost.b(bpr_idx) = NaN; CFG3.cost.m(bpr_idx) = NaN;
%CFG_list{end+1} = CFG3;

[ScanList, IntervalList, SummaryList, labels] = goal2_run_multi(CFG_list, labels_custom);
labels = labels_custom;

% Print summaries (optional)
print_full_scan = true;

for i = 1:numel(ScanList)
    fprintf('\n============================\n');
    fprintf('CASE %d: %s\n', i, labels{i});

    disp('=== Summary ===');
    disp(SummaryList{i});

    disp('=== Scan ===');
    if print_full_scan
        disp(ScanList{i});
    else
        disp(head(ScanList{i}));
    end

    if ~isempty(IntervalList{i})
        disp('=== Paradox Intervals ===');
        disp(IntervalList{i});
    end
end

% One-figure overlay
viz = struct();
viz.showParadoxMarkers = false;   % highlight IsParadox points
viz.legendLocation     = 'eastoutside'; % or 'best'
viz.lineWidth          = 1.8;
viz.markerSize         = 6;

plot_deltaATT_vs_Q_multi(ScanList, labels, viz);

end
```

```
164
165
166 % =========================================================
167 % CORE FUNCTIONS
168 % =========================================================
169
170 function [Scan, Intervals, Summary] = goal2_run(CFG)
171
172 opts        = getdef(CFG,'opts', struct());
173 paradox_eps = getdef(opts,'paradox_eps', 1e-3);
174 zero_tol    = getdef(opts,'zero_tol',    1e-2);
175 tol         = getdef(opts,'tol',         1e-8);
176 max_iter    = getdef(opts,'max_iter',    5e4);
177
178 % Parse Q values
179 Qv = CFG.Q;
180 if isstruct(Qv), Qv = Qv.min:Qv.step:Qv.max; end
181 Qv = Qv(:).';
182
183 % (0) Ladder: apply workflow to generate added edges (matches uploaded file)
184 if isfield(CFG,'net') && isfield(CFG.net,'kind') && strcmpi(CFG.net.kind,'ladder')
185     CFG.net = ladder_apply_workflow(CFG.net, getdef(CFG,'workflow', struct()));
186 end
187
188 % (1) Build networks
189 [linksB, linksA, s, t] = build_network_from_CFG(CFG.net);
190
191 fprintf('\n--- BEFORE link legend (index: u->v) ---\n');
192 print_link_legend(linksB);
193 fprintf('--- AFTER link legend (index: u->v) ---\n');
194 print_link_legend(linksA);
195
196 % (2) Visualize networks
197 added_idx = detect_added_edges(linksB, linksA);
198 plot_networks_for_debug(linksB, linksA, added_idx, s, t);
199
200 % (3) Enumerate paths (strict simple-path enumeration)
201 PMAX = 20000;
202 fprintf('\n[PATH ENUM] Enumerating paths in BEFORE network...\n');
203 B1 = incidence_from_all_paths_DAG(linksB, s, t, PMAX);
204 fprintf('[PATH ENUM] BEFORE: %d paths found.\n', size(B1,1));
205
206 fprintf('[PATH ENUM] Enumerating paths in AFTER network...\n');
207 B2 = incidence_from_all_paths_DAG(linksA, s, t, PMAX);
208 fprintf('[PATH ENUM] AFTER: %d paths found.\n', size(B2,1));
209
210 % DEBUG: Print sizes of B1 and B2
211 fprintf('[DEBUG] B1 size: %d x %d\n', size(B1,1), size(B1,2));
212 fprintf('[DEBUG] B2 size: %d x %d\n', size(B2,1), size(B2,2));
213 fprintf('[DEBUG] linksB edges: %d, linksA edges: %d\n', size(linksB,1), size(linksA,1));
214
215 assert(~isempty(B1) && size(B1,1)>0, 'No s->t path in BEFORE network.');
216 assert(~isempty(B2) && size(B2,1)>0, 'No s->t path in AFTER network.');
217
218 % (4) Build cost objects (Ladder auto cost-fill matches uploaded file)
219 [F_B, F_A] = build_costs_before_after(linksB, linksA, CFG.cost, added_idx, CFG.net);
220
221 % Validation
222 mA = size(linksA,1);
223 fprintf('[DEBUG] Cost validation: |F_A.link_type|=%d, mA=%d\n', numel(F_A.link_type), mA);
224 assert(numel(F_A.link_type)==mA, 'CFG.cost.link_type length must equal #linksA.');
225 assert(numel(F_A.b)==mA && numel(F_A.m)==mA && numel(F_A.t0)==mA && numel(F_A.cap)==mA, ...
226   'cost arrays (b,m,t0,cap) must have length == #linksA.');
227
228 % (5) Assignment loop
229 assign_kind = upper(CFG.assign.kind);
230 theta_def   = getdef(CFG.assign,'theta',0.35);
231
232 rows = {};
```

```
NQ = numel(Qv);
fprintf('\n=== START ASSIGNMENT over %d demand levels ===\n', NQ);

for k = 1:NQ
    Q = Qv(k);
    fprintf('\n[ASSIGN] Q = %g   (%d / %d)\n', Q, k, NQ);

    switch assign_kind
        case 'DUE'
            [xB, XB, cB, CB] = calc_DUE(F_B, B1, Q, tol, max_iter, [], restart_default());
            [xA, XA, cA, CA] = calc_DUE(F_A, B2, Q, tol, max_iter, [], restart_default());
            tval = NaN;
        case 'SUE'
            [xB, XB, cB, CB] = calc_SUE(F_B, B1, Q, tol, max_iter, [], theta_def, restart_default());
            [xA, XA, cA, CA] = calc_SUE(F_A, B2, Q, tol, max_iter, [], theta_def, restart_default());
            tval = theta_def;
        otherwise
            error('Unknown assignment type: %s', assign_kind);
    end

    % EXTENDED DEBUG OUTPUT
    fprintf('[DEBUG] Q=%g\n', Q);
    fprintf('  xB size: %s, sum(xB)=%g\n', mat2str(size(xB)), sum(xB));
    fprintf('  XB size: %s, sum(XB)=%g\n', mat2str(size(XB)), sum(XB));
    fprintf('  B1 size: %s (for verification)\n', mat2str(size(B1)));
    fprintf('  XB first 5 elements: [%s]\n', sprintf('%.6g ', XB(1:min(5,end))));
    fprintf('  XB is empty? %d\n', isempty(XB));

    fprintf('  xA size: %s, sum(xA)=%g\n', mat2str(size(xA)), sum(xA));
    fprintf('  XA size: %s, sum(XA)=%g\n', mat2str(size(XA)), sum(XA));
    fprintf('  B2 size: %s (for verification)\n', mat2str(size(B2)));
    fprintf('  XA first 5 elements: [%s]\n', sprintf('%.6g ', XA(1:min(5,end))));
    fprintf('  XA is empty? %d\n', isempty(XA));

    % Compute average travel times
    TTb_p = xB * CB';
    TTa_p = xA * CA';

    assert(abs(TTb_p - (xB * B1 * cB')) <= 1e-6*max(1,abs(TTb_p)), 'Before TT mismatch.');
    assert(abs(TTa_p - (xA * B2 * cA')) <= 1e-6*max(1,abs(TTa_p)), 'After TT mismatch.');

    avgB = TTb_p / max(Q, eps);
    avgA = TTa_p / max(Q, eps);
    dAvg = avgA - avgB;

    if abs(dAvg) <= zero_tol
        isParadox = false;
    else
        isParadox = (dAvg > paradox_eps);
    end

    fprintf('[ASSIGN] Q = %g done. AvgTT_B = %.4g, AvgTT_A = %.4g,  = %.4g, paradox = %d\n', ...
            Q, avgB, avgA, dAvg, isParadox);

    % Store with proper flow formatting
    rows(end+1,:) = {Q, avgB, avgA, dAvg, isParadox, tval, vecfmt(XB), vecfmt(XA)};
end

Scan = cell2table(rows, 'VariableNames', ...
 {'Q','AvgTT_before','AvgTT_after','Delta_AvgTT','IsParadox','theta', ...
  'LinkFlow_before','LinkFlow_after'});

Intervals = table();
Summary   = table();

if ~isempty(Qv)
    fprintf('\n--- Quick check at Q = %g ---\n', Qv(end));
    fprintf('Before link flows: %s\n', Scan.LinkFlow_before{end});
    fprintf('After  link flows: %s\n', Scan.LinkFlow_after{end});
```

```
end
end


% ========================================================
% NETWORK BUILDERS
% ========================================================

function [linksB, linksA, s, t] = build_network_from_CFG(netCFG)

kind = lower(netCFG.kind);
dirD = getdef(netCFG, 'dir', 'oneway');
dirL = getdef(netCFG, 'dir_ladder', 'twoway'); % match uploaded file

switch kind
    case 'diamond'
        [linksB, linksA, s, t] = build_links_diamond(dirD);
    case 'ladder'
        n     = getdef(netCFG, 'n', 3);
        nodes = getdef(netCFG, 'added_edges_nodes', []);
        if n == 0
            % n=0 transition matches uploaded file: "bidirectional-bypass diamond"
            [linksB, linksA, s, t] = build_links_transition_n0(dirL, nodes);
        else
            [linksB, linksA, s, t] = build_links_ladder_unified(n, dirL, nodes);
        end
    otherwise
        error('Unknown network kind: %s', kind);
end
end

function [linksB, linksA, s, t] = build_links_diamond(dir)

linksB = [1 2;
          2 4;
          1 3;
          3 4];

edgesA = [linksB;
          2 3];

if strcmpi(dir, 'twoway')
    edgesA = [edgesA;
              3 2];
end

linksA = unique(edgesA, 'rows', 'stable');
s = 1;
t = 4;
end

function [linksB, linksA, s, t] = build_links_transition_n0(dirLadder, added_edges_nodes)
% n=0 transition: mimic diamond with a bypass that can be bidirectional.
% Node mapping (for n=0):
%   U(1)=1, L(1)=2, S=3, T=4
% BEFORE: S->U1, S->L1, U1->T, L1->T
% AFTER:  + optional bypass edges from workflow/added_edges_nodes

n = 0;
U = @(i) i;        %#ok<NASGU>
L = @(j) j + (n+1); %#ok<NASGU>
S = 2*n + 3;  % 3
T = 2*n + 4;  % 4

U1 = 1;
L1 = 2;

edgesB = [S U1;
          S L1;
```

```
          U1 T;
          L1 T];

linksB = edgesB;

edgesA = linksB;
if ~isempty(added_edges_nodes)
    added = validate_and_clip_edges(added_edges_nodes, n);
    for r = 1:size(added, 1)
        u = added(r, 1);
        v = added(r, 2);
        edgesA = [edgesA; u v];
        if strcmpi(dirLadder, 'twoway')
            edgesA = [edgesA; v u];
        end
    end
end

linksA = unique(edgesA, 'rows', 'stable');

% Sanity: in transition, ensure both connectors exist (matches uploaded logic)
assert(ismember([S U1], linksA, 'rows') && ismember([S L1], linksA, 'rows'), ...
    'Transition n=0: missing S->U1 or S->L1 in AFTER.');

s = S; t = T;
end

function [linksB, linksA, s, t] = build_links_ladder_unified(n, dir, added_edges_nodes)

U = @(i) i;
L = @(j) j + (n+1);
S = 2*n + 3;
T = 2*n + 4;

edgesB = [];
edgesB = [edgesB; S U(1)];
edgesB = [edgesB; S L(1)];

for i = 1:n
    edgesB = [edgesB; U(i) U(i+1)];
end

for i = 1:n
    edgesB = [edgesB; L(i) L(i+1)];
end

edgesB = [edgesB; U(n+1) T];
edgesB = [edgesB; L(n+1) T];

linksB = edgesB;

edgesA = linksB;
if ~isempty(added_edges_nodes)
    added = validate_and_clip_edges(added_edges_nodes, n);
    for r = 1:size(added, 1)
        u = added(r, 1);
        v = added(r, 2);
        edgesA = [edgesA; u v];
        if strcmpi(dir, 'twoway')
            edgesA = [edgesA; v u];
        end
    end
end

linksA = unique(edgesA, 'rows', 'stable');
s = S;
t = T;
end

```

```
function newE = validate_and_clip_edges(newE, n)
maxNode = 2*n + 4;
bad = any(newE<1, 2) | any(newE>maxNode, 2) | (newE(:,1)==newE(:,2));
newE(bad, :) = [];
end

function netCFG = ladder_apply_workflow(netCFG, wf)
% Generate added_edges_nodes for ladder according to workflow.
% This matches the idea in the uploaded ladder runner: you specify mode and
% whether to include empty; edges are generated by bypass positions.

n = getdef(netCFG,'n',3);

% If user already provides explicit edges, keep them.
if isfield(netCFG,'added_edges_nodes') && ~isempty(netCFG.added_edges_nodes)
    return;
end

mode = getdef(wf,'mode','single');
include_empty = getdef(wf,'include_empty', true);

% Determine bypass positions
pos = getdef(wf,'positions', []);
if isempty(pos)
    pos = 1:(n+1); % bypass between Ui and Li at each column i
end
pos = unique(pos(:).');

U = @(i) i;
L = @(j) j + (n+1);

% Edge list for each position: bidirectional bypass
Epos = cell(numel(pos),1);
for k = 1:numel(pos)
    i = pos(k);
    Epos{k} = [U(i) L(i); L(i) U(i)];
end

switch lower(mode)
    case 'single'
        % default single: if include_empty, start with empty (no added)
        if include_empty
            netCFG.added_edges_nodes = [];
        else
            netCFG.added_edges_nodes = Epos{1};
        end
    case 'cumulative'
        % cumulative: add all positions
        E = [];
        for k = 1:numel(Epos)
            E = [E; Epos{k}]; %#ok<AGROW>
        end
        if include_empty
            netCFG.added_edges_nodes = [];
        else
            netCFG.added_edges_nodes = E;
        end
    case 'subsets'
        % subsets are handled in multi-run; here we keep empty by default
        if include_empty
            netCFG.added_edges_nodes = [];
        else
            netCFG.added_edges_nodes = Epos{1};
        end
    otherwise
        error('Unknown workflow mode: %s', mode);
end
end

```

