POLITECNICO DI MILANO
Corso di Laurea Magistrale in Computer Science and Engineering
Scuola di Ingegneria Industriale e dell'Informazione

Bifurcation Analysis of Auction Mechanisms

Relatore: Prof. Nicola Gatti

Tesi di Laurea di:
Chianella Claudia Beatrice
Matricola: 877612

Anno Accademico 2018 - 2019

Sommario

Quello dei meccanismi economici è uno dei campi attualmente studiati della comunità scientifica. Abbiamo applicato strumenti della teoria dei giochi algoritmica a problemi d'asta con l'obiettivo di progettare un meccanismo in grado di guidare il sistema verso uno stato con un Social Welfare migliore rispetto a quello dell'equilibrio di Nash. Abbiamo studiato ed analizzato i meccanismi della First Price, Generalized Second Price e della GSP ripetuta con budget fissato. Ogni giocatore che prende parte al meccanismo utilizza strategie adattive, comunemente dette di learning. Oggetto di studio sono state le dinamiche evolutive del Q-Learning, FAQ e Gradient Ascent. Per modellare le dinamiche di apprendimento e studiare la calibrazione dei parametri abbiamo utilizzato strumenti di teoria dei sistemi, mentre, per lo studio di biforcazione, abbiamo utilizzato MATCONT un tool di Matlab in grado di analizzare sistemi di equazioni dinamiche. Infine è stato progettato un meccanismo in grado di guidare il sistema verso uno stato con Social Welfare migliore rispetto a quello del Nash che non è raggiungibile nel caso di giocatori razionali.

Abstract

This thesis focuses on the field of algorithmic applied to problems of decision with opponents, commonly called algorithmic game theory. This field combines mathematical models, which describe situations of strategic interaction, with algorithmic tools, which allow to find solutions that prescribe to each player the best strategies to be implemented. Within this framework, a topic currently studied by the scientific community is the development of economic mechanisms when the players, who will take part in the mechanism, use adaptive strategies, commonly called learning strategies. For example, a typically used algorithm is Q-Learning. To design the best economic mechanism it is necessary to model the learning dynamics of the players and study the calibration of the parameters of the mechanism using system theory tools, such as bifurcation studies. The objective of this project is to study this type of problem in the context of the auction scenarios. In particular, online advertising auctions (used, for example, by Google, Microsoft, Amazon) have been analyzed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Aim of the work

This work focuses on the field of Algorithmic Game Theory. This field combines mathematical models, which describe situations of strategic interaction, with algorithmic tools, which allow one to find for each player the best strategy to implement. A topic currently studied by the scientific community is the development of economic mechanisms satisfying notable properties. In this work, we will study this problem in the context of auction scenarios.

In particular, we focus on mechanisms for online advertising. This scenario represents the main source of marketing and, in the recent years, its revenue is dramatically increasing. This growth is due to the fact that the number of people connected to the Internet in the world and the time spent online are constantly growing and, also compared to other communication channels such as radio, TV, print, etc. is faster and more accessible. Online advertising is the main source of revenue of important companies like Google, Microsoft, Yahoo etc. Online adds are sold throw actions where bidders compete for the available slots. The study of auctions for advertising is central in Artificial Intelligence community and explored along different perspectives, e.g., Ceppi et al. [2011]; Gatti et al. [2012, 2015]

We will study and analyze three auction mechanisms: First Price Sealed bid auction, Generalized Second Price auction, and Repeated GSP with budget constraints.

First Price Sealed Bid Auction, simply called First Price (FP), is the simplest and most common used auction mechanism where the winner player is allocated in the first slot and pays the same amount he bids. The Generalized Second Price auction (GSP) is a non-truthful mechanism and it is a little more involved with respect the FP. Each player for the allocation pays a value equal to the bid value of the player allocated in the next slot. The third auction we study is a slightly modification of GSP in which constraints are introduced. The allocation mechanism is the same as GSP, the auction is repeated over time and each player participates until it runs out of budget.

These mechanisms will be analyzed from a dynamic perspective in which each agent will learn the best strategy through learning algorithms. This perspective is very common in practice. Reinforcement Learning is a field of Machine Learning in which each agent learns the optimal strategy through iteration with the environment. Agent selects an action and the environments respond to this action presenting a new situation to the agent and providing a reward which is a value that the agent wants to maximize. Learning in a Multi-Agents environment is significantly more complex than single-agent learning, as the optimal behaviour will depends not only form the environment but also from the strategy of other players.

Recently evolutionary game theory has been linked to reinforcement learning algorithms, we will analyze the Q-Learning, FAQ and Gradient Ascent dynamics. The Q-learning is a value-iteration method for solving the optimal strategies in Markov decision process where agent learns his strategy throw Q-learning and Boltzmann selection rules. The tread off between exploration and exploitation is controlled by the temperature parameter. The FAQ dynamics is a variation of Q-Learning which uses a softmax activation function for policy generation. Gradient Ascent is a well-known optimisation technique in the field of machine learning, given a set of differential equation the learning process follows the direction of the gradient in order to find the local optima. This concept can be applied to multiagent learning where each agent learns the optimal policy following the gradient direction of its individual expected reward. This approach assumes that the expected payoff function is known to the agents, which may not be feasible in real context.

To design the best economic mechanism is necessary to model the learning dynamics and study the calibration of parameters using system theory tools such as bifurcation studies. Bifurcation theory is a mathematical study of qualitatively changes in systems dynamics produced by varying parameters. We have studied and analyzed the stability of the system of dynamic equations using MATCONT, a MATLAB tool for the interactive bifurcation analysis of dynamical systems.

Finally, we design an Optimal Control Mechanism capable of driving the system towards a state with a Social Welfare higher than the Nash Equilibria. Using the Q-Learning dynamics and controlling the temperature parameters we are able to reach a better state than those achievable by perfectly rational agent.

1.2 Thesis structure

The thesis is structured in the following way:

- Chapter 2 provides the basics of game theory, evolutionary game theory, multiagent learning and it also presents the state of art.
- Chapter 3 presents the auction mechanism that we study and the first positive result: auction games cannot be trace back to coordination games.
- Chapter 4 provides the basic notion of dynamical system and presents the results of the bifurcation study applied to auctions problems.
- Chapter 5 designs a mechanism to improve the social welfare controlling the temperature parameter.
- Chapter 6 contains the conclusions drawn from the results obtained in the previous chapters.

Chapter 2

Preliminaries and state of art

2.1 Game Theory

Game Theory is the field which studies mathematical models of strategic interaction between rational decision-makers, that are commonly called agents or players. Each player has a goal to pursue and based on this he will choose his actions, that will affect the outcome of all the other decision makers. Each player is perfectly rational, he has a clear preference over outcomes and chooses the action that maximizes his reward assuming that also the others do the same.

A **game** $\mathcal{G}$ is described by the following elements:

- a set of players: $\mathcal{N} = \{1, ..., n\}$;
- a set of actions: $\mathcal{A}_i$ for each player;
- payoff function: $u_i : \mathcal{A} \rightarrow R$ for each player.

The main representation of a game is the *Normal Form*, that is a matrix-based representation which describes situations in which decision makers play simultaneously and each entry of the matrix corresponds to an outcome of the game.

2.1.1 Games in normal form

Normal form games model the scenario in which agents execute actions simultaneously according to their strategies. The fact that agents play simultaneously does not mean that they play at the same time but that a player does not know what the opponents will do.

Definition 1 (Normal Form Game)**.** *The normal-form representation of a game is a triplet* $(\mathcal{N}, \mathcal{A}, \mathcal{U})$ *where:*

- $\mathcal{N} = \{1, ..., n\}$ *is the set of player;*
- $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_n\}$ *is the set of actions of all the players where* $\mathcal{A}_i = \{a_1, a_2, ..., a_{m_i}\}$ *is the set of player i's actions;*
- $\mathcal{U} = \{u_1, u_2, ..., u_n\}$ *is the set of the utility functions of all the players where* $u_i : \mathcal{A}_1 \times \mathcal{A}_2 \times ... \times \mathcal{A}_n \rightarrow \mathcal{R}$ *is the utility function of player i.*

These games can be represented as n-dimensional matrix where the row corresponds to a possible action of player A and column to possible actions of player B, the combination of actions will lead to an outcome that is reported in the corresponding cell. An example is reported below

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},$$

Figure 2.1: 2×2 *Normal Form Game.*

where A and B are the payoff matrix of the two players, element a_{ij} is the reward of player A when he chooses actions i and his opponent chooses action j and similarly for element b_{ij}.

Definition 2 (Action profile)**.** *An action profile* $\mathbf{a}$ *is a tuple* $(a_1, a_2, ..., a_n)$ *with* $a_i \in \mathcal{A}_i$*, containing one action per player. Action profile* $\mathbf{a_{-i}}$ *is a tuple* $(a_1, a_2, ..., a_{i-1}, a_{i+1}, ..., a_n)$ *with* $a_j \in \mathcal{A}_j$*, containing one action per player except for player i.*

We denote by $\mathcal{A}_{-i} = \mathcal{A}_1 \times \mathcal{A}_2 \times ...\mathcal{A}_{i-1} \times \mathcal{A}_{i+1} \times ...\mathcal{A}_n$ *the space of* a_{-i}*.*

2.1.2 Strategies

In normal form games the behaviour of each player is described by his strategy which is the set of actions that he plans to choose during the game. A strategy profile is a collection of strategy one for each player. Formally:

Definition 3 (Strategy)**.** *Strategy* $\sigma_i : \mathcal{A}_i \rightarrow [0, 1]$ *with* $\sigma_i \in \Delta(\mathcal{A}_i)$ *is a function returning the probability with which each action* $a_i \in \mathcal{A}_i$ *is played by player i.*

Definition 4 (Strategy profile)**.** *A strategy profile* σ *is a tuple* $(\sigma_1, \sigma_2, ..., \sigma_n)$ *containing one strategy per player. Strategy profile* σ_{-i} *is a tuple* $(\sigma_1, \sigma_2, ..., \sigma_{i-1}, \sigma_{i+1}, ..., \sigma_n)$*, containing one strategy per player except for player i.*

If an agent plays deterministically one action the strategy is *pure* otherwise if there is a probability distribution over actions the strategy is *mixed.* A pure strategy can be seen as a special case of mixed strategy where the agent plays one of his actions with probability of 1 and the other with 0 probability.