```
509
510 % ========================================================
511 % COST FUNCTION
512 % ========================================================
513
514 function [F_B, F_A] = build_costs_before_after(linksB, linksA, costCFG, added_idx, netCFG)
515 % Build costs for BEFORE and AFTER.
516 % - For ladder, optionally auto-fill costs (Toyota T-model) to match uploaded file.
517
518 mA   = size(linksA, 1);
519 isAdd = ~isnan(added_idx);
520
521 if ~strcmpi(costCFG.type, 'per-link')
522     error('Only per-link cost is supported.');
523 end
524
525 % (A) Start from user per-link arrays for AFTER
526 F_A.kind      = 'per-link';
527 F_A.link_type = expand_cell(costCFG, 'link_type', mA, 'linear');
528 F_A.t0        = expand_vec(costCFG, 't0',   mA, NaN);
529 F_A.cap       = expand_vec(costCFG, 'cap',  mA, NaN);
530 F_A.alpha     = getdef(costCFG, 'alpha', 1.6);
531 F_A.beta      = getdef(costCFG, 'beta',  2.0);
532 F_A.b         = expand_vec(costCFG, 'b',    mA, NaN);
533 F_A.m         = expand_vec(costCFG, 'm',    mA, NaN);
534
535 % (B) Ladder auto cost-fill (matches uploaded behaviour)
536 if isfield(netCFG,'kind') && strcmpi(netCFG.kind,'ladder') && getdef(costCFG,'auto_ladder',false)
537     Tm = getdef(costCFG,'ladder_Tmodel', struct());
538     [F_A] = ladder_autofill_costs(linksA, F_A, netCFG, Tm);
539 end
540
541 % (C) Optional override for added edges (kept from your code)
542 if isfield(costCFG, 'added_override') && ~isempty(costCFG.added_override)
543     ov  = costCFG.added_override;
544     idx = find(isAdd);
545
546     if ~isempty(idx)
547         if isfield(ov, 'link_type') && ~isempty(ov.link_type)
548             tt = ov.link_type;
549             if ischar(tt), tt = {tt}; end
550             K = min(numel(tt), numel(idx));
551             for k = 1:K
552                 F_A.link_type{idx(k)} = tt{k};
553             end
554         end
555
556         if isfield(ov, 't0') && ~isempty(ov.t0)
557             K = min(numel(ov.t0), numel(idx));
558             F_A.t0(idx(1:K)) = ov.t0(1:K);
559         end
560
561         if isfield(ov, 'cap') && ~isempty(ov.cap)
562             K = min(numel(ov.cap), numel(idx));
563             F_A.cap(idx(1:K)) = ov.cap(1:K);
564         end
565
566         if isfield(ov, 'b') && ~isempty(ov.b)
567             b_val = ov.b;
568             if isscalar(b_val)
569                 F_A.b(idx) = b_val;
570             else
571                 K = min(numel(b_val), numel(idx));
572                 F_A.b(idx(1:K)) = b_val(1:K);
573             end
574         end
575
576         if isfield(ov, 'm') && ~isempty(ov.m)
577             m_val = ov.m;
```

```
            if isscalar(m_val)
                F_A.m(idx) = m_val;
            else
                K = min(numel(m_val), numel(idx));
                F_A.m(idx(1:K)) = m_val(1:K);
            end
        end
    end
end

% (D) BEFORE = subset of AFTER links that are in linksB
keep = false(mA,1);
for e = 1:mA
    keep(e) = ismember(linksA(e,:), linksB, 'rows');
end
F_B = sub_cost_object(F_A, keep);
end

function F = ladder_autofill_costs(linksA, F, netCFG, Tm)
% Toyota-style ladder T-model costs (uploaded file style):
% - connectors (S->U1, S->L1, Uend->T, Lend->T): linear 0
% - rails: BPR with chosen/default t0/cap, alpha/beta from F
% - bypass (Ui<->Li): often BPR too; we keep BPR unless user already set otherwise

n = getdef(netCFG,'n',3);
U = @(i) i;
L = @(j) j + (n+1);
S = 2*n + 3;
T = 2*n + 4;

default_t0  = getdef(Tm,'default_t0',50);
default_cap = getdef(Tm,'default_cap',2000);

for e = 1:size(linksA,1)
    u = linksA(e,1); v = linksA(e,2);

    isConnector = (u==S && (v==U(1) || v==L(1))) || ...
                  ((u==U(n+1) || u==L(n+1)) && v==T);

    if isConnector
        F.link_type{e} = 'linear';
        F.b(e) = 0; F.m(e) = 0;
        F.t0(e) = NaN; F.cap(e) = NaN;
        continue;
    end

    % If user already set a link_type explicitly (not default padding), respect it.
    tcur = lower(F.link_type{e});
    if isempty(tcur), tcur = 'linear'; end

    % Auto set rails/bypasses as BPR if not explicitly BPR already.
    if strcmpi(tcur,'linear') || strcmpi(tcur,'affine')
        % For ladder, default to BPR for non-connectors
        F.link_type{e} = 'bpr';
        F.t0(e)  = default_t0;
        F.cap(e) = default_cap;
        F.b(e) = NaN; F.m(e) = NaN;
    else
        % Ensure BPR fields exist
        if strcmpi(tcur,'bpr')
            if isnan(F.t0(e)),  F.t0(e)=default_t0;  end
            if isnan(F.cap(e)), F.cap(e)=default_cap; end
            F.b(e)=NaN; F.m(e)=NaN;
        end
    end
end
end

function F2 = sub_cost_object(F, keep)
```

```
F2           = F;
F2.link_type = F.link_type(keep);
F2.t0        = F.t0(keep);
F2.cap       = F.cap(keep);
F2.b         = F.b(keep);
F2.m         = F.m(keep);
end

function c = eval_link_costs(F, X)

X = X(:);
m = numel(X);
c = zeros(m, 1);

for e = 1:m
    t = lower(F.link_type{e});
    switch t
        case 'bpr'
            t0    = F.t0(e);
            cap   = F.cap(e);
            alpha = F.alpha;
            beta  = F.beta;
            if isnan(cap) || cap <= 0, cap = inf; end
            c(e) = t0 * (1 + alpha * ((X(e)/cap)^beta));
        case {'linear', 'affine'}
            b    = F.b(e);
            mlin = F.m(e);
            if isnan(b),    b    = 0; end
            if isnan(mlin), mlin = 0; end
            c(e) = b + mlin * X(e);
        otherwise
            error('Unknown link cost type: %s (edge %d)', t, e);
    end
end

c = c.';
end


% ============================================================
% ASSIGNMENT (DUE / SUE)
% ============================================================

function [x, X, c, C] = calc_DUE(F, B, Q, tol, max_iter, ~, restart)

if nargin < 7 || isempty(restart), restart = restart_default(); end
P = size(B, 1);

if isfield(restart, 'x0') && ~isempty(restart.x0)
    x = restart.x0;
else
    x = zeros(1, P);
end

% Initialize X with initial path flows
X = x * B;

rs.restart_k   = 200;
rs.restart_tol = 1e-6;
rs.restart_max = 5;

best_improve    = Inf;
last_improve_it = 0;
local_it        = 0;
num_restart     = 0;

for it = 1:max_iter
    local_it = local_it + 1;

```

```
716     c = eval_link_costs(F, X);
717     C = (B * c.').';
718     Cmin = min(C);
719     I = find(C <= Cmin + 1e-12);
720     y = zeros(1, P);
721     y(I) = Q / numel(I);
722     Y = y * B;
723 
724     a     = 1 / local_it;
725     x_new = x + a*(y - x);
726     X_new = x_new * B;
727 
728     rel = norm(X_new - X, 1) / max(Q, 1);
729 
730     if mod(it, 1000) == 0
731         fprintf('    [DUE] it = %6d, rel = %.3e\n', it, rel);
732     end
733 
734     if rel < tol
735         x = x_new;
736         X = X_new;
737         fprintf('    [DUE] converged at it = %d, rel = %.3e\n', it, rel);
738         break;
739     end
740 
741     if rel + rs.restart_tol < best_improve
742         best_improve    = rel;
743         last_improve_it = it;
744     elseif (it - last_improve_it) >= rs.restart_k && num_restart < rs.restart_max
745         fprintf('    [DUE] restart #%d at it = %d (rel = %.3e)\n', ...
746                 num_restart+1, it, rel);
747         x = y;
748         X = Y;
749         best_improve    = Inf;
750         last_improve_it = it;
751         local_it        = 0;
752         num_restart     = num_restart + 1;
753         continue;
754     end
755 
756     x = x_new;
757     X = X_new;
758 end
759 
760 fprintf('    [DUE Final] x size=%s, X size=%s, sum(X)=%.6g\n', ...
761         mat2str(size(x)), mat2str(size(X)), sum(X));
762 
763 c = eval_link_costs(F, X);
764 C = (B * c.').';
765 
766 % Ensure X is not empty
767 if isempty(X)
768     fprintf('WARNING: X is empty at end of calc_DUE!\n');
769     X = zeros(1, size(B, 2));
770 end
771 end
772 
773 
774 function [x, X, c, C] = calc_SUE(F, B, Q, tol, max_iter, ~, theta, restart)
775 
776 if nargin < 8 || isempty(theta),  theta  = 0.5; end
777 if nargin < 9 || isempty(restart), restart = restart_default(); end
778 P = size(B, 1);
779 
780 if isfield(restart, 'x0') && ~isempty(restart.x0)
781     x = restart.x0;
782 else
783     x = zeros(1, P);
784 end
```

```
785
786 % Initialize X with initial path flows
787 X = x * B;
788
789 rs.restart_k   = 200;
790 rs.restart_tol = 1e-6;
791 rs.restart_max = 5;
792
793 best_improve    = Inf;
794 last_improve_it = 0;
795 local_it        = 0;
796 num_restart     = 0;
797
798 for it = 1:max_iter
799     local_it = local_it + 1;
800
801     c = eval_link_costs(F, X);
802     C = (B * c.').';
803     z = -C / max(theta, 1e-12);
804     z = z - max(z);
805     p = exp(z);
806     p = p / sum(p);
807
808     y     = Q * p;
809     Y     = y * B;
810     a     = 1 / local_it;
811     x_new = x + a*(y - x);
812     X_new = x_new * B;
813
814     rel = norm(X_new - X, 1) / max(Q, 1);
815
816     if mod(it, 1000) == 0
817         fprintf('    [SUE] it = %6d, rel = %.3e\n', it, rel);
818     end
819
820     if rel < tol
821         x = x_new;
822         X = X_new;
823         fprintf('    [SUE] converged at it = %d, rel = %.3e\n', it, rel);
824         break;
825     end
826
827     if rel + rs.restart_tol < best_improve
828         best_improve    = rel;
829         last_improve_it = it;
830     elseif (it - last_improve_it) >= rs.restart_k && num_restart < rs.restart_max
831         fprintf('    [SUE] restart #%d at it = %d (rel = %.3e)\n', ...
832                 num_restart+1, it, rel);
833         x = y;
834         X = Y;
835         best_improve    = Inf;
836         last_improve_it = it;
837         local_it        = 0;
838         num_restart     = num_restart + 1;
839         continue;
840     end
841
842     x = x_new;
843     X = X_new;
844 end
845
846 fprintf('    [SUE Final] x size=%s, X size=%s, sum(X)=%.6g\n', ...
847         mat2str(size(x)), mat2str(size(X)), sum(X));
848
849 c = eval_link_costs(F, X);
850 C = (B * c.').';
851
852 % Ensure X is not empty
853 if isempty(X)
```

```
    fprintf('WARNING: X is empty at end of calc_SUE!\n');
    X = zeros(1, size(B, 2));
end
end


function r = restart_default()
r = struct('x0', []);
end


% ========================================================
% PATH ENUMERATION (FIXED VERSION)
% ========================================================

function B = incidence_from_all_paths_DAG(links, s, t, PMAX)
% Enumerate all simple s->t paths and return the pathlink incidence matrix B.
% Uses per-branch 'visited' to avoid pruning valid paths in shared-node networks.

if nargin < 4, PMAX = 20000; end

nodes = unique(links(:));
N     = max(nodes);

% Adjacency list
adj = cell(N, 1);
for k = 1:size(links, 1)
    u = links(k, 1);
    v = links(k, 2);
    adj{u} = [adj{u}, v];
end

paths = {};
cur   = zeros(1, PMAX);

    function dfs(u, depth, visited)
        if depth > PMAX
            return;
        end
        if u == t
            paths{end+1} = cur(1:depth-1);
            return;
        end

        for v = adj{u}
            if visited(v)
                continue;
            end
            cur(depth) = v;
            visited_local    = visited;
            visited_local(v) = true;
            dfs(v, depth+1, visited_local);
        end
    end

visited0 = false(1, N);
if s <= N
    visited0(s) = true;
end
cur(1) = s;
dfs(s, 2, visited0);

% Build incidence matrix B from collected paths
m = size(links, 1);
P = numel(paths);
B = zeros(P, m);

for p = 1:P
    seq = [s paths{p}];
```

```
    for i = 1:(numel(seq)-1)
        u   = seq(i);
        v   = seq(i+1);
        idx = find(links(:,1)==u & links(:,2)==v, 1, 'first');
        if ~isempty(idx)
            B(p, idx) = 1;
        end
    end
end
end