Definition 5 (Pure/mixed strategies). *Strategy σ_i is pure, if there is action $a \in \mathcal{A}_i$ such that $\sigma_i(a) = 1$, and it is mixed otherwise. Strategy σ_i is fully mixed if it holds $\sigma_i(a) > 0$ for each action $a \in \mathcal{A}_i$.*

The expected utility of a player can be computed as the sum over the payoff of all possible strategy profiles.

Definition 6 (Expected utility). *Expected utility $E_{a\sim\sigma}[U_i(\mathbf{a})]$ returns the expected value of the utility of player i given strategy profile σ. The formula $E_{a\sim\sigma}[U_i(\mathbf{a})]$ can be written as:*

$$E_{a\sim\sigma}[U_i(\mathbf{a})] = \sum_{a_1\in A_1}\sum_{a_2\in A_2}...\sum_{a_n\in A_n}\sigma_1(a_1)\sigma_2(a_2)...\sigma_n(a_n)U_i(a_1,a_2,...,a_n)$$

The degree of the polynomial is n and, given player i, the expected utility is linear in player i's strategy.

The main problem for a player in strategic form game is to decide which strategy to play making prediction about what other players will do while predicting how the other players will play is easy when a player has a dominant strategy which is the one that produces the highest payoff regardless the strategies undertaken by the opponents.

Definition 7 (Dominant strategy). *Let σ_i and σ'_i be two strategies for player i and Σ_{-i} the set of all strategy profiles of the remaining players. Strategy σ_i is a dominant strategy for player i if $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$ for all σ'_i and all $\sigma_{-i} \in \Sigma_{-i}$. A strategy is strictly dominant if $u_i(\sigma_i, \sigma_{-i}) > u_i(\sigma'_i, \sigma_{-i})$.*

2.1.3 Nash equilibrium

Nash Equilibrium (NE) is one of the most important concepts in Game Theory and is named from mathematician John Forbes Nash Jr. Informally, a strategy profile is a Nash Equilibrium if no player has incentive to change his strategy.

The concept of *Best Response* is linked with NE and is defined as the strategy that will lead to the most favorable outcome for a player. Due to rationality assumption, all player are assumed to choose their best action. A mixed strategy σ_i is a best response for player i if there is no other strategy σ_i' that allows player i to achieve a better outcome. Formally:

Definition 8 (Best Response). *Player i's best response to the strategy profile σ_{-i} is a mixed strategy $\sigma_i^* \in \Sigma$ such that $u_i(\sigma_i^*, \sigma_{-i}) \geq u_i(\sigma_i, \sigma_{-i})$ for all strategies $\sigma_i \in \Sigma_i$.*

The concept of best response is central in the definition of *Nash Equilibrium*, in which no player has incentive to change his strategy if he knows what strategy the other players will follow. A game can have multiple Nash Equilibria. Formally:

Definition 9 (Nash Equilibrium). *A strategy profile $\sigma = (\sigma_1, ..., \sigma_n)$ is a Nash Equilibrium if, for all the agents i, σ_i is a best response to σ_i.*

Example 1 (Prisoner's dilemma). *In this game there are two players who are suspected of having committed a crime, they are placed into two different rooms and they can "confess" or "defect" the crime: $a_1 = \{C, D\}$ and $a_2 = \{c, d\}$.*

	c	d
C	3,3	0,5
D	5,0	1,1

Figure 2.2: Prisoner's dilemma.

If both confess they will spend 3 *years in jail, if only one of them confess and the other does not the collaborator will be free while the other will spend* 5 *years. If nobody confesses, but there is enough evidence to charge them for a minor crime, both will spend* 1 *year in jail.*

Individually, defection is a best response against any opponent strategy, and as a result mutual defection is the single Nash equilibrium of the game. However, both players would be better off if both would cooperate – hence the dilemma.

A central problem is computer science is the computation of equilibria, e.g., algorithms to compute Nash equilibria with two players as those proposed by Ceppi et al. [2010], algorithms to verify Nash equilibria with specific properties as those proposed by Gatti et al. [2013b], or algorithms to compute other solution concepts as those proposed by Coniglio et al. [2017]. Many application of this research field can be found in security, see, e.g., the works by Munoz de Cote et al. [2013]; Basilico et al. [2016, 2017], or in negotiations, see, e.g., the works by Di Giunta and Gatti [2006]; An et al. [2009, 2013]; Lopes and Coelho [2014]. In this thesis, our aim is not to compute an equilibrium, but to learn it. In the following sections, we survey the topics related to learning.

2.2 Evolutionary Game Theory

Evolutionary Game Theory (EGT) has origin from a series of publications by the mathematical biologist John Maynard Smith (Sandholm [2010], Shoham and Leyton-Brown [2010]). EGT studies the evolution of populations of agents and dynamical systems. The two central concepts are the Replicator Dynamics and Evolutionary Stable Strategy (ESS).

The *replicator dynamics* describe how a population evolves over time. It is composed of a set of differential equations that are derived from biological operators such as selection, mutation and cross-over.

In biology the replicator dynamic can be interpreted as a model of natural selection while in economic as a model of imitation (Sandholm [2007]).

In Evolutionary game theory the notion of Nash equilibrium is redefined with the concept of *Evolutionary Stable Strategy (ESS)*. An evolutionary stable strategy is a strategy that is immune to invasion by a small group of mutants who play an alternative mixed strategy.

2.2.1 Replicator Dynamics

The replicator dynamics describe how a population of individuals evolves over time under an evolutionary process. The probability distribution of the individuals inside the population is described by the vector $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, with $0 \leq x_i \leq 1 \forall i$ and $\sum_i x_i = 1$,that is equivalent to a policy for one player where x_i represents the probability of playing action i, or the fraction of the population that belongs to species i.

Let $f_i(\mathbf{x})$ be the fitness function of species i and $f(\bar{\mathbf{x}}) = \sum_j x_j f_j(\mathbf{x})$ the average fitness of the population. The change of the population over time is described by

$$\dot{x_i} = x_i[f_i(\mathbf{x}) - \bar{f}(\mathbf{x})], \tag{2.1}$$

where $\dot{x_i}$ is used to denote $\frac{dx_i}{dt}$. Equation 2.1 is the general formulation of the replicator dynamics.

If two populations $\mathbf{x}$ and $\mathbf{y}$ are present in the model then two systems of differential equations are needed, one for player. Let $\mathbf{A}$ and $\mathbf{B}$ be the payoff matrices of the two players, the fitness function can be written as

$$f_i(\mathbf{x}) = \sum_j a_{ij} y_j = (\mathbf{A}\mathbf{y})_i, \tag{2.2}$$

and the average population fitness as

$$\bar{f}(\mathbf{x}) = \sum_i x_i \sum_j a_{ij} y_j = \mathbf{x}^T \mathbf{A}\mathbf{y}, \tag{2.3}$$

the same holds for the other players.

Substituting 2.2 and 2.3 in 2.1 the differential equation of replicator dynamics for two players game can be derived

$$\begin{aligned} \dot{x_i} &= x_i[(\mathbf{A}\mathbf{y})_i - \mathbf{x}^T \mathbf{A}\mathbf{y}] \\ \dot{y_i} &= y_i[(\mathbf{x}^\mathbf{T}\mathbf{B})_i - \mathbf{x}^T \mathbf{B}\mathbf{y}] \end{aligned} \tag{2.4}$$

The replicator dynamics can be extended to case of $\mathcal{N}$ players and n actions per player as follows:

$$\dot{x}_{ia}(t) = x_{ia}(t)\hat{F}_{ia}(t) \tag{2.5}$$

where

$$\hat{F}_{ia} = e_{ia} U_i \prod_{j \neq i} x_j(t) - U_i \prod_j x_j(t). \tag{2.6}$$

$e_{ia} U_i \prod_{j \neq i} x_j(t)$ is the expected payoff when a player is playing his pure strategy while $U_i \prod_j x_j(t)$ represent the expected payoff.

In addition to the Replicator Dynamics, other dynamical equations are studied. We mention, e.g., the Logit, Smith, and BNN. Furthermore, dynamical equations may differ according to the specific game representation to which they are applied. We point an interested reader to Gatti et al. [2013a]; Gatti and Restelli [2016]

2.2.2 Evolutionary Stable Strategies

An evolutionary stable strategy(ESS) is a stability concept that was inspired by the replicator dynamics. However, its definition is of general interest, being applicable regardless the specific dynamic equation one uses. ESS is a mixed strategy that is resistent to invasion by new strategies. Suppose to have a population that is playing a particular strategy then introduce a new population "the invaders" that is playing a different strategy. The original strategy is ESS if it leads to a higher payoff against the resulting combination of the old and the new strategies.