% ========================================================
% UTILITIES
% ========================================================

function added_idx = detect_added_edges(linksB, linksA)

mA        = size(linksA, 1);
added_idx = nan(mA, 1);

for k = 1:mA
    if ~ismember(linksA(k, :), linksB, 'rows')
        added_idx(k) = k;
    end
end
end

function v = getdef(S, name, default)
if isfield(S, name)
    v = S.(name);
else
    v = default;
end
end

function v = expand_vec(S, name, L, defaultVal)
if isfield(S, name)
    v = S.(name);
    if isscalar(v), v = repmat(v, L, 1); end
else
    v = defaultVal * ones(L, 1);
end

v = v(:);
if numel(v) < L
    v(end+1:L, 1) = defaultVal;
end
end

function C = expand_cell(S, name, L, defaultVal)
if isfield(S, name)
    C = S.(name);
    if ischar(C), C = {C}; end
else
    C = cell(0, 1);
end

if numel(C) < L
    C(end+1:L) = {defaultVal};
end
C = C(:);
end

function s = vecfmt(v, maxN)
% Format vector v as a string for display
% Handle empty vectors gracefully

if nargin < 2, maxN = 120; end
```

```
992
993 if isempty(v)
994     s = '[]';
995     return;
996 end
997
998 v   = v(:).';
999 txt = sprintf(' %.6g', v);
1000 txt = ['[' txt(2:end) ']'];
1001
1002 if numel(txt) > maxN
1003     K    = max(1, floor((maxN-10)/12));
1004     head = sprintf(' %.6g', v(1:min(K, numel(v))));
1005     tail = sprintf(' %.6g', v(max(1, numel(v)-K+1):end));
1006     txt  = ['[' head '  .. .' tail ' ]'];
1007 end
1008
1009 s = txt;
1010 end
1011
1012 function print_link_legend(links)
1013 for e = 1:size(links, 1)
1014     fprintf('%2d: %d->%d\n', e, links(e, 1), links(e, 2));
1015 end
1016 end
1017
1018
1019 % ========================================================
1020 % NETWORK VISUALISATION
1021 % ========================================================
1022
1023 function plot_networks_for_debug(linksB, linksA, added_idx, s, t)
1024
1025 figure('Name', 'Goal2 networks (BEFORE vs AFTER)', 'NumberTitle', 'off');
1026
1027 subplot(1, 2, 1);
1028 hB   = plot_one_network(linksB, s, t, [], 'BEFORE network');
1029 posX = hB.XData;
1030 posY = hB.YData;
1031
1032 subplot(1, 2, 2);
1033 isAdd = ~isnan(added_idx);
1034 plot_one_network(linksA, s, t, isAdd, 'AFTER network', posX, posY);
1035 end
1036
1037
1038 function h = plot_one_network(links, s, t, isAdded, ttl, posX, posY)
1039
1040 G = digraph(links(:, 1), links(:, 2));
1041
1042 if nargin >= 6 && ~isempty(posX) && ~isempty(posY)
1043     h = plot(G, 'XData', posX, 'YData', posY, 'ArrowSize', 8);
1044 else
1045     h = plot(G, 'Layout', 'layered', 'ArrowSize', 8);
1046 end
1047
1048 title(ttl, 'Interpreter', 'none');
1049
1050 nodeIDs    = 1:numnodes(G);
1051 nodeLabels = arrayfun(@num2str, nodeIDs, 'UniformOutput', false);
1052 labelnode(h, nodeIDs, nodeLabels);
1053
1054 if nargin >= 3 && ~isempty(s) && ~isempty(t)
1055     if s <= numnodes(G)
1056         highlight(h, s, 'Marker', 'o', 'MarkerSize', 8);
1057     end
1058     if t <= numnodes(G)
1059         highlight(h, t, 'Marker', 's', 'MarkerSize', 8);
1060     end
```

```
end

if nargin >= 4 && ~isempty(isAdded)
    idxEdges = find(isAdded);
    if ~isempty(idxEdges)
        u = links(idxEdges, 1);
        v = links(idxEdges, 2);
        highlight(h, u, v, 'EdgeColor', 'r', 'LineWidth', 1.5);
    end
end
end


% ========================================================
% VISUALIZATION (ATT vs Q) + MULTI-RUN
% ========================================================

function [ScanList, IntervalList, SummaryList, labels] = goal2_run_multi(CFG_list, labels)
% Run multiple CFGs and collect Scan outputs for joint visualization.

if nargin < 1 || isempty(CFG_list)
    error('goal2_run_multi: CFG_list is empty.');
end
if ~iscell(CFG_list)
    CFG_list = {CFG_list};
end

K = numel(CFG_list);

% Auto labels if not provided
if nargin < 2 || isempty(labels)
    labels = cell(K,1);
    for i = 1:K
        labels{i} = cfg2label(CFG_list{i}, i);
    end
else
    if ischar(labels) || isstring(labels), labels = {char(labels)}; end
    labels = labels(:);
    if numel(labels) < K
        for i = numel(labels)+1:K
            labels{i} = cfg2label(CFG_list{i}, i);
        end
    end
end

ScanList     = cell(K,1);
IntervalList = cell(K,1);
SummaryList  = cell(K,1);

for i = 1:K
    fprintf('\n\n########################################\n');
    fprintf('[MULTI] Running case %d / %d: %s\n', i, K, labels{i});
    fprintf('########################################\n');
    [ScanList{i}, IntervalList{i}, SummaryList{i}] = goal2_run(CFG_list{i});
end
end

function s = cfg2label(CFG, idx)
% Compact label used in legend/title.
net  = getdef(CFG,'net', struct());
asg  = getdef(CFG,'assign', struct());

kind = upper(getdef(net,'kind','?'));
ak   = upper(getdef(asg,'kind','?'));

if strcmpi(ak,'SUE')
    th = getdef(asg,'theta', NaN);
    s  = sprintf('Case %d | %s-%s(\\theta=%.3g)', idx, kind, ak, th);
else
```

```
    s  = sprintf('Case %d | %s-%s', idx, kind, ak);
end

if isfield(net,'kind') && strcmpi(net.kind,'ladder')
    s = sprintf('%s | n=%d', s, getdef(net,'n', NaN));
end
end

function hFig = plot_deltaATT_vs_Q_multi(ScanList, labels, viz)
% Overlay ATT-Q curves from multiple Scan tables into one figure.

if nargin < 1 || isempty(ScanList)
    error('plot_deltaATT_vs_Q_multi: ScanList empty.');
end
if istable(ScanList), ScanList = {ScanList}; end

if nargin < 2 || isempty(labels)
    labels = arrayfun(@(k) sprintf('Case %d',k), 1:numel(ScanList), 'UniformOutput', false);
elseif ischar(labels) || isstring(labels)
    labels = {char(labels)};
end

if nargin < 3 || isempty(viz), viz = struct(); end
showMk = getdef(viz,'showParadoxMarkers', true);
legLoc = getdef(viz,'legendLocation',     'best');
lw     = getdef(viz,'lineWidth',          1.8);
ms     = getdef(viz,'markerSize',         6);

hFig = figure('Name','ATT vs Q (multi-config)','NumberTitle','off');
hold on; grid on; box on;

h0 = yline(0, '-', 'LineWidth', 1.0);
set(h0, 'HandleVisibility', 'off');

for i = 1:numel(ScanList)
    S = ScanList{i};
    need = {'Q','Delta_AvgTT'};
    if ~all(ismember(need, S.Properties.VariableNames))
        error('Scan table must contain columns: Q, Delta_AvgTT');
    end

    Q  = double(S.Q(:));
    dA = double(S.Delta_AvgTT(:));

    plot(Q, dA, 'LineWidth', lw, 'DisplayName', labels{min(i,numel(labels))});

    if showMk && ismember('IsParadox', S.Properties.VariableNames)
        ip = logical(S.IsParadox(:));
        if any(ip)
            plot(Q(ip), dA(ip), 'o', 'MarkerSize', ms, 'HandleVisibility','off');
        end
    end
end

title('Paradox indicator (\Delta ATT) across configurations', 'Interpreter','tex');
xlabel('Total demand Q', 'Interpreter','none');
ylabel('\Delta ATT = ATT_{after} - ATT_{before}', 'Interpreter','tex');
legend('Location', legLoc);

hold off;
end


% ============================================================
% PRESET (Optional)
% ============================================================

function CFG = preset_braess_classic(CFG)
CFG.net.kind       = 'diamond';
```

```
CFG.net.dir        = 'oneway';
CFG.cost.link_type = {'linear', 'linear', 'linear', 'linear', 'linear'};
CFG.cost.b         = [0, 45, 45, 0, 0];
CFG.cost.m         = [1,  0,  0,  1, 0];
CFG.cost.t0        = [NaN NaN NaN NaN NaN];
CFG.cost.cap       = [NaN NaN NaN NaN NaN];
CFG.opts.paradox_eps = 1e-6;
CFG.opts.zero_tol    = 1e-8;
CFG.opts.tol         = 1e-8;
CFG.opts.max_iter    = 5e4;
CFG.Q               = 1:1:20;
end
```

B.3. Script 2: `braessfigure.m`

B.3.1. What this script computes

The script `braessfigure.m` implements a reproducibility-oriented multi-case simulation of the classical Braess diamond network under deterministic user equilibrium (DUE). For each cost-function case and each demand level Q in a specified scan grid, it computes equilibria for:

- the *before* network (4 links, 2 r–s paths), and
- the *after* network (5 links, 3 r–s paths, with the shortcut link).

For both networks it outputs average travel time (ATT), total system travel time (TSTT), link-flow vectors, and convergence diagnostics. A paradox flag is raised only when both equilibria converge and ΔATT exceeds a user-specified numerical threshold.

B.3.2. Numerical method (solver and tie-handling)

Equilibria are computed by a path-based Frank–Wolfe procedure applied to the Beckmann potential, with a line search over the convex combination step. Let x be the path-flow vector and $X = Bx$ the implied link flows (with incidence matrix B). Link costs are evaluated as $C_i = F_i(X_i)$, and path costs are computed by $c = C^\top B$. At each iteration, an all-or-nothing auxiliary solution y is constructed by assigning the entire demand to the minimum-cost path set. To avoid numerical ambiguity under ties, `braessfigure.m` uses a near-shortest tolerance:

$$\texttt{tie_tol} = \texttt{tie_rel_tol} \cdot \max(1, |c_{\min}|),$$

and splits the demand equally among all paths satisfying $c - c_{\min} \leq \texttt{tie_tol}$. Convergence is assessed via a relative gap metric, and the run is declared converged if the gap is below `gap\_tol`.

B.3.3. User-editable parameters (what to edit)

All user-editable settings are located at the top of the script:

- **Solver controls:** `max_iter`, `gap_tol`, `tie_rel_tol`.

- **Paradox threshold:** `paradox_tol` (ATT threshold used to avoid numerical noise).
- **Demand scan:** `minQ`, `maxQ`, and `Qrange=minQ:maxQ`.
- **Plot/export:** `doPlot`, `savePlots`, and `plotFormat` (`'png'` or `'pdf'`).
- **Cost-function test cases:** `test_cases` (cell array of function handles) and `test_names`.

The canonical path–link incidence matrices are hard-coded as: `B1` (4 links, 2 paths) for the before network and `B2` (5 links, 3 paths) for the after network. These should be modified only if the path set is changed.

B.3.4. Outputs (what the script generates)

Tabular output. The script assembles a numeric table `ResultsTable` and writes it to the folder containing the script:

- `braess_scan_results.xlsx` (preferred),
- `braess_scan_results.csv` (fallback if Excel writing fails).

The exported columns include (case identifier/name, demand Q, ATT before/after, ΔATT, paradox flag, TSTT before/after, link flows for both networks, relative gaps, iteration counts, and convergence flags).

Figures. If `doPlot=true`, the script generates:

- an overview figure aggregating all cases and reserving a dedicated right legend panel: `fig_overview_allcases.<plotFormat>`,
- one per-case dashboard figure (2×2 layout) exported as: `fig_case\%02d_dashboard.<plotFormat>`.

B.3.5. Current configuration in this submission (default values)

The current defaults encoded in `braessfigure.m` are:

- **Demand scan:** `minQ=1`, `maxQ=15`, `Qrange=1:15`.
- **Solver:** `max_iter=4000`, `gap_tol=1e-10`, `tie_rel_tol=1e-9`.
- **Paradox threshold:** `paradox_tol=1e-6`.
- **Export/plot:** `doPlot=true`, `savePlots=true`, `plotFormat='png'`.
- **Cases:** seven cases with names listed in `test_names` (Case 0–6).

B.3.6. Source code listing

Listing B.2: Script 2: Frank–Wolfe DUE scan + export + figures (`braessfigure.m`)

```
%%%% Braess Paradox Multi-Case Simulation
```

```
%%%% FrankWolfe DUE solver (Beckmann) + robust tie-handling
%%%% Excel numeric output (all link flows) to script folder
%%%% FIX (2026-02-09):
%%%%   - Overview figure uses MANUAL axes layout with a dedicated right legend panel:
%%%%       (1) Figure canvas enlarged
%%%%       (2) Right column plots shifted left (space reserved for legend)
%%%%       (3) Legend font/token size reduced to avoid truncation
%%%%       (4) Bottom plots (ATT, TSTT) allocate more space above 0

clear; clc;

% ========================
% Controls
% ========================
max_iter    = 4000;        % FW usually converges fast on this tiny network
gap_tol     = 1e-10;       % relative gap tolerance
tie_rel_tol = 1e-9;        % near-shortest-path tolerance (relative)
paradox_tol = 1e-6;        % ATT threshold for paradox flag (avoid numerical noise)

minQ = 1; maxQ = 15;
Qrange = minQ:maxQ;

% ========================
% Visualisation
% ========================
% doPlot:     show figures on screen
% savePlots:  export figures to the same folder as this script
% plotFormat: 'png' (quick review) or 'pdf' (LaTeX-ready)
doPlot     = true;
savePlots  = true;
plotFormat = 'png';

% ========================
% Cost function test cases
% ========================
test_cases = {
    { % Case 0: Braess et al. (2005) reference
        {@(x) 10*x, @(x) 50 + x, @(x) 10*x, @(x) 50 + x, @(x) 10 + x}, ...
        {@(x) 10*x, @(x) 50 + x, @(x) 10*x, @(x) 50 + x}
    },
    { % Case 1: linear + constant
        {@(x) 10*x, @(x) 100, @(x) 10*x, @(x) 100, @(x) 10*x}, ...
        {@(x) 10*x, @(x) 100, @(x) 10*x, @(x) 100}
    },
    { % Case 2: quadratic + constant
        {@(x) 10*(x.^2), @(x) 100, @(x) 10*(x.^2), @(x) 100, @(x) 10*x}, ...
        {@(x) 10*(x.^2), @(x) 100, @(x) 10*(x.^2), @(x) 100}
    },
    { % Case 3: all quadratic
        {@(x) 10*(x.^2), @(x) 10*(x.^2), @(x) 10*(x.^2), @(x) 10*(x.^2), @(x) 10*(x.^2)}, ...
        {@(x) 10*(x.^2), @(x) 10*(x.^2), @(x) 10*(x.^2), @(x) 10*(x.^2)}
    },
    { % Case 4: linear + quadratic mix
        {@(x) 10*x, @(x) 10*(x.^2), @(x) 10*x, @(x) 10*(x.^2), @(x) 100}, ...
        {@(x) 10*x, @(x) 10*(x.^2), @(x) 10*x, @(x) 10*(x.^2)}
    },
    { % Case 5: constant + linear + quadratic
        {@(x) 100, @(x) 10*x, @(x) 10*(x.^2), @(x) 100, @(x) 10*x}, ...
        {@(x) 100, @(x) 10*x, @(x) 10*(x.^2), @(x) 100}
    },
    { % Case 6: constant-shortcut + quadratic
        {@(x) 10*(x.^2), @(x) 10*(x.^2), @(x) 100, @(x) 10*(x.^2), @(x) 10*(x.^2)}, ...
        {@(x) 10*(x.^2), @(x) 10*(x.^2), @(x) 10*(x.^2), @(x) 10*(x.^2)}
    }
};

test_names = {
    'Case 0: Braess(2005) reference',
    'Case 1: linear+const',
```