Let $f(\mathbf{x}, \mathbf{y})$ be the fitness of $\mathbf{x}$ against $\mathbf{y}$. The strategy $\mathbf{x}$ is ESS if for every mutant strategy $\mathbf{y}$ the following two properties hold:

- $f(\mathbf{x}, \mathbf{x}) \geq f(\mathbf{y}, \mathbf{x})$ and,
- if $f(\mathbf{x}, \mathbf{x}) = f(\mathbf{y}, \mathbf{x})$, then $f(\mathbf{x}, \mathbf{y}) \geq f(\mathbf{y}, \mathbf{y})$.

The first condition says that an ESS strategy is also a Nash Equilibrium of the original game. The second condition states that if the invading strategy does as well against the original strategy as the original strategy does against itself, then the original strategy must do better against the invader than the invader does against itself.

Every ESS is an asymptotically stable fixed point of the replicator dynamics (Weibull [1997]).

2.3 Multi-Agent Learning

A *Multi-Agent System (MAS)* is a collection of multiple intelligent agents that can cooperate or compete to achieve their personal goals. *Multi-Agent learning (MAL)* is a research field that integrates Machine Learning techniques in Multi-Agent System. The most common is Reinforcement Learning (RL) that is based on the concept of "trial−and−error" (Sutton and Barto [1998]).

Performing an action the agent will receive a reward, that depends on the interaction with other agents and the environment. The reward can be seen as a measure of the goodness of an action, if an action is followed by low reward, in the future the agent may choose to change his policy and select another action in the same situation.

The goal of RL is to find the optimal policy which is the one that maximizes the cumulative reward over long run of the game.

The most important challenge in RL is the tradeoff between *exploration* and *exploitation*, to avoid getting stuck in local optima. The agent must balance between exploiting what is already known and exploring other actions.

Two alternative can be used: *ϵ-greedy* and *softmax*.

- **ϵ-greedy** is based on the idea of behaving greedily most of the time, but once in a while select a random action to make sure the better actions are not missed in the long term. Using this approach the greedy option is chosen with high probability $1-\epsilon$, and with a small probability ϵ a random action is played;

- the **softmax**, or Boltzmann exploration, which uses a temperature parameter τ to balance exploration and exploitation.

$$p_i = \frac{e^{Q(s,a_i)/\tau}}{\sum_j e^{Q(s,a_j)/\tau}} \tag{2.7}$$

  Good actions have an exponentially higher probability of being selected and the degree of exploration is based on the temperature parameter τ.

  For $\tau \to 0$ the agent always acts greedly and choose the strategy that corresponds to the highest Q-value (pure exploitation), while for $\tau \to \infty$ the agent strategy is completely random (pure exploration).

The single−agent reinforcement learning setting can be formalized as *Markov decision process* (MDP).

Definition 10 (Markov process)**.** *A **Markov process** is a tuple* $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mu \rangle$

- $\mathcal{S}$ *is the set of states;*
- $\mathcal{A}$ *is the set of actions;*
- $\mathcal{P}$ *is a state transition probability matrix,* $P(S_{t+1} = s' | S_t = s, A_t = a) = \sum_{r \in R} p(s', r | s, a)$*;*
- $\mathcal{R}$ *is the reward function,* $r(s, a) = E[R_{t+1} | S_t = s, A_t = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in S} p(s', r | s, a)$*;*
- γ *is a discount factor,*$\gamma \in [0, 1]$*. If* $\gamma = 0$ *the agent is "myopic" and maximize only the immediate reward while if* $\gamma = 1$ *the agent has a "far-sighted" evaluation.* γ *can be interpreted as a probability that the process will go on.*
- μ *is the set of initial probability,* $\mu_i^0 = P(X_0 = i)$*.*

The learning goal for an agent is to find the policy π that maps states to action selection probabilities, maximizing the expected reward.

The **value function** V^π denotes the values of being in state s following policy π as the total amount of reward R the agents expects to accumulate when starting in state s and following π:

$$V^\pi(s) = E_\pi[v_t | s_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s] \tag{2.8}$$

Values indicate the long-term desirability of the states. For example, a state might always yield a low immediate reward but still have a high value because it is regularly followed by other states that yield high rewards. Or the reverse can be true.

The optimal value $V^*(s) = \max_\pi V^\pi(s) \quad \forall s \in S$ gives the value of the state given the optimal policy.

A policy is *optimal* if it achieves the best expected return from any initial state.

The value function can be iteratively computed using the *Bellman equation*

$$V^\pi(s) = \sum_{a \in A} \pi(a|s)(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a)V^\pi(s')) \tag{2.9}$$

which expresses the relationship between the value of a state and the value of its successor.

Similarly, it is possible to define the **action value function** $Q^\pi(s,a)$, as the expected reward starting from state s, taking action a and following policy π

$$Q^\pi(s,a) = E_\pi[v_t|s_t = s, a_t = a] = E_{pi}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a]. \tag{2.10}$$

The optimal action-value function can be defined as

$$Q^*(s,a) = \max_\pi q_\pi(s,a) \quad s \in S, a \in A(s) \tag{2.11}$$

V^π and Q^π can be estimated from experience.

2.3.1 Algorithms

In this section the main learning algorithm and their relation with the replicator dynamics will be illustrated.

Cross Learning

One of the most famous learning algorithms is the *Cross Learning* (Daan Bloembergen and Kaisers [2015]).

At each iteration of the game the agent behaviour can be described by the policy $\pi = \{\pi_1, \pi_2, ..., \pi_n\}$, the algorithm updates the policy on the base of the received reward r after taking action j.

$$\pi(i) \leftarrow \pi(i) + \begin{cases} r - \pi(i)r & if i = j \\ -\pi(i)r & otherwise \end{cases} \tag{2.12}$$

The update maintains a valid policy as long as the reward are normalized, i.e $0 \leq r \leq 1$. At each iteration, the probability of selecting an action is increased unless the payoff is exactly equal to 0, the aim is to increase probability of actions that lead to an higher expected payoff.

The behavior of Cross learning converges to the replicator dynamics in the infinitesimal time limit.

Equation 2.12 describes how the probability of taking action i is updated, this probability is updated both if i is selected and if another action j is selected. Let $E[\Delta\pi(i)]$ be the expected change in the policy and $E_i[r]$ be the expected reward of taking action i defined as follows:

$$\begin{aligned} E[\Delta\pi(i)] &= \pi(i)\big[E_i[r] - \pi(i)E_i[r]\big] + \sum_{j \neq i} \pi(j)\big[-E_j[r]\pi(i)\big] \\ &= \pi(i)\big[E_i[r] - \sum_j \pi(j)E_j[r]\big] \end{aligned} \tag{2.13}$$

Assuming to have small steps of update, the continuous time limit of Equation 2.13 is

$$\pi_{t+\delta}(i) = \pi_t(i) + \delta\Delta\pi_t(i)$$

with $\lim_{\delta\to 0}$. This yields a continuous time system that can be expressed with the following partial differential equation

$$\dot{\pi}(i) = \pi(i)\big[E_i[r] - \sum_j \pi(j)E_j[r]\big] \tag{2.14}$$

For two players games Equation 2.14 is equivalent to equation of replicator dynamics 2.4.

Regret Minimization

The notion of Regret Minimization (RM) forms the basis for another type of reinforcement-learning algorithm. Each agent calculates the loss (or regret) l_i of taking action i rather than the best action in hindsight as $l_i = r^* - r$ where r^* is the reward of taking the best actions.

The learner maintains a vector of weights $\mathbf{w}$ for all actions and this vector is updated according to the perceived loss as follows:

$$
\begin{aligned}
w_i &\leftarrow w_i[1 - \alpha l_i] \\
\pi_i &= \frac{w_i}{\sum_j w_j}
\end{aligned}
\tag{2.15}
$$

The algorithm ensures a valid policy until reward are normalized.

It is shown (Klos et al. [2010]) that the infinitesimal time limit of regret minimization can similarly be linked to a dynamical system with replicator dynamics in the numerator:

$$
\dot{x} = \frac{\alpha x_i[(\mathbf{Ay})_i - \mathbf{x^T Ay}]}{1 - \alpha[\max_k(\mathbf{Ay})_k - \mathbf{x^T Ay}]} \tag{2.16}
$$

This dynamic can be extended to the case of $\mathcal{N}$ players and n actions per player

$$
\dot{x_{ia}} = \frac{\alpha x_{ia}(t)\hat{F}_{ia}(t)}{1 - \alpha\big[\max e_{ia} U_i \prod_{j \neq i} x_j(t) - U_i \prod_j x_j(t)\big]} \tag{2.17}
$$

Q-Learning

Another learning algorithm is the *Q-Learning* (Daan Bloembergen and Kaisers [2015]). Q-learning maintains a value function over state-action pairs, $Q(s, a)$, which it updates based on the immediate reward and the discounted expected future reward according to Q:

$$
Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{2.18}
$$

γ is the discount factor for future rewards and $\alpha \in [0, 1]$ is the learning rate that determines how quickly Q is updated based on a new reward information, r is the immediate reward. Q function expresses how good is taking action a in state s.