```
    'Case 2: quadratic+const',
    'Case 3: all quadratic',
    'Case 4: linear+quadratic mix',
    'Case 5: constant+linear+quadratic',
    'Case 6: constant-shortcut + quadratic'
};

colors = {'k','r','g','b','c','m',[0 0.5 0.5]};

% =======================
% Network incidence matrices
% =======================
B2 = [1 0 1;   % u1
      0 1 0;   % u2
      0 1 1;   % u3
      1 0 0;   % u4
      0 0 1];  % u5

B1 = [1 0;     % u1
      0 1;     % u2
      0 1;     % u3
      1 0];    % u4

% =======================
% Allocate Excel result storage
% =======================
nCases = length(test_cases);
nQ = length(Qrange);
nRows = nCases * nQ;

CaseID   = zeros(nRows,1);
CaseName = strings(nRows,1);
Qcol     = zeros(nRows,1);

ATT_before  = zeros(nRows,1);
ATT_after   = zeros(nRows,1);
Delta_ATT   = zeros(nRows,1);
IsParadox   = false(nRows,1);

TSTT_before = zeros(nRows,1);
TSTT_after  = zeros(nRows,1);

u1_b = zeros(nRows,1); u2_b = zeros(nRows,1); u3_b = zeros(nRows,1); u4_b = zeros(nRows,1);
u1_a = zeros(nRows,1); u2_a = zeros(nRows,1); u3_a = zeros(nRows,1); u4_a = zeros(nRows,1); u5_a = zeros(
    nRows,1);

rel_gap_before = zeros(nRows,1);
rel_gap_after  = zeros(nRows,1);
iters_before   = zeros(nRows,1);
iters_after    = zeros(nRows,1);

Converged_before = false(nRows,1);
Converged_after  = false(nRows,1);

row = 0;

% =======================
% In-memory matrices (for plotting)
% =======================
ATTb_mat      = nan(nCases, nQ);
ATTa_mat      = nan(nCases, nQ);
DeltaATT_mat  = nan(nCases, nQ);
TSTTb_mat     = nan(nCases, nQ);
TSTTa_mat     = nan(nCases, nQ);
DeltaTSTT_mat = nan(nCases, nQ);

FlowB_mat     = nan(nCases, nQ, 4);
FlowA_mat     = nan(nCases, nQ, 5);
Paradox_mat   = false(nCases, nQ);
```

```
ConvB_mat     = false(nCases, nQ);
ConvA_mat     = false(nCases, nQ);

fprintf('\n=== Summary ===\n');
fprintf('DUE Solver: FrankWolfe (Beckmann) with line search\n');
fprintf('gap_tol=%.1e, max_iter=%d, tie_rel_tol=%.1e\n\n', gap_tol, max_iter, tie_rel_tol);

for c = 1:nCases
    F2 = test_cases{c}{1};
    F1 = test_cases{c}{2};

    x0_5 = [];
    x0_4 = [];

    fprintf('--- %s ---\n', test_names{c});
    fprintf('%4s %12s %12s %12s %10s %10s %10s\n', ...
        'Q','ATT_before','ATT_after','Delta_ATT','Paradox','Conv_b','Conv_a');
    fprintf('%4s %12s %12s %12s %10s %10s %10s\n', ...
        '----','------------','------------','------------','--------','--------','--------');

    for qi = 1:nQ
        Q = Qrange(qi);

        [x2, X2, C2, relgap2, it2] = calc_DUE_FW(F2, B2, Q, gap_tol, max_iter, tie_rel_tol, x0_5);
        T5 = sum(C2(:) .* X2(:));
        A5 = T5 / Q;
        x0_5 = x2;

        [x1, X1, C1, relgap1, it1] = calc_DUE_FW(F1, B1, Q, gap_tol, max_iter, tie_rel_tol, x0_4);
        T4 = sum(C1(:) .* X1(:));
        A4 = T4 / Q;
        x0_4 = x1;

        dA = A5 - A4;

        conv_b = (relgap1 < gap_tol);
        conv_a = (relgap2 < gap_tol);

        paradoxFlag = (conv_b && conv_a && (dA > paradox_tol));

        fprintf('%4d %12.4f %12.4f %12.4f %10s %10s %10s\n', ...
            Q, A4, A5, dA, string(paradoxFlag), string(conv_b), string(conv_a));

        row = row + 1;

        CaseID(row)   = c-1;
        CaseName(row) = string(test_names{c});
        Qcol(row)     = Q;

        ATT_before(row) = A4;
        ATT_after(row)  = A5;
        Delta_ATT(row)  = dA;
        IsParadox(row)  = paradoxFlag;

        TSTT_before(row) = T4;
        TSTT_after(row)  = T5;

        u1_b(row)=X1(1); u2_b(row)=X1(2); u3_b(row)=X1(3); u4_b(row)=X1(4);
        u1_a(row)=X2(1); u2_a(row)=X2(2); u3_a(row)=X2(3); u4_a(row)=X2(4); u5_a(row)=X2(5);

        rel_gap_before(row) = relgap1;
        rel_gap_after(row)  = relgap2;
        iters_before(row)   = it1;
        iters_after(row)    = it2;

        Converged_before(row) = conv_b;
        Converged_after(row)  = conv_a;

        ATTb_mat(c,qi)      = A4;
```

```
        ATTa_mat(c,qi)      = A5;
        DeltaATT_mat(c,qi)  = dA;
        TSTTb_mat(c,qi)     = T4;
        TSTTa_mat(c,qi)     = T5;
        DeltaTSTT_mat(c,qi) = (T5 - T4);
        FlowB_mat(c,qi,:)   = X1(:);
        FlowA_mat(c,qi,:)   = X2(:);
        Paradox_mat(c,qi)   = paradoxFlag;
        ConvB_mat(c,qi)     = conv_b;
        ConvA_mat(c,qi)     = conv_a;
    end
    fprintf('\n');
end

% ======================
% Write Excel (numeric columns only)
% ======================
ResultsTable = table( ...
    CaseID, CaseName, Qcol, ...
    ATT_before, ATT_after, Delta_ATT, IsParadox, ...
    TSTT_before, TSTT_after, ...
    u1_b,u2_b,u3_b,u4_b, ...
    u1_a,u2_a,u3_a,u4_a,u5_a, ...
    rel_gap_before, rel_gap_after, ...
    iters_before, iters_after, ...
    Converged_before, Converged_after);

thisFile = mfilename('fullpath');
if isempty(thisFile)
    outDir = pwd;
else
    outDir = fileparts(thisFile);
end

outXlsx = fullfile(outDir, 'braess_scan_results.xlsx');

try
    writetable(ResultsTable, outXlsx, 'FileType', 'spreadsheet');
    fprintf('[Saved] Excel results: %s\n', outXlsx);
catch ME
    warning('Failed to write Excel via writetable: %s\nTrying CSV fallback...', ME.message);
    outCsv = fullfile(outDir, 'braess_scan_results.csv');
    writetable(ResultsTable, outCsv);
    fprintf('[Saved] CSV results: %s\n', outCsv);
end

% ======================
% Visualise results
% ======================
if doPlot
    plot_braess_overview(outDir, Qrange, test_names, colors, ...
        ATTb_mat, ATTa_mat, DeltaATT_mat, DeltaTSTT_mat, plotFormat, savePlots);

    for c = 1:nCases
        plot_braess_case(outDir, Qrange, test_names{c}, c, colors, ...
            ATTb_mat(c,:), ATTa_mat(c,:), DeltaATT_mat(c,:), DeltaTSTT_mat(c,:), ...
            squeeze(FlowB_mat(c,:,:)), squeeze(FlowA_mat(c,:,:)), Paradox_mat(c,:), ...
            plotFormat, savePlots);
    end
end


% ======================================================
% FrankWolfe DUE solver (path-based) with line search
% ======================================================
function [x, X, C, rel_gap, it_used] = calc_DUE_FW(F, B, Q, gap_tol, max_iter, tie_rel_tol, x_init)
    sB = size(B,2);
    sA = size(B,1);

```

```
    if isempty(x_init)
        x = repmat(Q/sB, 1, sB);
    else
        x = x_init(:).';
        x = x * (Q / sum(x));
    end

    rel_gap = inf;
    it_used = 0;

    while it_used < max_iter
        it_used = it_used + 1;

        X = B * x(:);
        C = zeros(sA,1);
        for i = 1:sA
            C(i) = F{i}(X(i));
        end

        c = (C.' * B);
        cmin = min(c);

        tie_tol = tie_rel_tol * max(1, abs(cmin));
        idx = find((c - cmin) <= tie_tol);

        y = zeros(1,sB);
        y(idx) = Q / numel(idx);

        gap_num = sum( x .* (c - cmin) );
        gap_den = sum( x .* c );
        rel_gap = gap_num / max(1e-12, gap_den);

        if rel_gap < gap_tol
            return;
        end

        d = (y - x);

        obj_raw = @(a) beckmann_potential_trapz(F, B, x + a*d);
        obj = @(a) safe_scalar(obj_raw(a));
        aopt = fminbnd(obj, 0, 1);

        x = x + aopt * d;

        x(x < 0) = 0;
        s = sum(x);
        if s > 0
            x = x * (Q / s);
        else
            x = repmat(Q/sB, 1, sB);
        end
    end

    X = B * x(:);
    C = zeros(sA,1);
    for i = 1:sA
        C(i) = F{i}(X(i));
    end
end

function v = safe_scalar(v)
    if ~isscalar(v) || ~isfinite(v)
        v = Inf;
    end
end

function Z = beckmann_potential_trapz(F, B, x)
    X = B * x(:);
    sA = length(X);
```

```
    Z = 0;

    for i = 1:sA
        Xi = max(0, X(i));
        if Xi <= 0
            continue;
        end

        n = max(80, min(4000, ceil(40*Xi)));
        s = linspace(0, Xi, n);

        ti = F{i}(s);

        if isscalar(ti)
            ti = repmat(ti, size(s));
        end

        s  = s(:);
        ti = ti(:);

        if any(~isfinite(ti))
            Z = Inf;
            return;
        end

        Z = Z + trapz(s, ti);
    end

    if ~isscalar(Z) || ~isfinite(Z)
        Z = Inf;
    end
end


% ============================================================
% Plotting helpers
% ============================================================
function plot_braess_overview(outDir, Qrange, test_names, colors, ...
    ATTb_mat, ATTa_mat, DeltaATT_mat, DeltaTSTT_mat, plotFormat, savePlots)
% Robust overview layout:
% - Manually reserve a right legend panel (no truncation / no overlap)
% - Enlarge canvas
% - Shift right-column plots left (because legend panel is reserved)
% - Shrink legend (FontSize + ItemTokenSize)
% - Give more y headroom above 0 for bottom plots

    fig = figure('Color','w','Name','Braess overview: all cases');

    % ---- enlarge canvas (width + a bit height) ----
    if isprop(fig,'Position')
        p = fig.Position;
        p(3) = round(p(3) * 1.90);  % wider
        p(4) = round(p(4) * 1.10);  % slightly taller
        fig.Position = p;
    end

    % ---- Manual layout parameters (normalized) ----
    L = 0.07;   R = 0.02;
    B = 0.10;   T = 0.12;
    gapCol   = 0.06;
    gapRow   = 0.10;
    gapLegend = 0.02;
    legendW  = 0.28;      % reserved legend panel width

    colW = (1 - L - R - legendW - gapLegend - gapCol) / 2;
    rowH = (1 - T - B - gapRow) / 2;

    x1 = L;
```

```
414     x2 = L + colW + gapCol;                % right plots are naturally "left-shifted" by reserved legend
         panel
415     yTop = B + rowH + gapRow;
416     yBot = B;
417
418     % axes positions
419     ax1 = axes('Parent', fig, 'Position', [x1 yTop colW rowH]);
420     ax2 = axes('Parent', fig, 'Position', [x2 yTop colW rowH]);
421     ax3 = axes('Parent', fig, 'Position', [x1 yBot colW rowH]);
422     ax4 = axes('Parent', fig, 'Position', [x2 yBot colW rowH]);
423
424     sgtitle(fig, 'Multi-case comparison (all cases)', 'Interpreter', 'none');
425
426     nCases = size(ATTb_mat,1);
427
428     % Panel 1: ATT before
429     hold(ax1,'on'); grid(ax1,'on');
430     for c = 1:nCases
431         plot(ax1, Qrange, ATTb_mat(c,:), 'LineWidth', 1.4, 'Color', colors{c});
432     end
433     xlabel(ax1,'Total demand Q');
434     ylabel(ax1,'ATT (before, 4-link)');
435     title(ax1,'ATT before');
436
437     % Panel 2: ATT after
438     hold(ax2,'on'); grid(ax2,'on');
439     for c = 1:nCases
440         plot(ax2, Qrange, ATTa_mat(c,:), 'LineWidth', 1.4, 'Color', colors{c});
441     end
442     xlabel(ax2,'Total demand Q');
443     ylabel(ax2,'ATT (after, 5-link)');
444     title(ax2,'ATT after');
445
446     % Panel 3: Delta ATT
447     hold(ax3,'on'); grid(ax3,'on');
448     for c = 1:nCases
449         plot(ax3, Qrange, DeltaATT_mat(c,:), 'LineWidth', 1.4, 'Color', colors{c});
450     end
451     yline(ax3,0,'-');
452     xlabel(ax3,'Total demand Q');
453     ylabel(ax3,'\Delta ATT = ATT_{after} - ATT_{before}');
454     title(ax3,'Paradox indicator (\Delta ATT)');
455
456     % Panel 4: Delta TSTT (+ legend handles)
457     hold(ax4,'on'); grid(ax4,'on');
458     hLeg = gobjects(nCases,1);
459     for c = 1:nCases
460         hLeg(c) = plot(ax4, Qrange, DeltaTSTT_mat(c,:), 'LineWidth', 1.4, 'Color', colors{c});
461     end
462     yline(ax4,0,'-');
463     xlabel(ax4,'Total demand Q');
464     ylabel(ax4,'\Delta TSTT = TSTT_{after} - TSTT_{before}');
465     title(ax4,'System cost difference (\Delta TSTT)');
466
467     xlim(ax1,[Qrange(1) Qrange(end)]);
468     xlim(ax2,[Qrange(1) Qrange(end)]);
469     xlim(ax3,[Qrange(1) Qrange(end)]);
470     xlim(ax4,[Qrange(1) Qrange(end)]);
471
472     % ---- More space above 0 for bottom plots ----
473     set_asymmetric_ylim(ax3, DeltaATT_mat(:), 0.35);
474     set_asymmetric_ylim(ax4, DeltaTSTT_mat(:), 0.25);
475
476     % ---- Legend: put it into the reserved right panel ----
477     xLeg = x2 + colW + gapLegend;
478     yLeg = yBot;
479     hLegPanel = 2*rowH + gapRow;
480     lgd = legend(ax4, hLeg, test_names, 'Interpreter','none', 'Location','none', ...
481         'Position', [xLeg yLeg legendW hLegPanel]);
```

```
482
483     % shrink legend overall (requested)
484     lgd.FontSize = max(7, lgd.FontSize - 2);
485     if isprop(lgd,'ItemTokenSize')
486         lgd.ItemTokenSize = [10 8]; % shorter line sample + tighter rows
487     end
488
489     % small safety: if still tight, reduce 1 more step
490     drawnow;
491     if isprop(lgd,'Position')
492         if lgd.Position(1) + lgd.Position(3) > 0.995
493             lgd.FontSize = max(6, lgd.FontSize - 1);
494             drawnow;
495         end
496     end
497
498     if savePlots
499         save_figure(fig, fullfile(outDir, ['fig_overview_allcases.' plotFormat]));
500     end
501 end
502
503
504 function plot_braess_case(outDir, Qrange, caseName, caseIdx, colors, ...
505     ATTb, ATTa, dATT, dTSTT, FlowB, FlowA, paradoxFlag, plotFormat, savePlots)
506     %#ok<INUSD>
507
508     fig = figure('Color','w','Name',sprintf('Case %d', caseIdx-1));
509     hasTiled = (exist('tiledlayout','file') == 2);
510     if hasTiled
511         t = tiledlayout(2,2,'TileSpacing','compact','Padding','compact');
512         title(t, sprintf('Case %d: %s', caseIdx-1, caseName), 'Interpreter', 'none');
513     else
514         sgtitle(sprintf('Case %d: %s', caseIdx-1, caseName), 'Interpreter', 'none');
515     end
516
517     col = colors{caseIdx};
518
519     if hasTiled, ax1 = nexttile; else, ax1 = subplot(2,2,1); end
520     hold(ax1,'on'); grid(ax1,'on');
521     plot(ax1, Qrange, ATTb, '-',  'LineWidth', 1.7, 'Color', col);
522     plot(ax1, Qrange, ATTa, '--', 'LineWidth', 1.7, 'Color', col);
523     xlabel(ax1, 'Total demand Q');
524     ylabel(ax1, 'Average travel time (ATT)');
525     legend(ax1, {'Before (4-link)','After (5-link)'}, 'Location', 'best');
526     title(ax1, 'ATT curves');
527
528     if hasTiled, ax2 = nexttile; else, ax2 = subplot(2,2,2); end
529     hold(ax2,'on'); grid(ax2,'on');
530     plot(ax2, Qrange, dATT, 'LineWidth', 1.7, 'Color', col);
531     yline(ax2, 0, '-');
532     hasP = any(paradoxFlag);
533     if hasP
534         scatter(ax2, Qrange(paradoxFlag), dATT(paradoxFlag), 35, 'filled');
535     end
536     xlabel(ax2, 'Total demand Q');
537     ylabel(ax2, '\Delta ATT');
538     title(ax2, 'Paradox region (\Delta ATT > 0)');
539     if hasP
540         legend(ax2, {'\Delta ATT','0-line','Paradox Q'}, 'Location', 'best');
541     else
542         legend(ax2, {'\Delta ATT','0-line'}, 'Location', 'best');
543     end
544
545     if hasTiled, ax3 = nexttile; else, ax3 = subplot(2,2,3); end
546     hold(ax3,'on'); grid(ax3,'on');
547     plot(ax3, Qrange, FlowB, 'LineWidth', 1.3);
548     xlabel(ax3, 'Total demand Q');
549     ylabel(ax3, 'Equilibrium link flow');
550     title(ax3, 'Link flows (before, u1..u4)');
```

```
    legend(ax3, {'u1','u2','u3','u4'}, 'Location', 'best');

    if hasTiled, ax4 = nexttile; else, ax4 = subplot(2,2,4); end
    hold(ax4,'on'); grid(ax4,'on');
    plot(ax4, Qrange, FlowA, 'LineWidth', 1.3);
    xlabel(ax4, 'Total demand Q');
    ylabel(ax4, 'Equilibrium link flow');
    title(ax4, 'Link flows (after, u1..u5)');
    legend(ax4, {'u1','u2','u3','u4','u5'}, 'Location', 'best');

    if savePlots
        base = sprintf('fig_case%02d_dashboard', caseIdx-1);
        save_figure(fig, fullfile(outDir, [base '.' plotFormat]));
    end
end


function set_asymmetric_ylim(ax, yData, posRatio)
    y = yData(:);
    y = y(isfinite(y));
    if isempty(y), return; end

    ymin = min(y);
    ymax = max(y);

    if ymin >= 0
        ylim(ax, [0, ymax*1.10 + eps]);
        return;
    end

    negAmp = abs(ymin);

    bottom = ymin * 1.05;
    topMin = posRatio * negAmp;
    top    = max([ymax*1.15, topMin, 0]);

    ylim(ax, [bottom, top]);
end


function save_figure(fig, filePath)
    [outFolder, base, ext] = fileparts(filePath);
    if ~isempty(outFolder) && ~exist(outFolder,'dir')
        mkdir(outFolder);
    end
    if isempty(ext)
        ext = '.png';
        filePath = [filePath ext];
        [outFolder, base, ext] = fileparts(filePath);
    end

    try
        if exist('exportgraphics','file') == 2
            exportgraphics(fig, filePath, 'Resolution', 300);
        else
            set(fig, 'PaperPositionMode', 'auto');
            if strcmpi(ext, '.pdf')
                print(fig, fullfile(outFolder, base), '-dpdf', '-r300');
            else
                print(fig, fullfile(outFolder, base), '-dpng', '-r300');
            end
        end
    catch ME
        warning('Failed to save figure (%s): %s', filePath, ME.message);
    end
end
```

B.4. Script 3: braess_batch_atlas.m (symbolic atlas generator)

B.4.1. What this script computes

The file braess_batch_atlas.m constructs a batch *symbolic* parameter atlas for the classical Braess diamond network under DUE with linear link costs. For each combination of demand D, grouped before-network parameters (a_U, b_U, a_M, b_M), and a target shift dt, it symbolically solves for the added-link parameters (a_5, b_5) on link $(2 \to 3)$ such that

$$\Delta(a_5, b_5) = dt,$$

and records the resulting expressions (including possible free parameters and validity conditions).

B.4.2. User-editable parameters (what to edit)

At the top of the file:

- Demand grid: D_list.
- Before-network grouped parameters: aU_list, bU_list, aM_list, bM_list.
- Target values: dt_list.
- Output path: out_file=fullfile(pwd,'braess_paradox_atlas.xlsx').

B.4.3. How to run (reproducibility)

A minimal reproducible sequence is:

1. set the working directory to the code folder (so that outputs are written locally), e.g. cdCode;
2. run the atlas generator, obtaining T and writing braess_paradox_atlas.xlsx to pwd.

Note: the entry function defined in the file is braess_paradox_atlas(). In MATLAB, it is recommended that the primary function name matches the file name. If a reader encounters a name-mismatch warning, they may either rename the file to braess_paradox_atlas.m or change the first line to functionT=braess_batch_atlas().

B.4.4. Outputs

The script exports:

- an Excel table braess_paradox_atlas.xlsx (written to the current working directory pwd),
- a MATLAB table T returned to the workspace.

Each row stores scenario parameters, the symbolic expression `delta\_expr`, the solved `a5\_expr`/`b5\_expr`, and any free-parameter/condition strings.

B.4.5. Current configuration in this submission (default values)

Default settings encoded in the file are: `D\_list=2:4:18`, `aU\_list=[2610]`, `aM\_list=[123]`, `bU\_list=[02040]`, `bM\_list=[405060]`, and `dt\_list=[1,-1,0]`.

B.4.6. Source code listing

Listing B.3: Script 3: Symbolic atlas generator (`braess\_batch\_atlas.m`)

```
function T = braess_paradox_atlas()
% BRAESS_PARADOX_ATLAS

% Batch symbolic atlas for Braess' paradox on the diamond network
% (linear cost, DUE), following Prof. Mussone's rough program.
%
% Assumptions:
%   - Links 1-2 and 3-4 share the same linear cost:
%           c_U(x) = aU * x + bU
%   - Links 1-3 and 2-4 share another linear cost:
%           c_M(x) = aM * x + bM
%   => BEFORE network coefficients:
%           a1 = aU, a2 = aM, a3 = aM, a4 = aU
%           b1 = bU, b2 = bM, b3 = bM, b4 = bU
%
% For each combination of:
%   - demand D
%   - (aU,bU,aM,bM)
%   - target delta dt  (delta = C_after - C_before)
%
% we solve symbolically for the added link parameters (a5,b5):
%           delta(a5,b5) = dt
%
% and collect:
%   - symbolic delta(a5,b5)
%   - symbolic a5, b5 (possibly with free parameter)
%   - parameter names and conditions
%
% Output:
%   - Excel file "braess_paradox_atlas.xlsx" in the current folder
%   - Table T returned to the workspace


% 1. USER SETTINGS: D, before parameters, dt list
% You can freely edit these lists.

% 1.1 Demand values
D_list = 2:4:18;   % e.g. three demand levels

% 1.2 BEFORE-network grouped parameters
%     aU_list : slopes for links 1-2 & 3-4
%     aM_list : slopes for links 1-3 & 2-4
%     bU_list : constants for links 1-2 & 3-4
%     bM_list : constants for links 1-3 & 2-4
aU_list = [2  6  10 ];       % e.g. two options for upper/lower links
aM_list = [1 2 3 ];          % main links (you can add more)
bU_list = [0 20 40 ];          % often 0 in textbook Braess
bM_list = [40 50 60 ];      % two options for middle links

% 1.3 Target delta values
%   dt > 0  : paradox target
%   dt <= 0 : non-paradox / boundary targets
```

```
dt_list = [1, -1, 0];   % paradox, non-paradox, and boundary

% 1.4 Output Excel path (same folder as this code / current folder)
out_file = fullfile(pwd, 'braess_paradox_atlas.xlsx');


% 2. PREALLOCATE RESULTS
results = struct('ScenarioID', {}, ...
                 'D', {}, ...
                 'aU', {}, 'bU', {}, 'aM', {}, 'bM', {}, ...
                 'a1', {}, 'a2', {}, 'a3', {}, 'a4', {}, ...
                 'b1', {}, 'b2', {}, 'b3', {}, 'b4', {}, ...
                 'dt_target', {}, ...
                 'ParadoxFlag', {}, ...   % dt>0 ?
                 'delta_expr', {}, ...    % (a5,b5)
                 'a5_expr', {}, ...       % expression for a5
                 'b5_expr', {}, ...       % expression for b5
                 'param_expr', {}, ...    % free parameter(s)
                 'cond_expr', {});        % conditions on parameters

row     = 0;
scen_id = 0;


%3. MAIN LOOPS: D  (aU,bU,aM,bM)  dt
for iD = 1:numel(D_list)
    D = D_list(iD);

    for iU = 1:numel(aU_list)
        for jU = 1:numel(bU_list)
            for iM = 1:numel(aM_list)
                for jM = 1:numel(bM_list)

                    scen_id = scen_id + 1;

                    aU = aU_list(iU);
                    bU = bU_list(jU);
                    aM = aM_list(iM);
                    bM = bM_list(jM);

                    % BEFORE coefficients on the four links
                    av14 = [aU, aM, aM, aU];
                    bv14 = [bU, bM, bM, bU];

                    fprintf('\n=== Scenario %d: D=%g, aU=%g, bU=%g, aM=%g, bM=%g ===\n', ...
                        scen_id, D, aU, bU, aM, bM);

                    for idt = 1:numel(dt_list)
                        dt = dt_list(idt);
                        fprintf('  Target dt = %g\n', dt);

                        % single-case symbolic solver (professor-style)
                        [sa5,sb5,P,delta,sol] = braess_solver_single(D,av14,bv14,dt);

                        % Convert symbolic objects to strings for Excel
                        a5_str    = char(sa5);
                        b5_str    = char(sb5);
                        delta_str = char(delta);

                        if isfield(sol,'parameters')
                            param_str = char(sol.parameters);
                        else
                            param_str = '';
                        end
                        if isfield(sol,'conditions')
                            cond_str = char(sol.conditions);
                        else
                            cond_str = '';
                        end
```

```
122
123                         % Store a row
124                         row = row + 1;
125                         results(row).ScenarioID  = scen_id;
126                         results(row).D           = D;
127
128                         results(row).aU = aU;
129                         results(row).bU = bU;
130                         results(row).aM = aM;
131                         results(row).bM = bM;
132
133                         results(row).a1 = av14(1);
134                         results(row).a2 = av14(2);
135                         results(row).a3 = av14(3);
136                         results(row).a4 = av14(4);
137
138                         results(row).b1 = bv14(1);
139                         results(row).b2 = bv14(2);
140                         results(row).b3 = bv14(3);
141                         results(row).b4 = bv14(4);
142
143                         results(row).dt_target   = dt;
144                         results(row).ParadoxFlag = (dt > 0);   % dt>0  paradox case target
145
146                         results(row).delta_expr  = delta_str;
147                         results(row).a5_expr     = a5_str;
148                         results(row).b5_expr     = b5_str;
149                         results(row).param_expr  = param_str;
150                         results(row).cond_expr   = cond_str;
151
152                         fprintf('    a5 = %s\n', a5_str);
153                         fprintf('    b5 = %s\n', b5_str);
154                         if ~isempty(cond_str)
155                             fprintf('    conditions: %s\n', cond_str);
156                         end
157                     end
158                 end
159             end
160         end
161     end
162 end
163
164
165 % 4. TABLE + EXCEL OUTPUT
166 if isempty(results)
167     warning('No results generated. Please adjust D_list / parameter lists.');
168     T = table();
169     return;
170 end
171
172 T = struct2table(results);
173 writetable(T, out_file);
174 fprintf('\nBraess paradox atlas saved to:\n  %s\n', out_file);
175
176 end  % ===== end of main function =====
177
178
179
180 %  INTERNAL FUNCTION: SINGLE-CASE SYMBOLIC SOLVER
181 %  (faithful to professor's rough code, extended with delta(a5,b5))
182
183 function [sa5,sb5,P,delta,sol] = braess_solver_single(d,av14,bv14,dt)
184 % symbolic modelling of Braess' paradox
185 % diamond network
186 % DUE
187 % linear cost functions
188 % (Mussone - Notari rough program, adapted for batch use)
189
190     if nargin < 4
```