In 2×2 games the corresponding dynamics is described by a pair of differential equations:

$$
\dot{x}_i = x_i[(\mathbf{Ay})_i - \mathbf{x^T Ay} + T_x \sum_j x_j \log \frac{x_j}{x_i}] \tag{2.19}
$$

where A is the payoff matrices, $\mathbf{x}$ the policy of the agent and T_x the temperature parameter. This equation can be decomposed as the sum of two terms that balance the exploration/exploitation tread off. The first term $x_i[(\mathbf{Ay})_i - \mathbf{x^T Ay}]$ is exactly the replicator dynamics which drives the system to a state with high utility for both players. It can be considered as a selection process. The second therm $x_i T_x \sum_j x_j \log \frac{x_j}{x_i}$ corresponds to the mutation process. Mutation is controlled by the temperature T_x, from RL point of view this term can be seen as an exploration process.

Similarly for the other player it holds

$$\dot{y}_i = y_i \Big[(\mathbf{x}^\mathbf{T}\mathbf{B})_i - \mathbf{x}^\mathbf{T}\mathbf{B}\mathbf{y} + T_y \sum_j y_j \log \frac{y_j}{y_i} \Big]. \tag{2.20}$$

Equation 2.21 describes the Q-Learning dynamics for $\mathcal{N}$ players and n actions per player

$$\dot{x}_{ia}(t) = x_{ia}(t) \Big[\hat{F}_{ia}(t) + T_x \sum_j x_j \log \frac{x_j}{x_i} \Big] \tag{2.21}$$

where $\hat{F}_{ia}(t)$ is defined as in Equation 2.6.

Frequency-Adjusted Q-learning (FAQ) is a variation of Q-Learning dynamics which uses softmax activation function for policy-generation, and an update rule inversely proportional to x_i

$$Q(i) \leftarrow Q(i) + \frac{1}{x_i} \alpha [r + \max_j Q(j) - Q(i)]. \tag{2.22}$$

For two player matrix game FAQ-Learning is described by:

$$\begin{aligned} \dot{x}_i &= x_i \alpha \Big(\tau^{-1} [(\mathbf{A}\mathbf{y})_i - \mathbf{x}^\mathbf{T}\mathbf{A}\mathbf{y}] - \log x_i + \sum_j x_j \log x_j \Big) \\ \dot{y}_i &= y_i \alpha \Big(\tau^{-1} [(\mathbf{x}^\mathbf{T}\mathbf{B})_i - \mathbf{x}^\mathbf{T}\mathbf{B}\mathbf{y}] - \log y_i + \sum_j x_j \log y_j \Big) \end{aligned} \tag{2.23}$$

In a similar way as Q-Learning dynamics we can derive the extension for the case with $\mathcal{N}$ players and n actions per player.[1]

2.3.2 Learning Dynamics

Gradient ascent (or descent) is an optimization technique in the field of Machine Learning.

The learning process follows the direction of the gradient in order to find local optima. This concept can be applied to multi-agent learning by improving the learning agents' policies along the gradient of their payoff

[1] It is known a similar derivation for the case in which the game is represented in extensive form, see Panozzo et al. [2014].

function. This approach assumes that the payoff function, or more precisely the gradient of the expected payoff, is known to the learners.

Consider a two-agent normal form game, let e_i denote the i^{th} unit vector and let n be the number of actions. Gradient ascent is defined using the orthogonal projection function Φ which projects the gradient onto the policy simplex thereby ensuring a valid policy $(\forall \pi_i : 0 \leq \pi_i \leq 1)$

$$\begin{aligned} \Delta\pi_i \leftarrow \alpha\frac{\delta V(\pi,\sigma)}{\delta\pi_i} &= \alpha \lim_{\delta\to 0}\frac{[\pi + \Phi(\delta e_i)]A\sigma^T - \pi A\sigma^T}{\delta} \\ = \alpha\Phi(e_i)A\sigma^T &= \alpha\Big(e_i A\sigma^T - \frac{1}{n}\sum_j e_j A\sigma^T\Big) \end{aligned} \tag{2.24}$$

Infinitesimal Gradient Ascent (IGA)

Each agent updates its policy by taking infinitesimal steps in the direction of the gradient of its expected payoff, Satinder Singh and Mansour [2000]. It has been proven that, in two-player two-action games, IGA converges to a Nash equilibrium.

The policy update rule for IGA is defined as

$$\begin{aligned} \Delta x_i &\leftarrow \alpha\frac{\delta V(x)}{\delta x_i} \\ \mathbf{x} &\leftarrow \mathit{projection}(\mathbf{x} + \Delta x) \end{aligned} \tag{2.25}$$

where α denotes the learning step size.

The intended change Δx may take $\mathbf{x}$ outside of the valid policy space, if this occurs the projection function will project it back to the nearest valid policy.

Win or Learn Fast (WoLF)

Win or learn fast (IGA-WoLF) (Bowling and Veloso [2002]) is a variation of IGA which uses a variable learning rate. The idea is to allow an agent to adapt quickly if it is performing worse than expected, whereas it should be more cautious when it is winning. The modified learning rule of IGA-WoLF is

$$\begin{aligned} \Delta x_i &\leftarrow \frac{\delta V(x)}{\delta x_i}\begin{cases}\alpha_{\min} & if \quad V(x) > V(x^*) \\ \alpha_{\max} & otherwise\end{cases} \\ \mathbf{x} &\leftarrow \mathit{projection}(\mathbf{x} + \Delta x) \end{aligned} \tag{2.26}$$

where x^* a policy belonging to an arbitrary Nash equilibrium.

Weighted Policy Learner (WPL)

Another variation of IGA is the weighted policy learner (WPL) (Abdallah and Lesser [2008]) that also modulates the learning rate, but differently from IGA-WoLF it does not require a reference policy. The update rule of WPL is defined as

$$\begin{aligned} \Delta x_i &\leftarrow \alpha \frac{\delta V(x)}{\delta x_i} \begin{cases} x_i & if \quad \frac{\delta V(x)}{\delta x_i} < 0 \\ 1 - x_i & otherwise \end{cases} \\ \mathbf{x} &\leftarrow projection(\mathbf{x} + \Delta x) \end{aligned} \tag{2.27}$$

where the update is weighted by x_i or by $1 - x_i$ depending on the sign of the gradient.

For two actions two players game the dynamic model can be simplified.

Let $\mathbf{h} = (1, -1)$, $\mathbf{x} = (x, 1-x)$ and $\mathbf{y} = (y, 1-y)$. The pair $(\dot{x}, \dot{y})$ describes the learning dynamics. The simplified version for CL is

$$\begin{aligned} \dot{x} &= x[(\mathbf{Ay})_1 - \mathbf{x^T Ay}] \\ &= x(1-x)[y(a_{11} - a_{12} - a_{21} + a_{22}) + a_{12} - a_{22}] \\ &= x(1-x)[y\mathbf{hAh^T} + a_{12} - a_{22}] \end{aligned} \tag{2.28}$$

where a_{12} and a_{22} are elements of the payoff matrix $\mathbf{A}$.

The shorten notation for gradient δ can be written as

$$\delta = (\mathbf{Ay^T})_1 - (\mathbf{Ay^T})_2 = y\mathbf{hAh^T} + a_{12} - a_{22} \tag{2.29}$$

while the dynamics examined are summarized in the following table:

Algorithm	$\dot{x}$
IGA	$\alpha\delta$
IGA-WoLF	$\delta\begin{cases}\alpha_{\min} & if \quad V(x) > V(x^*)\\ \alpha_{\max} & otherwise\end{cases}$
WPL	$\alpha\delta\begin{cases}x & if \quad \delta < 0\\ 1-x & otherwise\end{cases}$
CL	$x(1-x)\delta$
FAQ	$\alpha x(1-x)[\delta\tau^{-1} - \log\frac{x}{1-x}]$
RM	$\alpha x(1-x)\delta\begin{cases}(1+\alpha x\delta)^{-1} & if \quad \delta < 0\\ (1-\alpha(1-x)\delta)^{-1} & otherwise\end{cases}$

Table 2.1: Learning dynamic of two-agent two-action game, Michael Kaisers and Tuyls [2012].

2.4 Bifurcation Analysis for Mechanism Design

Pilouras in *Bifurcation Mechanism Design — From Optimal Flat Taxes to Better Cancer Treatments* perform a quantitative analysis of bifurcation phenomena connected to Q-learning dynamics in 2×2 games with the goal of quantify the effects of rationaly-driven bifurcations to the social welfare changing the temperature parameter.

He propose two different types of mechanism: hysteresis and optimal control mechanisms.

He starts with the following example:

Example 2 (Hysteresis effect). *Consider a 2×2 coordination game with two pure NEs and no dominant strategy for either players.*

$$A = \begin{pmatrix} 10 & 0 \\ 0 & 5 \end{pmatrix}, B = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix},$$

Given x and Ty [2], the value of y can be uniquely determined. Assuming the system follows the Q-learning dynamics, as we slowly vary Tx, x tends to stay on the line segment that is the closest to where it was originally corresponding to a stable but inefficient fixed point.

[2] T_x and T_y are the temperature parameter of Q-Learning dynamics

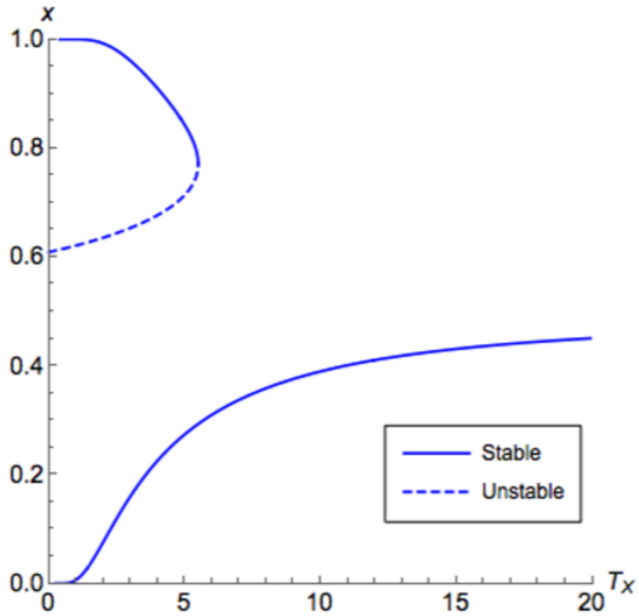


Figure 2.3: Bifurcation diagram for $T_y = 0.5$.