```
        error('braess_solver_single:NotEnoughInputs', ...
              'Usage: braess_solver_single(d,av14,bv14,dt)');
    end

    clear a b c

    % --- symbolic variables (as in professor's code) ---
    syms a [1 5];   % angular coefficients of link costs
    syms b [1 5];   % constant terms of link costs

    syms D;         % demand
    syms CB [1 2];  % route costs before
    syms CA [1 3];  % route costs after

    % AFTER-network incidence matrix (diamond with link 2-3)
    % edges: 1-2, 1-3, 2-4, 3-4, 2-3
    BA = [1 0 1;    % 1-2
          0 1 0;    % 1-3
          1 0 0;    % 2-4
          0 1 1;    % 3-4
          0 0 1];   % 2-3

    % BEFORE incidence (not explicitly used later, kept for completeness)
    BB = [1 0 1;
          0 1 0;
          1 0 0;
          0 1 1];

    % BEFORE path cost (professor's hand formula) -----
    CB(1) = ( D ...
        + (b(1)+b(2))/(a(1)+a(3)) ...
        + (b(2)+b(4))/(a(2)+a(4)) ) ...
        / ( 1/(a(1)+a(3)) + 1/(a(2)+a(4)) );

    % AFTER path cost (matrix form)
    M    = BA'*diag(a)*BA;
    adjM = adjoint(M);

    CA(1) = ( D*det(M) + [1 1 1]*((adjM*BA')*transpose(b)) ) ...
          ./ ( ([1 1 1]*adjM)*[1;1;1] );

    % -substitute BEFORE parameters, keep a(5),b(5) symbolic
    z1 = subs(CA(1), [D, a(1:4), b(1:4)], [d, av14, bv14]);  % AFTER
    z2 = subs(CB(1), [D, a(1:4), b(1:4)], [d, av14, bv14]);  % BEFORE

    % rename a(5),b(5)  a5,b5 for clarity; define delta(a5,b5)
    syms a5 b5 real
    delta = simplify( subs(z1 - z2, [a(5), b(5)], [a5, b5]) );

    % paradox flag (just a marker: dt>0  paradox target)
    P = 0;
    if dt > 0
        P = 1;
    end

    % Solve delta(a5,b5) == dt for (b5,a5)
    eqn = delta == dt;
    sol = solve(eqn, [b5 a5], 'ReturnConditions', true);

    sa5 = sol.a5;
    sb5 = sol.b5;
end
```

B.5. Script 4: plot_paradox_maps.m (atlas visualisation)

B.5.1. What this script computes

The function `plot\_paradox\_maps()` visualises the symbolic atlas produced by Script B.4. It reads `braess\_paradox\_atlas.xlsx`, evaluates each symbolic `b5\_expr` into a numeric value (substituting free parameters by a fixed default), and generates relationship plots of b_5 against each single parameter while grouping by the remaining parameter tuple.

B.5.2. Inputs and dependencies

- **External file dependency:** the script requires `braess\_paradox\_atlas.xlsx` and will terminate with an error if the file is not found (it is read via `readtable`). This file is generated by Script B.4 (`braess\_batch\_atlas.m`).
- **File location requirement:** since the script changes the working directory to its own location via `cd(fileparts(mfilename('fullpath')))`, the atlas file must be located *in the same folder* as `plot\_paradox\_maps.m`.
- **Toolbox requirement:** symbolic evaluation uses `str2sym` and substitution; Symbolic Math Toolbox is required.

Minimal reproducible run: first run Script B.4 from the code folder (so that `braess\_paradox\_atlas.xlsx` is written locally), then run `plot\_paradox\_maps()` from the same folder. If a free parameter is present in `param\_expr`, the script applies default substitutions `x=1` or `z=1` for projection.

B.5.3. Outputs

Six figures (displayed on screen) are produced:

$$b_5 \text{ vs } D,\ a_U,\ b_U,\ a_M,\ b_M,\ dt.$$

Curves are rendered with alpha transparency (implemented via `surface` with `EdgeAlpha`) so that dense overlaps appear darker.

B.5.4. Source code listing

Listing B.4: Script 4: Atlas visualisation (`plot\_paradox\_maps.m`)

```
function plot_paradox_maps()
% PLOT_PARADOX_MAPS
% Visualize how b5 changes with each single parameter,
% while all the other parameters are kept fixed (per group).
%
% UPDATE (density-emphasis rendering):
% - Lines are drawn with transparency so that dense/overlapping regions
%   appear darker (more visible) and sparse regions appear lighter.
% - This is implemented via SURFACE (supports EdgeAlpha), because PLOT
%   line objects do not support alpha transparency.
```

```
11
12 clc; close all;
13 cd(fileparts(mfilename('fullpath')));  % go to script folder
14
15 % 1. Read atlas and basic columns
16 fname = 'braess_paradox_atlas.xlsx';
17 if ~isfile(fname)
18     error('File %s not found in current folder.', fname);
19 end
20
21 T  = readtable(fname);
22
23 D   = T.D;           % demand
24 aU  = T.aU;          % slope on 1-2, 3-4
25 bU  = T.bU;          % const on 1-2, 3-4
26 aM  = T.aM;          % slope on 1-3, 2-4
27 bM  = T.bM;          % const on 1-3, 2-4
28 dt  = T.dt_target;  % target delta t (from symbolic dt)
29 a5e = T.a5_expr;    %#ok<NASGU> % symbolic a5 (not used for plotting here)
30 b5e = T.b5_expr;    % symbolic b5
31 pe  = T.param_expr; % 'x','z' or ''
32
33 n = height(T);
34
35 %% 2. Evaluate b5 for x=1,z=1 (if present)
36 b5 = nan(n,1);
37 for i = 1:n
38     expr = str2sym(b5e{i});
39     p    = strtrim(pe{i});
40     try
41         if strcmp(p,'x')
42             b5(i) = double(subs(expr,'x',1));
43         elseif strcmp(p,'z')
44             b5(i) = double(subs(expr,'z',1));
45         else
46             % no free parameter
47             b5(i) = double(subs(expr));
48         end
49     catch ME
50         warning('Row %d skipped when evaluating b5: %s', i, ME.message);
51         b5(i) = NaN;
52     end
53 end
54
55 % Keep only rows with valid numeric b5
56 valid = ~isnan(b5) & isfinite(b5);
57 D  = D(valid);
58 aU = aU(valid);
59 bU = bU(valid);
60 aM = aM(valid);
61 bM = bM(valid);
62 dt = dt(valid);
63 b5 = b5(valid);
64
65 %% 3. Plots: one varying parameter, others fixed (grouped by full tuple)
66
67 % b5 vs D
68 plot_one_param(D,  [aU bU aM bM dt], b5, ...
69     'Demand D', ...
70     'Relationship between b_5 and D (other parameters fixed)');
71
72 % b5 vs aU
73 plot_one_param(aU, [D bU aM bM dt], b5, ...
74     'a_U (slope on links 1-2 and 3-4)', ...
75     'Relationship between b_5 and a_U (other parameters fixed)');
76
77 % b5 vs bU
78 plot_one_param(bU, [D aU aM bM dt], b5, ...
79     'b_U (constant on links 1-2 and 3-4)', ...
```

```
    'Relationship between b_5 and b_U (other parameters fixed)');

% b5 vs aM
plot_one_param(aM, [D aU bU bM dt], b5, ...
    'a_M (slope on links 1-3 and 2-4)', ...
    'Relationship between b_5 and a_M (other parameters fixed)');

% b5 vs bM
plot_one_param(bM, [D aU bU aM dt], b5, ...
    'b_M (constant on links 1-3 and 2-4)', ...
    'Relationship between b_5 and b_M (other parameters fixed)');

% b5 vs dt_target
plot_one_param(dt, [D aU bU aM bM], b5, ...
    '\Delta t_{target}', ...
    'Relationship between b_5 and \Delta t_{target} (other parameters fixed)');
end


function plot_one_param(x, others, y, x_label, ttl)
% PLOT_ONE_PARAM
%   x      : varying parameter (column vector)
%   others : matrix of all *other* parameters (one row per observation)
%   y      : b5 values
%   x_label: xlabel string
%   ttl    : title string
%
% For each unique row of 'others', connect the corresponding (x,y) points
% with a polyline, so that "within-group" trend is visible.
%
% DENSITY-EMPHASIS RENDERING:
% - Draw each group's polyline with alpha transparency.
% - Overlapping regions become darker/more visible by alpha accumulation.
% - Implemented with SURFACE + EdgeAlpha (PLOT lines do not support alpha).

    % Group index by full tuple of "other" parameters
    [~,~,gid] = unique(others,'rows');   % gid(i) is group ID for row i
    groups = unique(gid);
    G = numel(groups);

    % ------------------------------
    % Appearance / density controls
    % ------------------------------
    useGroupColors = false;         % false => one color to emphasize density
    baseColor      = [0 0 0];       % single-color mode (black)
    lineWidth      = 0.8;
    markerSize     = 10;            % scatter size (points)
    showMarkers    = true;

    % Auto alpha: more groups => lower per-line alpha to avoid over-darkening
    alphaLine = max(0.02, min(0.22, 1.8 / sqrt(max(G,1))));
    alphaMark = min(0.60, alphaLine * 2.0);

    % Create figure with OpenGL renderer for stable transparency
    figure('Color','w');
    set(gcf,'Renderer','opengl');
    hold on;

    if useGroupColors
        cmap = lines(G);
    else
        cmap = repmat(baseColor, G, 1);
    end

    for k = 1:G
        idx = (gid == groups(k));
        c   = cmap(k,:);

        if nnz(idx) < 2
```

```
149             % Only one point in this group: plot a single marker
150             if showMarkers
151                 hk = scatter(x(idx), y(idx), markerSize, 'o', ...
152                     'MarkerEdgeColor', c, ...
153                     'MarkerFaceColor', 'none', ...
154                     'LineWidth', 0.8);
155                 if isprop(hk,'MarkerEdgeAlpha')
156                     hk.MarkerEdgeAlpha = alphaMark;
157                 end
158             else
159                 % Fallback: draw a tiny transparent dot via surface
160                 xx = x(idx); yy = y(idx);
161                 surface([xx xx], [yy yy], [0 0; 0 0], ...
162                     'FaceColor','none', 'EdgeColor', c, ...
163                     'LineWidth', lineWidth, 'EdgeAlpha', alphaLine);
164             end
165         else
166             % Sort by x for a clean polyline
167             [xk, order] = sort(x(idx));
168             yk = y(idx);
169             yk = yk(order);
170 
171             xk = xk(:);
172             yk = yk(:);
173 
174             % Draw transparent polyline via surface (supports EdgeAlpha)
175             % Create an N-by-2 "strip" with two identical columns.
176             % FaceColor 'none' ensures only edges are drawn.
177             surface([xk xk], [yk yk], zeros(numel(xk),2), ...
178                 'FaceColor', 'none', ...
179                 'EdgeColor', c, ...
180                 'LineWidth', lineWidth, ...
181                 'EdgeAlpha', alphaLine);
182 
183             % Optional markers (also semi-transparent)
184             if showMarkers
185                 hk = scatter(xk, yk, markerSize, 'o', ...
186                     'MarkerEdgeColor', c, ...
187                     'MarkerFaceColor', 'none', ...
188                     'LineWidth', 0.8);
189                 if isprop(hk,'MarkerEdgeAlpha')
190                     hk.MarkerEdgeAlpha = alphaMark;
191                 end
192             end
193         end
194     end
195 
196     xlabel(x_label);
197     ylabel('b_5');
198     title(ttl);
199     grid on;
200     box on;
201     hold off;
202 end
```

B.6. Script 5: `sf\_equiv\_family\_link23.m` (symbolic equivalent-family generator)

B.6.1. What this script computes

The function `sf\_equiv\_family\_link23(d,m14,b14,dt,opts)` symbolically derives an equivalent family for the added link (2→3) in the Braess diamond network with linear

link costs. Given a demand d, the linear parameters of the four before links (ordered consistently with the diamond legend used in `goal2\_main`), and a target shift dt, it solves $\Delta(a_5, b_5) = dt$. If the solution contains a free parameter, it samples numeric pairs (b_5, a_5) over a user-defined grid and prints paste-ready assignments for edge 5.

B.6.2. Inputs (parameter interface)

- `d`: demand (single Q).
- `m14`: 1×4 slopes in the order `[1->2,2->4,1->3,3->4]`.
- `b14`: 1×4 intercepts in the same order.
- `dt`: target $\Delta = \mathrm{AvgTT}^{(A)} - \mathrm{AvgTT}^{(B)}$.
- `opts`: sampling/filtering options (`N\_hits`, `t\_grid`, `require\_real`, `require\_pos`, etc.).

B.6.3. Outputs

Symbolic objects `sa5`, `sb5`, `delta\_expr`, and `sol` are returned, together with numeric samples `pairs=[b5,a5]`. The function also prints paste-ready lines `CFG.cost.b(5)=...;` and `CFG.cost.m(5)=...;`.

B.6.4. Source code listing

Listing B.5: Script 5: Symbolic equivalent-family tool (`sf\_equiv\_family\_link23.m`)

```
function [sa5,sb5,P,delta_expr,sol,pairs] = sf_equiv_family_link23(d, m14, b14, dt, opts)
%SF_EQUIV_FAMILY_LINK23
%
% Symbolic-Formulation (SF) equivalent-family generator for Braess diamond.
% DUE, linear link costs. AFTER adds link 2->3 (link 5).
%
% IMPORTANT (interface aligned with goal2_main diamond ordering):
%   In goal2_main (diamond), BEFORE links are ordered as:
%     1: 1->2
%     2: 2->4
%     3: 1->3
%     4: 3->4
%   AFTER adds:
%     5: 2->3
%
% INPUTS:
%   d     : demand (Q)
%   m14   : [m1 m2 m3 m4] slopes for links [1->2, 2->4, 1->3, 3->4]
%   b14   : [b1 b2 b3 b4] intercepts for links [1->2, 2->4, 1->3, 3->4]
%   dt    : target delta = AvgTT_after - AvgTT_before   (or same metric you use)
%   opts  : optional struct
%
% OUTPUTS:
%   sa5, sb5      : symbolic solution(s) for a5 and b5 (here: slope/intercept of 2->3)
%   P             : marker (dt>0 -> 1 else 0)
%   delta_expr    : symbolic expression Delta(a5,b5)
%   sol           : solve() result (may contain parameters/conditions)
%   pairs         : numeric samples [b5, a5] (up to opts.N_hits)
%
```

```
% NOTES:
% - This file assumes linear link costs: c_i(x) = b_i + a_i*x.
% - The internal symbolic model matches goal2_main diamond ordering above.
% - If you use AvgTT in goal2_main, dt must be AvgTT_after - AvgTT_before.