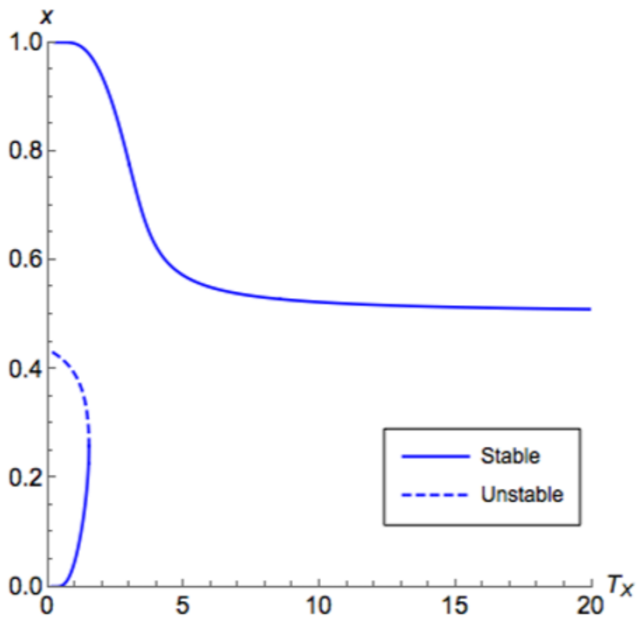


Figure 2.4: Bifurcation diagram for $T_y = 2$.

Figure 2.3 shows the Bifurcation diagram to vary T_x fixed $T_y = 0.5$. The *horizontal axis correspond to the temperature while the vertical shows the probability with which the player plays his first action. There exist three branches, two of which are stable and the other unstable.*

Figure 2.4 shows the bifurcation diagram to vary T_x fixed $T_y = 2$.

Before illustrating the two mechanisms it is necessary to introduce some definitions:

Definition 11 (Quantal response equilibrium)**.** *A strategy profile (x_{QRE}, y_{QRE}) is a QRE with respect to temperature T_x and T_y if*

$$x_{QRE} = \frac{e^{\frac{1}{T_x}(\mathbf{A}\mathbf{y}_{QRE})_1}}{\sum_{j\in\{1,2\}} e^{\frac{1}{T_x}(\mathbf{A}\mathbf{y}_{QRE})_j}} \quad 1 - x_{QRE} = \frac{e^{\frac{1}{T_x}(\mathbf{A}\mathbf{y}_{QRE})_2}}{\sum_{j\in\{1,2\}} e^{\frac{1}{T_x}(\mathbf{A}\mathbf{y}_{QRE})_j}}$$

$$y_{QRE} = \frac{e^{\frac{1}{T_y}(\mathbf{B}\mathbf{x}_{QRE})_1}}{\sum_{j\in\{1,2\}} e^{\frac{1}{T_y}(\mathbf{B}\mathbf{x}_{QRE})_j}} \quad 1 - y_{QRE} = \frac{e^{\frac{1}{T_y}(\mathbf{B}\mathbf{x}_{QRE})_2}}{\sum_{j\in\{1,2\}} e^{\frac{1}{T_y}(\mathbf{B}\mathbf{x}_{QRE})_j}}$$

QRE can be consider as the case where players not only maximize the expected utility but considered also the entropy. QRE are the solutions that maximize the linear combination of the following problem

$$x_{QRE} \in \arg\max_x \left\{ \mathbf{x}^{\mathbf{T}}\mathbf{A}\mathbf{y}_{QRE} - T_x \sum_j x_j \log(x_j) \right\}$$

$$y_{QRE} \in \arg\max_y \left\{ \mathbf{x}^{\mathbf{T}}\mathbf{B}\mathbf{y}_{QRE} - T_y \sum_j y_j \log(y_j) \right\}$$

The *social welfare* provides the performance of a system, it can be defined as the sum of the expected payoff of all agents in the system.

In the context of algorithmic game theory, comparing the best social welfare with the social welfare of equilibrium system states is possible to measure the efficiency of a game. The strategy profile that achieves the maximal social welfare is called the *socially optimal (SO)* strategy profile. The notion of the *price of anarchy (PoA)* and the *price of stability (PoS)* are used to describe the efficiency of a game.

Definition 12. *Given a 2×2 game with payoff matrices A and B,and a set of equilibrium system states $S \subseteq [0, 1]^2$, the price of anarchy (PoA) and the price of stability (PoS) are defined as*

$$PoA(S) = \frac{\max_{(x,y)\in[0,1]^2} SW(x, y)}{\min_{(x,y)\in S} SW(x, y)} \quad PoS(S) = \frac{\max_{(x,y)\in[0,1]^2} SW(x, y)}{\max_{(x,y)\in S} SW(x, y)}$$

The ***hysteresis mechanism*** uses transient changes to the system parameters to induce permanent improvements to its performance via optimal (Nash) equilibrium selection. It reflects a time-based dependence between the system's present output and its past inputs. This mechanism can ensure performance equivalent to the price of stability instead of the price of anarchy.

In Example 2 by sequentially changing T_x, we move the equilibrium state from around $(0, 0)$ to around $(1, 1)$, which is the social optimum state.

Theorem 1 (Hysteresis Mechanism)**.** *Consider a* 2×2 *game that satisfies the following properties*

1. *Its diagonal form satisfies* $a_x, b_x, a_y, b_y > 0$.
2. *Exactly one of its pure Nash equilibria is the socially optimal state.*

Without loss of generality, we can assume $a_x \geq b_x$. *Then there is a mechanism to control the system to the social optimum by sequentially changing* T_x *and* T_y *if (1)* $a_y \geq b_y$ *and (2) the socially optimal state is* $(0, 0)$ *do not hold at the same time.*

How the QRE improves the social welfare is illustrate through an example.

Example 3. *By given the following pair of utility matrices:*

$$A = \begin{pmatrix} \epsilon & 1 \\ 0 & 1 + \epsilon' \end{pmatrix}, B = \begin{pmatrix} 1 + \epsilon & 0 \\ 1 & \epsilon' \end{pmatrix},$$

where ϵ *and* ϵ' *are small number (*$\epsilon > \epsilon' > 0$*). This game presents two PNE in* $(1, 1)$ *and* $(0, 0)$ *with social welfare* $1 + 2\epsilon$ *and* $1 + 2\epsilon'$. *For small* ϵ *and small* ϵ' *the social optimal state is* $(x, y) = (1, 0)$ *with a social welfare of* 2. $(1, 1)$ *is the PNE with the highest SW.*

At PNE, which is the point $T_x = T_y = 0$, *the social welfare is* $1 + 2\epsilon$. *Increasing* T_y *also the SW will increase.*

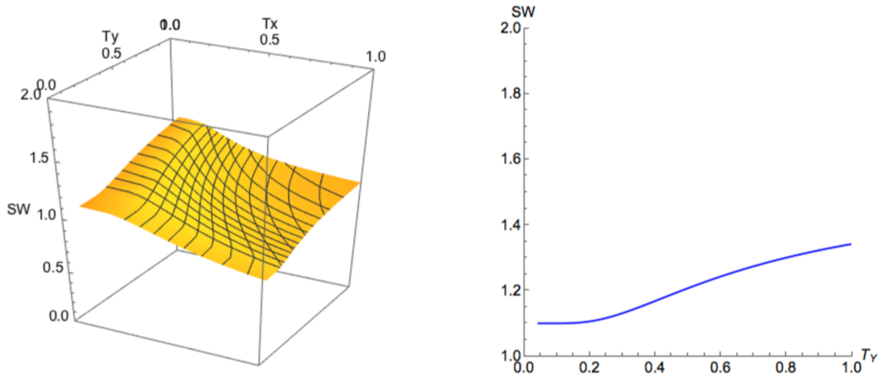


Figure 2.5: Social Welfare.

Optimal control mechanisms induce convergence to states whose performance is better than even the best Nash equilibrium. Controlling the exploration/exploitation trade off is possible to achieve better states than those achievable by perfectly rational agents.

Definition 13. *A state* $(x, y) \in [0,1]^2$ *is a QRE-achievable state if for every* $\epsilon > 0$*, there is a positive finite* T_x *and* T_y *and* (x', y') *such that* $|(x', y') - (x, y)| < \epsilon$ *and* $(x', y') \in QRE(T_x, T_y)$.