% ========================================================================
% 0) Defaults / options
% ========================================================================
if nargin == 0
    % Example default consistent with goal2_main diamond ordering:
    d   = 4;
    dt  = 6;
    b14 = [5 30 40 0];   % [b(1->2), b(2->4), b(1->3), b(3->4)]
    m14 = [6 2  1  12];  % [m(1->2), m(2->4), m(1->3), m(3->4)]
    opts = struct();
elseif nargin < 4
    error('sf_equiv_family_link23:NotEnoughInputs', ...
        'Usage: sf_equiv_family_link23(d, m14, b14, dt, [opts])');
end
if nargin < 5 || isempty(opts), opts = struct(); end

% opts defaults
N_hits       = getdef(opts,'N_hits',10);
t_grid       = getdef(opts,'t_grid',0:0.25:50);
require_real = getdef(opts,'require_real',true);
require_pos  = getdef(opts,'require_pos',false);
do_simplify  = getdef(opts,'simplify',true);
verbose      = getdef(opts,'verbose',true);

% sanitize inputs
m14 = m14(:).';
b14 = b14(:).';
if numel(m14) ~= 4 || numel(b14) ~= 4
    error('sf_equiv_family_link23:BadInputs', ...
        'm14 and b14 must be 1x4 vectors in goal2_main order: [12,24,13,34].');
end
if ~isscalar(d) || ~isscalar(dt)
    error('sf_equiv_family_link23:BadInputs', 'd and dt must be scalars.');
end

% marker
P = 0; if dt > 0, P = 1; end

% ========================================================================
% 1) Symbolic model
%    a(1..5) slopes, b(1..5) intercepts
%    ORDER MATCHES goal2_main diamond AFTER legend:
%      1: 1->2
%      2: 2->4
%      3: 1->3
%      4: 3->4
%      5: 2->3
% ========================================================================
a  = sym('a',  [1 5], 'real');   % slopes
b  = sym('b',  [1 5], 'real');   % intercepts
D  = sym('D',  'real');

CB = sym('CB', [1 2], 'real');
CA = sym('CA', [1 3], 'real');

% AFTER incidence-like matrix (must align with link ordering above)
% Rows correspond to links in order: (12,24,13,34,23)
BA = sym([1 0 1;    % 1->2   (same as old row for 1-2)
          1 0 0;    % 2->4   (old row for 2-4)
          0 1 0;    % 1->3   (old row for 1-3)
          0 1 1;    % 3->4   (old row for 3-4)
          0 0 1]);  % 2->3   (old row for 2-3)

% ------------------------------------------------------------------------
```

```
% BEFORE average path cost closed form (TWO parallel paths):
%   Path A: 1->2->4 uses links 1 and 2  => slope (a1+a2), intercept (b1+b2)
%   Path B: 1->3->4 uses links 3 and 4  => slope (a3+a4), intercept (b3+b4)
% -------------------------------------------------------------------------
CB(1) = ( D ...
    + (b(1)+b(2))/(a(1)+a(2)) ...
    + (b(3)+b(4))/(a(3)+a(4)) ) ...
    / ( 1/(a(1)+a(2)) + 1/(a(3)+a(4)) );

% AFTER average path cost (matrix form)
M    = BA' * diag(a) * BA;
adjM = adjoint(M);

CA(1) = ( D*det(M) + [1 1 1]*((adjM*BA')*transpose(b)) ) ...
      ./ ( ([1 1 1]*adjM)*[1;1;1] );

% Substitute known 1..4 parameters, keep a(5), b(5)
z1 = subs(CA(1), [D, a(1:4), b(1:4)], [d, m14, b14]); % AFTER
z2 = subs(CB(1), [D, a(1:4), b(1:4)], [d, m14, b14]); % BEFORE

% Rename a(5),b(5) -> a5,b5
a5 = sym('a5','real');
b5 = sym('b5','real');

delta_expr = subs(z1 - z2, [a(5), b(5)], [a5, b5]);
if do_simplify
    delta_expr = simplify(delta_expr);
end

% ========================================================================
% 2) Solve Delta(a5,b5) == dt
% ========================================================================
eqn = (delta_expr == dt);

% professor-compatible solve order [b5 a5]
sol = solve(eqn, [b5 a5], 'ReturnConditions', true);

sa5 = sym([]); sb5 = sym([]);
if isfield(sol,'a5'), sa5 = sol.a5; end
if isfield(sol,'b5'), sb5 = sol.b5; end

% ========================================================================
% 3) Sample numeric pairs (b5,a5)
% ========================================================================
pairs = [];

params = sym([]);
if isfield(sol,'parameters') && ~isempty(sol.parameters)
    params = sol.parameters;
end

sa5v = sa5(:);
sb5v = sb5(:);

if isempty(sa5v) || isempty(sb5v)
    if verbose
        fprintf('\n[sf_equiv_family_link23] No symbolic solution returned.\n');
    end
    return;
end

for k = 1:max(numel(sa5v), numel(sb5v))
    if numel(sa5v) >= k, a5k = sa5v(k); else, a5k = sa5v(end); end
    if numel(sb5v) >= k, b5k = sb5v(k); else, b5k = sb5v(end); end

    if isempty(params)
        a_num = double(a5k);
        b_num = double(b5k);

```

```
        if require_real && (~isreal(a_num) || ~isreal(b_num)), continue; end
        if ~is_ok_num(a_num) || ~is_ok_num(b_num), continue; end
        if require_pos && ~(a_num > 0 && b_num > 0), continue; end

        pairs = [pairs; b_num, a_num]; %#ok<AGROW>
        if size(pairs,1) >= N_hits, break; end
        continue;
    end

    % Has free parameter(s): sample only first parameter
    t = params(1);
    for tv = t_grid
        a_sub = subs(a5k, t, tv);
        b_sub = subs(b5k, t, tv);

        a_num = double(a_sub);
        b_num = double(b_sub);

        if require_real && (~isreal(a_num) || ~isreal(b_num)), continue; end
        if ~is_ok_num(a_num) || ~is_ok_num(b_num), continue; end
        if require_pos && ~(a_num > 0 && b_num > 0), continue; end

        % de-duplicate
        if ~isempty(pairs)
            if any(abs(pairs(:,1)-b_num) < 1e-9 & abs(pairs(:,2)-a_num) < 1e-9)
                continue;
            end
        end

        pairs = [pairs; b_num, a_num]; %#ok<AGROW>
        if size(pairs,1) >= N_hits, break; end
    end

    if size(pairs,1) >= N_hits, break; end
end

% =========================================================================
% 4) Print summary
% =========================================================================
if verbose
    fprintf('\n[sf_equiv_family_link23] Inputs (goal2_main diamond order):\n');
    fprintf('  Q=d       = %.12g\n', d);
    fprintf('  m14       = [%s]   (links: 12,24,13,34)\n', sprintf('%.12g ', m14));
    fprintf('  b14       = [%s]   (links: 12,24,13,34)\n', sprintf('%.12g ', b14));
    fprintf('  dt target = %.12g  (P=%d)\n\n', dt, P);

    fprintf('  Delta(a5,b5) = %s\n\n', char(delta_expr));

    fprintf('  Symbolic solution (a5): %s\n', char(sa5));
    fprintf('  Symbolic solution (b5): %s\n', char(sb5));

    if isfield(sol,'conditions') && ~isempty(sol.conditions)
        fprintf('  conditions: %s\n', char(sol.conditions));
    end
    if isfield(sol,'parameters') && ~isempty(sol.parameters)
        fprintf('  parameters: %s\n', char(sol.parameters));
        if numel(sol.parameters) > 1
            fprintf('  NOTE: multiple parameters detected; sampling only the first one.\n');
        end
    end

    if isempty(pairs)
        fprintf('\n  No numeric pairs sampled.\n');
        fprintf('  Try enlarging opts.t_grid, e.g.: opts.t_grid = 0:0.1:500;\n\n');
    else
        fprintf('\n  Sampled %d numeric (b5,a5) pairs (up to N_hits=%d):\n', size(pairs,1), N_hits);
        disp(array2table(pairs, 'VariableNames', {'b5','a5'}));

        fprintf('=== Paste-ready lines for goal2_main (diamond, edge 5 = 2->3) ===\n');
```

```
237         for i = 1:size(pairs,1)
238             fprintf('%% #%d  (b5,a5) = (%.12g, %.12g)\n', i, pairs(i,1), pairs(i,2));
239             fprintf('CFG.cost.b(5) = %.12g;  CFG.cost.m(5) = %.12g;\n\n', pairs(i,1), pairs(i,2));
240         end
241     end
242 end
243 
244 end
245 
246 % --------------------- local utilities (NOT nested) ---------------------
247 function v = getdef(S, name, default)
248 if isstruct(S) && isfield(S, name)
249     v = S.(name);
250 else
251     v = default;
252 end
253 end
254 
255 function tf = is_ok_num(x)
256 tf = isfinite(x) && ~isnan(x);
257 end
```

B.7. Script 6: goal2_find_global_linear_equiv.m (numerical equivalence search/verification)

B.7.1. What this script computes

The function goal2_find_global_linear_equiv() numerically searches for linear parameters (b, m) on the added link (edge 5, 2→3) such that, at a fixed demand value Q_given, the computed equilibrium outputs match a reference run:

- the same before-network link-flow vector $\mathbf{X}^{(B)}$,
- the same after-network link-flow vector $\mathbf{X}^{(A)}$,
- the same ΔAvgTT.

For each candidate slope m, it solves for b via fminbnd and accepts pairs meeting strict flow and delta tolerances.

B.7.2. User-editable parameters (what to edit)

At the top of the file:

- Reference configuration CFG (same interface as goal2_main).
- Verification demand: Q_given.
- Candidate slopes: m_candidates.
- Acceptance tolerances: tol_flow, tol_delta.
- Search range controls: b_search_span, b_fmin_tol.

B.7.3. Outputs

The script prints the reference equilibrium, validation results, accepted (b, m) pairs, and paste-ready assignments for edge 5. No files are written by default.

B.7.4. Source code listing

Listing B.6: Script 6: Numerical equivalence search at a fixed demand (`goal2\_find\_global\_linear\_equiv.m`)

```
function goal2_find_global_linear_equiv()
% GOAL2_FIND_GLOBAL_LINEAR_EQUIV
% Find GLOBAL linear equivalent family for link 2->3 in Braess diamond/ladder
% such that, at a GIVEN demand Q_given, the UE results are identical:
%   - same AFTER link flows XA (all edges)
%   - same BEFORE link flows XB (all edges)
%   - same Delta_AvgTT
%
% Interface: uses the same CFG fields as your current goal2_main.
%
% Output:
%   Prints a list of (b,m) for link 2-3 you can paste into goal2_main
%   (keeping other costs unchanged).
%
% Note:
%   This verifies equivalence at ONE Q_given (professor statement).
%   If you also want equivalence over a scan of Q, change objective below.

clc; close all; fprintf('\n[goal2_find_global_linear_equiv] start\n\n');

%% ==================== USER CFG (same interface as goal2_main) ====================
CFG = struct();

% NETWORK
CFG.net.kind = 'diamond';   % 'diamond' | 'ladder'
CFG.net.dir  = 'oneway';

% Ladder-only (ignored for diamond)
CFG.net.n = 3;
CFG.net.added_edges_nodes = [];

% COST (diamond AFTER has 5 edges: 1->2,2->4,1->3,3->4,2->3)
CFG.cost.type      = 'per-link';
CFG.cost.link_type = {'linear','linear','linear','linear','linear'};
CFG.cost.b = [0, 50, 50, 0, 10];   % reference b for link 2-3 is index 5
CFG.cost.m = [10, 1,  1,  10, 1];  % reference m for link 2-3 is index 5

% Keep BPR fields for compatibility
CFG.cost.t0  = [NaN NaN NaN NaN NaN];
CFG.cost.cap = [NaN NaN NaN NaN NaN];
CFG.cost.alpha = 1.6;
CFG.cost.beta  = 2.0;

CFG.cost.added_override = struct('link_type', {{}}, 't0', [], 'cap', [], 'b', [], 'm', []);

% ASSIGNMENT
CFG.assign.kind  = 'DUE';   % 'DUE' | 'SUE'
CFG.assign.theta = 0.8;     % SUE only

% SOLVER OPTIONS
CFG.opts = struct( ...
  'tol',         1e-10, ...
  'max_iter',    5e4, ...
  'paradox_eps', 1e-9, ...
  'zero_tol',    1e-10);
```

```
56
57 % ===== professor "given Q" =====
58 Q_given = 6;         % <=== change this
59 edge23  = 5;         % for diamond AFTER, 2->3 is edge #5 in stable order
60
61 % ===== search settings =====
62 m_candidates = unique([ ...
63     1, ...                % ref
64     2 3 4 5, ...          % professor examples slopes
65     linspace(0,10,81) ...   % coarse grid
66 ]);
67
68 % tolerance for accepting equivalence
69 tol_flow  = 1e-8;
70 tol_delta = 1e-8;
71
72 % for each m, solve b via 1D search around a sensible range
73 b_search_span = 200;   % +/- around b_init
74 b_fmin_tol    = 1e-12;
75
76 %% ==================== 1) REFERENCE RUN at Q_given ====================
77 CFG_ref = CFG;
78 CFG_ref.Q = Q_given;
79
80 [refOut] = run_goal2_singleQ(CFG_ref);
81
82 XB_ref    = refOut.XB;
83 XA_ref    = refOut.XA;
84 delta_ref = refOut.dAvg;
85
86 fprintf('Reference at Q=%.6g\n', Q_given);
87 fprintf('  XB_ref = [%s]\n', sprintf('%.12g ', XB_ref));
88 fprintf('  XA_ref = [%s]\n', sprintf('%.12g ', XA_ref));
89 fprintf('  Delta_ref = %.12g\n\n', delta_ref);
90
91 %% ==================== 2) QUICK CHECK professor examples directly ====================
92 profPairs = [ ...
93     10 1;  % x+10
94      8 2;  % 2x+8
95      6 3;  % 3x+6
96      4 4;  % 4x+4
97      2 5;  % 5x+2
98 ];
99
100 fprintf('Check professor example pairs (b,m):\n');
101 for i=1:size(profPairs,1)
102     b0 = profPairs(i,1); m0 = profPairs(i,2);
103     [ok, eFlow, eDel] = check_pair(CFG, Q_given, edge23, b0, m0, XB_ref, XA_ref, delta_ref, tol_flow,
        tol_delta);
104     fprintf('  (b=%.6g, m=%.6g)  ok=%d   max|dFlow|=%.3e   |dDelta|=%.3e\n', ...
105         b0,m0, ok, eFlow, eDel);
106 end
107 fprintf('\n');
108
109 %% ==================== 3) SEARCH "global linear" family ====================
110 sol = [];
111 solLabels = {};
112
113 fprintf('Search global linear equivalents at Q=%.6g ...\n', Q_given);
114
115 for im = 1:numel(m_candidates)
116     m_try = m_candidates(im);
117
118     % initial b guess: keep cost at current x23 approximately similar
119     x23_ref = XA_ref(edge23);
120     c23_ref = CFG.cost.b(edge23) + CFG.cost.m(edge23) * x23_ref;
121     b_init  = c23_ref - m_try * x23_ref;
122
123     % objective: min over b of max-norm flow mismatch + delta mismatch
```