The set of QRE-achievable states can be described as

$$S = \left\{ \left\{ x \in \left[\frac{1}{2}, 1\right], y \in \left[\frac{b_x}{b_x + a_x}, 1\right] \right\} \cup \left\{ x \in \left[0, \frac{1}{2}\right], y \in \left[0, \frac{b_x}{b_x + a_x}\right] \right\} \right\}$$
$$\cap \left\{ \left\{ x \in \left[\frac{b_y}{b_y + a_y}, 1\right], y \in \left[\frac{1}{2}, 1\right] \right\} \cup \left\{ x \in \left[0, \frac{b_y}{b_y + a_y}\right], y \in \left[0, \frac{1}{2}\right] \right\} \right\}$$

Theorem 2 (Optimal Control Mechanism)**.** *Given a* 2×2 *game, if it satisfies the following property:*

1. *Its diagonal form satisfies* $a_x, b_x, a_y, b_y > 0$.
2. *None of its pure Nash equilibria is the socially optimal state.*

Without loss of generality, we can assume $a_x \geq b_x$*. Then*

1. *there is a stable QRE-achievable state whose social welfare is better than any Nash equilibrium;*
2. *there is a mechanism to control the system to this state from the best Nash equilibrium by sequentially changing* T_x *and* T_y.

Figure 2.6 shows the set of QRE-achievable states. A point (x, y)represents a mixed strategy profile where the first agent chooses its first strategy with probability x and the second agent chooses its first strategy with probability y. The grey areas depict the set of mixed strategy profiles (x, y) that can be reproduced as QRE states for example i.e., these are outcomes for which there exists temperature parameters (T_x, T_y) for which the (x,y) mixed strategy profile is a QRE.

While Figure 2.7 shows the social welfare. The colour of the point (x, y) corresponds to the social welfare of that mixed strategy profile with states of higher social welfare corresponding to lighter shades. The optimal state is $(1, 0)$, whereas the worst state is $(0, 1)$.

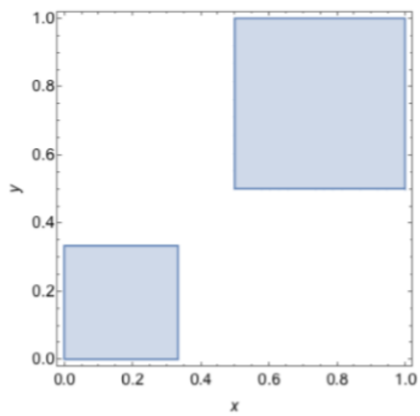


Figure 2.6: Set of QRE-achievable states.

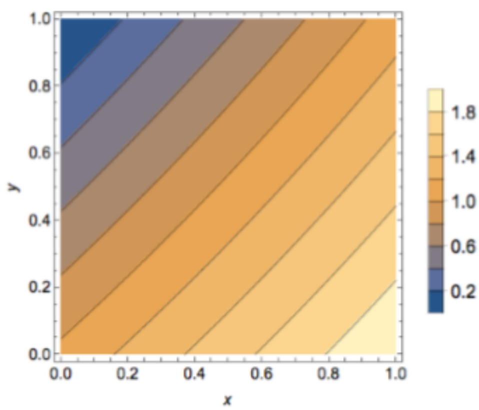


Figure 2.7: Social Welfare.

Chapter 3

Auction Mechanism and Coordination Games

In this chapter we illustrate different types of auctions mechanisms and prove if they can be modeled in terms of coordination games. We are interested in this configuration since we want to apply hysteresis mechanism in order to drive players to a states that is socially optimal.

An auction can be seen as a game where players are the bidders and their actions are the possible bid values. Each player has an intrinsic value for the item being auctioned, that is called *true value*, which corresponds to the maximum value he is willing to pay.

We consider three types of auction mechanisms:

1. **First-Price Sealed-Bid Auction (FPSBA)** simply called *First price*, is the most common and used mechanism;

2. **Generalized Second Price (GSP)** is a generalization of the Second Price sealed-bid auction in which truthfully bidding is not the optimal strategy, it is used because it can lead to a greater revenue;

3. **Repeated GSP with Budget Constrains (RGSPB)**, where each player have a fixed budget and auction is repeated over time.

3.1 Auction Mechanism

3.1.1 First Price

In this auction mechanism, all the players simultaneously submit a "sealed-bid" to the seller and the winner is the one that bids the highest value.

Bid values do not only affect whether the player will win or not but also how much he will have to pay for the item.

Definition 14 (First Price Auction)**.** *Consider the following scenario:*

- $N = \{1, 2, ..., n\}$ *is the set of players;*
- $\Theta = [0, 1]$ *for every player* $i \in N$*;*
- $b = \{b_1, b_2, ..., b_n\}$ *set of bid, one bid per player;*

$$U_i = \begin{cases} \theta_i - b_i & \text{if } i \text{ is the winner} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

N players take part to a single item sealed bid auction. Every player has his own evaluation for the item (θ_i*) but he will bid a value* b_i *lower than* θ_i*, the item is sold to the player with the highest value.* □

Example 4 (First Price Auction)**.** *In this example a single auction with five players and one bid value per player is considered:*

- $N = \{A, B, C, D, E\}$,
- $\Theta = \{20, 43, 37, 77, 80\}$,
- $b = \{10, 33, 15, 31, 52\}$.

Players are sorted by decreasing bid values, $rnk = \{E, B, D, C, A\}$*. The item is sold to player* E *and his utility is equal to* $80 - 52$ *while the other players have an utility equal to* 0. □

Truthfully bidding is not the optimal strategy. Bidding the true value (θ_i) the winner player i will get a zero profit while bidding a value b_i smaller than θ_i he will get a positive utility.

The challenge is to understand how a player should behave in this auction mechanism. If a player bids a value close to his true value he will get a small payoff while bidding a value that is too far from his true value the payoff will increase but at the same time decrease the winning probability.

Finding the tradeoff between these two factors is a complex problem that involves the knowledge of the bid value of other players and their distribution. A general solution is to increase the bid value in relation with the number of players that are involved in the auction. Increasing the number of players, the highest bid is likely to be larger therefore the probability for an agent to win the auction increase if he bid higher.

In a first price auction with two bidders whose values are drawn independently and uniformly in the interval $[0, 1]$ there is a Bayesian-Nash equilibrium when each player bids half of his value.

3.1.2 Generalized Second Price

Generalized Second Price Auction (GSP) is a non-truthful mechanism for multiple items. In online advertising each advertiser is a player, items are the slots where to place adds and the bid value is the price that the player is willing to pay per click and represents his strategy.

Players are ranked on the base of their bid value; the top slot is given to the highest bid, the second to the second highest bidder, and so on. Slot i is assigned to the i^{th} highest bidder at a price per click that depends on the i^{th+1} bidder. The utility of each player is given by the difference between the allocation value and payment

$$u_i = q_i b_i \lambda_i - q_{i+1} b_{i+1} \tag{3.2}$$

Definition 15 (Generalized Second Price Auction)**.** *Problem definition:*

- $N = \{1, 2, ..., n\}$ *is the set of players;*
- $\Theta = [0, 1]$ *for every player* $i \in N$ *represents the set of true values;*
- $b = \{b_1, b_2, ..., b_n\}$ *is the set of bid, one bid per player;*
- $q = [0, 0.1]$ *for every player* $i \in N$ *is the gain of the player when add is clicked;*
- K *is the number of available slot;*
- $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_K\}$ *is the set of discounts associated with the slots values;*

$$U_j = \begin{cases} q_j b_j \lambda_j - q_{j+1} b_{j+1} & \textit{if } j < K \quad j = 1...N \\ q_K b_K \lambda_K & \textit{if } j = K \quad j = 1...N \\ 0 & \textit{if } j > K \quad j = 1...N \end{cases}$$

□

Example 5 (Generalized Second Price Auction)**.** *In this example it is considered the same scenario analyzed for the First Price Auction.*

- $q = \{0.08, 0.0454, 0.0432, 0.0825, 0.0083\}$*;*
- $K = 10$*;*
- $\Lambda = \{1, 0.71, 0.56, 0.53, 0.49, 0.47, 0.44, 0.44, 0.43, 0.43\}$*;*

Player	E	$U_E = -1.0666$
Player	B	$U_B = -1.4938$
Player	D	$U_D = 0.7842$
Player	C	$U_C = -0.4566$
Player	A	$U_A = 0.3920$
Slot	6	
Slot	7	
Slot	8	
Slot	9	
Slot	10	

Table 3.1: GSP Auction mechanism example.

□

In this scenario truthfully bidding is not a dominant strategy, below the proof:

Proof. Suppose to be in the following situation

- $N = 2$ number of player;
- $K = 3$ number of slot;
- b_1, b_2 bid of player i;
- $q_1, q_2 \in [0, 1]$;
- $\lambda_1 = 1$ and $\lambda_2 \in [0, 1]$.

Generally speaking the payments for player i is given by $p_i = \frac{q_{i+1}b_{i+1}}{q_i}$. With two players if $q_1 b_1 \geq q_2 b_2$ payment for player 1 and 2 are respectively $p_1 = \frac{q_2 b_2}{q_1}$ and 0. The displayed configuration is

Player 1
Player 2

Under the assumption of truthfulness ($b_1 = \theta_1$) the expected utility of player 1 is $u_1 = (\lambda_1) q_1 b_1 - q_2 b_2$ and player 1 is allocated in the first slot. Instead under non-truthfulness assumption $b_1 \neq \theta_1, \quad b_1 \ll \theta_1$ player 1 is allocated in the second slot with a utility equal to $u_1 = q_1 b_1 (\lambda_2) - 0$.

GSP is not truthful because player 1 missreports his true value. □

3.1.3 Repeated GSP with budget constraints

In the previous mechanism budget constraints are not considered. In repeated GSP auction each player has a fixed budget and participates to the auction until it runs out. Like in GSP players are displayed according to their bid value, the top slot is assigned to the player with the highest bid value the second to the second highest and so on, player i is allocated on slot i. The utility of each player is given by

$$u_i = \sum_T q_i b_i \lambda_i - \sum_T q_{i+1} b_{i+1} + bgt_i \tag{3.3}$$

where T is the number of repeated auctions and $\sum_T q_{i+1} b_{i+1} = 0$ if player i is allocated in the last slot.

Definition 16 (Repeated GSP with budget constraints)**.** *Problem definition:*

- $N = \{1, 2, ..., n\}$ *is the set of players;*
- $\Theta = [0, 1]$ *for every player* $i \in N$ *represents the set of true values;*
- $b = \{b_1, b_2, ..., b_n\}$ *is the set of bid, one bid per player;*
- $q = [0, 0.1]$ *for every player* $i \in N$ *is the gain of the player when add is clicked;*
- K *is the number of available slot;*
- $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_K\}$ *are the slot values;*
- $BGT = \{bgt_1, bgt_2, ..., bgt_n\}$ *is set of initial Budget;*
- T *is number of auction repetition;*

$$U_j = \begin{cases} \sum_T q_j b_j \lambda_j - \sum_T q_{j+1} b_{j+1} + bgt_j & \text{if } j < K \\ \sum_T q_K b_K \lambda_K + bgt_K & \text{if } j = K \\ bgt_n & \text{if } n > K \end{cases}$$

□

Example 6 (Repeated GSP Auction with budget constraints)**.**

- $T = 100$
- $BGT = \{14.9865; 65.9605; 51.8595; 97.2975; 64.8991\}$*;*

Player	*E*	$U_E = 37.1675$	$res_E = 25.9459$
Player	**B**	$\mathbf{U_B = 27.1223}$	$\mathbf{res_B = -0.5345}$
Player	*D*	$U_D = 117.6867$	$res_D = 80.4495$
Player	*C*	$U_C = 39.8998$	$res_C = 31.0595$
Player	*A*	$U_A = 25.1785$	$res_A = 14.9865$
	Slot 6		
	Slot 7		
	Slot 8		
	Slot 9		
	Slot 10		

Table 3.2: Repeated GSP Auction mechanism: at iteration 26 player B reaches negative residual budget, at the next iteration he leaves the auction.

Player	*E*	$U_E = 35.0416$	$res_E = 23.3884$
Player	*D*	$U_D = 118.8545$	$res_D = 79.8015$
Player	*C*	$U_C = 39.5518$	$res_C = 30.2595$
Player	*A*	$U_A = 25.6025$	$res_A = 14.9865$
Player	**B**	$\mathbf{U_B = 27.1223}$	$\mathbf{res_B = -0.5345}$
	Slot 6		
	Slot 7		
	Slot 8		
	Slot 9		
	Slot 10		

Table 3.3: Repeated GSP Auction mechanism at iteration 27.

□

Repeated GSP is a non-truthfully mechanism, misreporting the bid value a player can gain a better utility. Below the proof:

Proof. Suppose to be in the following situation:

- $N = 2$;
- b_i bid of Player i for $i = 1, 2$;
- θ_i true value of player i;
- $q_i \in [0, 0.1]$ for $i = 1, 2$;
- $\lambda_1 = 1$ and $\lambda_2 \in [0, 1]$ are the slot values.

- B_i budget of player i for $i = 1, 2$.

We analyze the problem form the point of view of Player 1.

Slot 1	$q_1\theta_1\lambda_1$
Slot 2	$q_1 b_1\lambda_1$

Truthfully reporting: $n'_1(q_1\theta_1\lambda_1) - B_1$ where n'_1 is the number of time the player is visualized.

Miss-reporting: $n''_1(q_1 b_1\lambda_1) - B_1$ where n''_1 is the number of time the player is visualized.

To be in the fist slot the player pays more for the allocation therefore he will deplete first the budget we can conclude that $n''_1 \gg n'_1$. □

3.2 Introduction to coordination games

In this analysis' auctions with 2 players and 2 possible values of bid per player are considered. This situation can be seen as 2×2 game where each player can choose one action between two available options.

A and B are the payoff matrices of players A and B respectively

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},$$

where a_{ij} denotes the payoff of player A when he chooses action i and player B action j while b_{ij} denotes the payoff of player B when he chooses action j and player A action i.

The diagonal shape of the payoff matrix

$$A = \begin{pmatrix} a_x & 0 \\ 0 & b_x \end{pmatrix}, B = \begin{pmatrix} a_y & 0 \\ 0 & b_y \end{pmatrix},$$

is a coordination game if $a_x, b_x, a_y, b_y > 0$, where:

- $a_x = a_{11} - a_{21}$,
- $b_x = a_{22} - a_{12}$,
- $a_y = b_{11} - b_{12}$,
- $b_y = b_{22} - b_{21}$.

The aim is to understand under which conditions the First Price, Generalized Second Price and Repeated GSP with budget constraints auction can be modeled as a coordination game.

Under the assumption that overbidding [1] is not allowed, θ represents the true value of the player and b his bid value, therefore $\theta > b$ for both players.

x_1 and x_2 refer to the bid values of player A while y_1 and y_2 to player B. The analysis begins assuming that all bid values are different $x_1 \neq x_2 \neq y_1 \neq y_2$ then the case in which player have equal bid values will be analyzed.

3.3 Auctions analysis

3.3.1 First Price

In the First Price auction the winning player i is the one that bids the highest value and his utility is

$$u_i = \theta_i - b_i \tag{3.4}$$

that is always nonnegative ($\theta \geq b$), while the player who lost gets a utility equal to 0.

Theorem 3. *Given First Price auction with 2 players and 2 bids per players, fixing a_x to be positive, there is no way to satisfy the other condition of coordination game regardless of how bid values are assigned.*

Proof. a_x is positive if the utility value in a_{11} is greater than the utility value in a_{21}. There are three cases in which this condition is satisfied:

1. ($a_{21} = 0$ and $a_{11} > 0$)

   To have $a_{21} = 0$ player B must win the auction and therefore $y_1 > x_2$ and $b_{21} = \theta_2 - y_1$.

   $a_{11} > 0$ if player A wins $x_1 > y_1$ than $a_{11} = \theta_1 - x_1 > 0$ and $b_{11} = 0$. So,

   $$\begin{pmatrix} a_{11} & ? \\ 0 & ? \end{pmatrix}, \begin{pmatrix} 0 & ? \\ b_{21} & ? \end{pmatrix}$$

   where '?' denotes that the value is unknown.

[1] Overbidding occurs when a player submit a bid value higher than his true value ($\theta_i > b_i$)

To satisfy $a_y > 0$ being $b_{11} = 0$, b_{12} must be negative but this is not possible under the assumption that overbidding is not allowed. Therefore, it holds $a_y < 0$ and thus this case does not lead to a coordination game.

Below are the results of the simulation obtained for:

- $\theta = \begin{bmatrix} 0.47 & 0.64 \end{bmatrix}$;
- $bgt = \begin{bmatrix} 0.4046 & 0.4484 \end{bmatrix}$;
- $q = \begin{bmatrix} 0.094 & 0.0646 \end{bmatrix}$;
- $T = 100$.

	0.1	0.05
0.22	0.25,0	0.25,0
0.09	0,0.54	0.38,0

Table 3.4: FP: $a_{21} = 0$ and $a_{11} > 0$.

2. ($a_{21} > 0$ and $a_{11} = 0$)
If $a_{11} = 0$, player B wins the auction $y_1 > x_1$ this means that the only way to satisfy $a_x > 0$ is having $a_{21} < 0$,

$$\begin{pmatrix} 0 & ? \\ a_{21} & ? \end{pmatrix}, \begin{pmatrix} b_{11} & ? \\ 0 & ? \end{pmatrix}$$

thus, this case does not lead to a coordination game.

Below the result of the simulation obtained for:

- $\theta = \begin{bmatrix} 0.77 & 0.8 \end{bmatrix}$;
- $bgt = \begin{bmatrix} 0.7094 & 0.7547 \end{bmatrix}$;
- $q = \begin{bmatrix} 0.0439 & 0.0382 \end{bmatrix}$;
- $T = 100$.

	0.44	0.64
0.18	0,0.36	0,0.16
0.48	0.29,0	0,0.16

Table 3.5: FP: $a_{21} > 0$ and $a_{11} = 0$.

3. ($a_{21} > 0$ and $a_{11} > 0$)

   In this case player A always win against the first action of player B then $x_1 > y_1$ and $x_2 > y_1$. a_x is positive if the difference between a_{11} and a_{21} is nonnegative, therefore it must holds that $a_{11} > a_{21}$. Winning the auction player A utilities in a_{11} and a_{21} are respectively $\theta_1 - x_1$ and $\theta_1 - x_2$. To satisfy $a_{11} > a_{21}$ then the bid value x_1 must be lower than the bid value x_2 . If this condition is satisfied, it holds

$$\begin{pmatrix} a_{11} & ? \\ a_{21} & ? \end{pmatrix}, \begin{pmatrix} 0 & ? \\ 0 & ? \end{pmatrix}$$

   then $a_y > 0$ only if $b_{12} < 0$, but this is not possible since b_{12} is > 0.

   Below the results of the simulation obtained for:

   - $\theta = \begin{bmatrix} 0.23 & 0.93 \end{bmatrix}$;
   - $bgt = \begin{bmatrix} 0.7792 & 0.9340 \end{bmatrix}$;
   - $q = \begin{bmatrix} 0.0254 & 0.0814 \end{bmatrix}$;
   - $T = 100$.

	0.04	0.53
0.19	0.04,0	0,0.4
0.07	0.16,0	0,0.4

Table 3.6: FP: $a_{21} > 0$ and $a_{11} > 0$.