```
124     obj = @(b) objective_for_b(CFG, Q_given, edge23, b, m_try, XB_ref, XA_ref, delta_ref);
125
126     % search window
127     b_lo = b_init - b_search_span;
128     b_hi = b_init + b_search_span;
129
130     % 1D minimization
131     b_star = fminbnd(obj, b_lo, b_hi, optimset('TolX', b_fmin_tol, 'Display','off'));
132     [J, eFlow, eDel] = obj(b_star);
133
134     if eFlow <= tol_flow && eDel <= tol_delta
135         sol(end+1,:) = [b_star, m_try, eFlow, eDel]; %#ok<AGROW>
136         solLabels{end+1,1} = sprintf('FOUND: b=%.12g, m=%.12g', b_star, m_try); %#ok<AGROW>
137         fprintf('   m=%.6g -> b=%.12g   max|dFlow|=%.3e   |dDelta|=%.3e\n', m_try, b_star, eFlow, eDel);
138     end
139 end
140
141 fprintf('\nFound %d equivalent (b,m) pairs at Q=%.6g.\n\n', size(sol,1), Q_given);
142
143 %% ===================== 4) PRINT IN "goal2_main paste-ready" FORMAT =====================
144 if isempty(sol)
145     fprintf('No solutions found under current tolerances / search range.\n');
146     fprintf('Try: enlarge b_search_span, relax tol_flow/tol_delta, or increase max_iter.\n');
147     return;
148 end
149
150 % sort by slope then intercept
151 sol = sortrows(sol, [2 1]);
152
153 fprintf('========== PASTE INTO goal2_main ==========\n');
154 fprintf('Keep other edges unchanged. Replace ONLY edge-5 b,m:\n\n');
155
156 fprintf('%% Candidate (b,m) for link 2-3 (edge 5), verified at Q_given=%.6g\n', Q_given);
157 for i=1:size(sol,1)
158     fprintf('%% #%d:  b23=%.12g,  m23=%.12g   (max|dFlow|=%.3e, |dDelta|=%.3e)\n', ...
159         i, sol(i,1), sol(i,2), sol(i,3), sol(i,4));
160 end
161
162 fprintf('\n%% Example paste (choose one pair):\n');
163 fprintf('CFG.cost.b = [0, 50, 50, 0, %.12g];\n', sol(1,1));
164 fprintf('CFG.cost.m = [10, 1,  1,  10, %.12g];\n', sol(1,2));
165 fprintf('CFG.Q = %.6g;  %% IMPORTANT: verify at the same single Q_given\n', Q_given);
166 fprintf('============================================\n\n');
167
168 fprintf('[goal2_find_global_linear_equiv] done\n');
169
170 end
171
172 %% ========================================================================
173 % Objective wrapper: returns [J, eFlow, eDel]
174 function [J, eFlow, eDel] = objective_for_b(CFG, Q_given, edge23, b23, m23, XB_ref, XA_ref, delta_ref)
175     CFG_t = CFG;
176     CFG_t.Q = Q_given;
177     CFG_t.cost.b(edge23) = b23;
178     CFG_t.cost.m(edge23) = m23;
179
180     out = run_goal2_singleQ(CFG_t);
181
182     eB = max(abs(out.XB(:) - XB_ref(:)));
183     eA = max(abs(out.XA(:) - XA_ref(:)));
184     eFlow = max(eA, eB);
185     eDel  = abs(out.dAvg - delta_ref);
186
187     % scalar objective (weights can be tuned)
188     J = eFlow + 10*eDel;
189 end
190
191 function [ok, eFlow, eDel] = check_pair(CFG, Q_given, edge23, b23, m23, XB_ref, XA_ref, delta_ref, tol_flow
      , tol_delta)
```

```
    [~, eFlow, eDel] = objective_for_b(CFG, Q_given, edge23, b23, m23, XB_ref, XA_ref, delta_ref);
    ok = (eFlow <= tol_flow) && (eDel <= tol_delta);
end

%% ========================================================================
% Run UE once at single Q, return numeric flows and Delta
function out = run_goal2_singleQ(CFG)
    opts = CFG.opts;
    tol = opts.tol;
    max_iter = opts.max_iter;

    Q = CFG.Q;
    if numel(Q) ~= 1
        error('run_goal2_singleQ expects CFG.Q to be a scalar.');
    end

    [linksB, linksA, s, t] = build_network_from_CFG(CFG.net);
    added_idx = detect_added_edges(linksB, linksA);

    PMAX = 20000;
    B1 = incidence_from_all_paths_DAG(linksB, s, t, PMAX);
    B2 = incidence_from_all_paths_DAG(linksA, s, t, PMAX);

    [F_B, F_A] = build_costs_after(linksA, CFG.cost, added_idx);

    assign_kind = upper(CFG.assign.kind);
    theta_def   = CFG.assign.theta;

    switch assign_kind
        case 'DUE'
            [xB, XB, cB, CB] = calc_DUE(F_B, B1, Q, tol, max_iter);
            [xA, XA, cA, CA] = calc_DUE(F_A, B2, Q, tol, max_iter);
        case 'SUE'
            [xB, XB, cB, CB] = calc_SUE(F_B, B1, Q, tol, max_iter, theta_def);
            [xA, XA, cA, CA] = calc_SUE(F_A, B2, Q, tol, max_iter, theta_def);
        otherwise
            error('Unknown assignment kind.');
    end

    % total path-cost form (consistent with your goal2_main)
    TTb = xB * CB';
    TTa = xA * CA';

    avgB = TTb / max(Q, eps);
    avgA = TTa / max(Q, eps);

    out = struct();
    out.Q = Q;
    out.XB = XB(:).';
    out.XA = XA(:).';
    out.avgB = avgB;
    out.avgA = avgA;
    out.dAvg = avgA - avgB;
end

%% ========================================================================
% ====== Below: minimal dependency copies (aligned to your goal2_main) ======
function [linksB, linksA, s, t] = build_network_from_CFG(netCFG)
    kind = lower(netCFG.kind);
    dir  = getdef(netCFG, 'dir', 'oneway');
    switch kind
        case 'diamond'
            [linksB, linksA, s, t] = build_links_diamond(dir);
        case 'ladder'
            n     = getdef(netCFG, 'n', 3);
            nodes = getdef(netCFG, 'added_edges_nodes', []);
            [linksB, linksA, s, t] = build_links_ladder_unified(n, dir, nodes);
        otherwise
            error('Unknown network kind: %s', kind);
```

```
    end
end

function [linksB, linksA, s, t] = build_links_diamond(dir)
    linksB = [1 2; 2 4; 1 3; 3 4];
    edgesA = [linksB; 2 3];
    if strcmpi(dir,'twoway'), edgesA=[edgesA; 3 2]; end
    linksA = unique(edgesA,'rows','stable');
    s=1; t=4;
end

function [linksB, linksA, s, t] = build_links_ladder_unified(n, dir, added_edges_nodes)
    U = @(i) i;
    L = @(j) j + (n+1);
    S = 2*n + 3;
    T = 2*n + 4;

    edgesB = [];
    edgesB = [edgesB; S U(1)];
    edgesB = [edgesB; S L(1)];
    for i=1:n, edgesB=[edgesB; U(i) U(i+1)]; end
    for i=1:n, edgesB=[edgesB; L(i) L(i+1)]; end
    edgesB=[edgesB; U(n+1) T];
    edgesB=[edgesB; L(n+1) T];
    linksB = edgesB;

    edgesA = linksB;
    if ~isempty(added_edges_nodes)
        added = validate_and_clip_edges(added_edges_nodes, n);
        for r=1:size(added,1)
            u=added(r,1); v=added(r,2);
            edgesA=[edgesA; u v];
            if strcmpi(dir,'twoway'), edgesA=[edgesA; v u]; end
        end
    end
    linksA = unique(edgesA,'rows','stable');
    s=S; t=T;
end

function newE = validate_and_clip_edges(newE, n)
    maxNode = 2*n + 4;
    bad = any(newE<1,2) | any(newE>maxNode,2) | (newE(:,1)==newE(:,2));
    newE(bad,:) = [];
end

function added_idx = detect_added_edges(linksB, linksA)
    mA = size(linksA,1);
    added_idx = nan(mA,1);
    for k=1:mA
        if ~ismember(linksA(k,:), linksB,'rows')
            added_idx(k)=k;
        end
    end
end

function [F_B, F_A] = build_costs_after(linksA, costCFG, added_idx)
    mA   = size(linksA, 1);
    isAdd = ~isnan(added_idx);

    F_A.kind      = 'per-link';
    F_A.link_type = expand_cell(costCFG, 'link_type', mA, 'linear');
    F_A.t0        = expand_vec(costCFG, 't0',   mA, NaN);
    F_A.cap       = expand_vec(costCFG, 'cap',  mA, NaN);
    F_A.alpha     = getdef(costCFG, 'alpha', 1.6);
    F_A.beta      = getdef(costCFG, 'beta',  2.0);
    F_A.b         = expand_vec(costCFG, 'b',    mA, NaN);
    F_A.m         = expand_vec(costCFG, 'm',    mA, NaN);

    % BEFORE = AFTER without added edges
```

```
330     keep = ~isAdd;
331     F_B  = sub_cost_object(F_A, keep);
332 end
333 
334 function F2 = sub_cost_object(F, keep)
335     F2 = F;
336     F2.link_type = F.link_type(keep);
337     F2.t0        = F.t0(keep);
338     F2.cap       = F.cap(keep);
339     F2.b         = F.b(keep);
340     F2.m         = F.m(keep);
341 end
342 
343 function c = eval_link_costs(F, X)
344     X = X(:);
345     m = numel(X);
346     c = zeros(m,1);
347     for e=1:m
348         t = lower(F.link_type{e});
349         switch t
350             case 'linear'
351                 b = F.b(e); mm = F.m(e);
352                 if isnan(b), b=0; end
353                 if isnan(mm), mm=0; end
354                 c(e) = b + mm*X(e);
355             case 'bpr'
356                 t0 = F.t0(e); cap = F.cap(e);
357                 if isnan(cap) || cap<=0, cap=inf; end
358                 c(e) = t0 * (1 + F.alpha * (X(e)/cap)^F.beta);
359             otherwise
360                 error('Unknown cost type: %s', t);
361         end
362     end
363     c = c.';
364 end
365 
366 function [x, X, c, C] = calc_DUE(F, B, Q, tol, max_iter)
367     P = size(B,1);
368     x = zeros(1,P);
369     X = x*B;
370     for it=1:max_iter
371         c = eval_link_costs(F, X);
372         C = (B*c.').';
373         Cmin = min(C);
374         I = find(C <= Cmin + 1e-12);
375         y = zeros(1,P);
376         y(I) = Q/numel(I);
377 
378         a = 1/it;
379         x_new = x + a*(y-x);
380         X_new = x_new * B;
381 
382         rel = norm(X_new - X, 1) / max(Q,1);
383         if rel < tol
384             x=x_new; X=X_new; break;
385         end
386         x=x_new; X=X_new;
387     end
388     c = eval_link_costs(F, X);
389     C = (B*c.').';
390 end
391 
392 function [x, X, c, C] = calc_SUE(F, B, Q, tol, max_iter, theta)
393     P = size(B,1);
394     x = zeros(1,P);
395     X = x*B;
396     for it=1:max_iter
397         c = eval_link_costs(F, X);
398         C = (B*c.').';
```

```
        z = -C / max(theta,1e-12);
        z = z - max(z);
        p = exp(z); p = p/sum(p);
        y = Q*p;

        a = 1/it;
        x_new = x + a*(y-x);
        X_new = x_new * B;

        rel = norm(X_new - X, 1) / max(Q,1);
        if rel < tol
            x=x_new; X=X_new; break;
        end
        x=x_new; X=X_new;
    end
    c = eval_link_costs(F, X);
    C = (B*c.').';
end

function B = incidence_from_all_paths_DAG(links, s, t, PMAX)
    if nargin<4, PMAX=20000; end
    N = max(links(:));
    adj = cell(N,1);
    for k=1:size(links,1)
        adj{links(k,1)} = [adj{links(k,1)}, links(k,2)];
    end

    paths = {};
    cur = zeros(1, PMAX);

    function dfs(u, depth, visited)
        if depth > PMAX, return; end
        if u==t
            paths{end+1} = cur(1:depth-1);
            return;
        end
        for v = adj{u}
            if visited(v), continue; end
            cur(depth)=v;
            visited2 = visited; visited2(v)=true;
            dfs(v, depth+1, visited2);
        end
    end

    visited0 = false(1,N);
    visited0(s)=true;
    cur(1)=s;
    dfs(s,2,visited0);

    m = size(links,1);
    P = numel(paths);
    B = zeros(P,m);
    for p=1:P
        seq = [s paths{p}];
        for i=1:(numel(seq)-1)
            u=seq(i); v=seq(i+1);
            idx = find(links(:,1)==u & links(:,2)==v, 1, 'first');
            if ~isempty(idx), B(p,idx)=1; end
        end
    end
end

function v = getdef(S, name, default)
    if isstruct(S) && isfield(S,name), v=S.(name); else, v=default; end
end

function v = expand_vec(S, name, L, defaultVal)
    if isfield(S,name)
        v = S.(name);
```

```
        if isscalar(v), v=repmat(v,L,1); end
    else
        v = defaultVal*ones(L,1);
    end
    v=v(:);
    if numel(v)<L, v(end+1:L,1)=defaultVal; end
end

function C = expand_cell(S, name, L, defaultVal)
    if isfield(S,name)
        C = S.(name);
        if ischar(C), C={C}; end
    else
        C = cell(0,1);
    end
    if numel(C)<L, C(end+1:L)={defaultVal}; end
    C=C(:);
end
```