□

3.3.2 Generalized Second Price

The Generalized Second Price (GSP) auction is a non-truthful auction mechanism where the winner player is allocated in the first slot and pay the price bid by the second-highest bidder. If multiple slots are available, the loser player is allocated in the second slot and pay nothing. The utility of the winner player is

$$u_w = q_w b_w \lambda_1 - q_l b_l \tag{3.5}$$

where λ_1 is the slot value, q_w and b_w are respectively the q value and bid value of the winner player while q_l and b_l refer to the loser.

For sake of simplicity first the case in which there is only one available slot is analyzed then the analysis is exetended to the case in which multiple slot are available.

With one slot the loser player is not allocated, and he gets an utility equal to 0, while with multiple slots both players are allocated, the loser one pays nothing and his utility is equal to $u_l = q_l b_l \lambda_2$. Four main scenarios can be distinguished

$$\begin{pmatrix} W_1, L_1 & W_2, L_2 \\ W_3, L_3 & W_4, L_4 \end{pmatrix}, \begin{pmatrix} W_1, L_1 & W_2, L_2 \\ W_3, L_3 & L_4, W_4 \end{pmatrix}, \begin{pmatrix} W_1, L_1 & L_2, W_2 \\ W_3, L_3 & L_4, W_4 \end{pmatrix}, \begin{pmatrix} W_1, L_1 & W_2, L_2 \\ L_3, W_3 & L_4, W_4 \end{pmatrix},$$

W_i indicates that player i wins while L_i that he loses. We refer to the row player as player A while to the column player as player B.

The following case can be discarded

	y_1	y_2
x_1	W_1, L_1	L_2, W_2
x_2	L_3, W_3	W_4, L_4

because the following set of conditions never occurs.

- $W_1 \rightarrow x_1 > y_1$,
- $L_3 \rightarrow x_2 < y_1$,
- $x_1 > x_2$,
- $L_2 \rightarrow x_1 < y_2$,
- $W_4 \rightarrow x_2 > y_2$,
- $x_2 > x_1$.

It is not possible to have at the same time $x_1 > x_2$ and $x_2 > x_1$.

For player A, we focus on the sign of $(W_1 - W_3)$ and on the sign of $(W_4 - W_2)$ while for player B in the sign of $(L_1 - L_2)$ and in the sign of $(L_4 - L_3)$.

The following cell numbering will be used

1	2
3	4

Theorem 4. *Given Generalized Second Price auction with 2 players and 2 actions per player, independently on the number of available slot, there is no way to assign bid value in order to satisfy the coordination games condition.*

Proof. **One slot**

- ***Player A always wins***

	y_1	y_2
x_1	W_1, L_1	W_2, L_2
x_2	W_3, L_3	W_4, L_4

Since player A always wins we have that both his bid values are greater than the bid values of player B. If the value of x_1 is greater than the value of x_2 and since in cell 1 and cell 3 player A will pay the same amount for the allocation is possible to derive that W_1 is greater than W_3. For the same reasoning it follows that W_2 is greater than W_4, this implies that a_x is positive and b_x is negative.

Having only one slot player B is not allocated and therefore a_y and b_y are both equal to 0, thus this is not a coordination game.

An example is:

$$\theta = \begin{bmatrix}0.78 & 0.71\end{bmatrix}; bgt = \begin{bmatrix}0.7441 & 0.5\end{bmatrix}; q = \begin{bmatrix}0.0654 & 0.0494\end{bmatrix}; T = 100.$$

	0.19	0.02
0.33	-0.0242,0	0,0
0.7	0.0	0.0242,0

Table 3.7: GSP one slot: Player A always wins.

- ***Player A wins 3 times***

	y_1	y_2
x_1	W_1, L_1	W_2, L_2
x_2	W_3, L_3	L_4, W_4

To be in this situation the conditions $x_1 > x_2$ and $y_2 > y_1$ must hold. Therefore $W_1 > W_3$ implies that a_x is positive. With only one slot $L_4 = 0$ then the value of b_x is greater than zero if the difference between $q_1 x_1 \lambda_1$ and $q_2 y_2$ is negative, if it is satisfied the sign of a_y and b_y must be analyzed. As $L_1, L_2 = 0$ then $a_y = 0$, while the sign of b_y will depends on the value of W_4.

There are three configurations

$$\begin{pmatrix} W_1, L_1 & W_2, L_2 \\ L_3, W_3 & W_4, L_4 \end{pmatrix} \begin{pmatrix} L_1, W_1 & W_2, L_2 \\ W_3, L_3 & W_4, L_4 \end{pmatrix} \begin{pmatrix} W_1, L_1 & L_2, W_2 \\ W_3, L_3 & W_4, L_4 \end{pmatrix}$$

that can be seen as one rotation of the other. They will not be analyzed, because they lead to the same result.

An example is:

$\theta = \begin{bmatrix} 0.38 & 0.45 \end{bmatrix}; bgt = \begin{bmatrix} 0.7263 & 0.7133 \end{bmatrix}; q = \begin{bmatrix} 0.0412 & 0.0476 \end{bmatrix}; T = 100.$

	0.01	0.1
0.25	0.0091,0	0,0
0.03	0,0	-0.0055,0.0035

Table 3.8: GSP one slot: Player A wins 3 times.

- ***Player A always wins choosing an action and lose choosing the other***

	y_1	y_2
x_1	W_1, L_1	W_2, L_2
x_2	L_3, W_3	L_4, W_4

With only one slot $L_3, L_4 = 0$ then a_x is positive if $q_1 x_1 \lambda_1 - q_2 y_1 > 0$ and $b_x > 0$ if $q_1 x_1 \lambda - q_2 y_2 < 0$. As $L_1, L_2 = 0$ then $a_y = 0$, for this reason it is not possible to have a coordination game.

An example is:

$\theta = \begin{bmatrix} 0.71 & 0.78 \end{bmatrix}; bgt = \begin{bmatrix} 0.0616 & 0.7802 \end{bmatrix}; q = \begin{bmatrix} 0.0526 & 0.0730 \end{bmatrix}; T = 100.$

	0.55	0.39
0.69	-0.0039,0	0,0
0.28	0,0	-0.0078,-0.0117

Table 3.9: GSP one slot: Player A always win choosing one action and lose choosing the other.

- ***Player A always wins against one action on player B and loose against the other***

	y_1	y_2
x_1	W_1, L_1	L_2, W_2
x_2	W_3, L_3	L_4, W_4

The sign of a_x depends on the difference between W_1 and W_3. Knowing that W_1 is equal to $q_1x_1\lambda_1 - q_2y_1$ and W_3 is equal to $q_1x_2\lambda_1 - q_2y_1$ then a_x is positive if the value of x_1 is greater than the value of x_2.

Against the second action of player B player A always loses, his utility is equal to zero this imply that also b_x is equal to zero, therefore we cannot have a coordination game.

An example is:

$$\theta = \begin{bmatrix} 0.91 & 0.86 \end{bmatrix}, bgt = \begin{bmatrix} 0.6646 & 0.9169 \end{bmatrix}, q = \begin{bmatrix} 0.0898 & 0.0836 \end{bmatrix}, T = 100$$

	0.01	0.37
0.17	0.0063,-0.0157	0,0
0.1	0,0	0,0.0220

Table 3.10: GSP one slot: Player A always win against one action and lose against the other.

Situations in which player A always lose, lose three times, always lose against one action and wins against the other or always lose choosing one actions and win choosing the other

$$\begin{pmatrix} L_1, W_1 & L_2, W_2 \\ L_3, W_3 & L_4, W_4 \end{pmatrix} \begin{pmatrix} L_1, W_1 & L_2, W_2 \\ L_3, W_3 & W_4, L_4 \end{pmatrix} \begin{pmatrix} L_1, W_1 & W_2, L_2 \\ L_3, W_3 & W_4, L_4 \end{pmatrix} \begin{pmatrix} L_1, W_1 & L_2, W_2 \\ W_3, L_3 & W_4, L_4 \end{pmatrix}$$

are similar to those just described as it is like examining the same problem but from the point of view of player B.

Multiple slots

Now the analysis is extended to the case in which more than two slots are available. The same scenario as before will be discussed.

- ***Player A always wins***

  As already explained for the case in which only one slot is available, if the value of x_1 is greater than the value of x_2 then a_x is positive while b_x is negative.

From the point of view of the player B, if the value of y_1 is greater than the value of y_2 then the utility in cell 1 (L_1) is greater than the utility in cell 2 (L_2), the same holds for L_3 and L_4. We can conclude that a_y is positive and b_y is negative.

If the value of y_2 is greater than the value of y_1 then $L_2 > L_1$ and $L_4 > L_3$, this entails that a_y is negative and b_y is positive.

If $x_2 > x_1$ and player A wins independently from player B the same results but with opposite sign will be obtained.

An example is:

$$\theta = \begin{bmatrix} 0.78 & 0.71 \end{bmatrix}, bgt = \begin{bmatrix} 0.7441 & 0.5 \end{bmatrix}, q = \begin{bmatrix} 0.0654 & 0.0494 \end{bmatrix}, T = 100$$

	0.19	0.02
0.33	-0.0242,0.006	0,0
0.7	0.0	0.0242,-0.006

Table 3.11: GSP N slot: Player A always wins.

- ***Player A wins 3 times***

  In this situation a_x is positive. The sign of b_x depends on the difference between L_4 and W_2. If it is positive then the sign of a_y and b_y must be verified.

  Since the value of y_1 is smaller than the value of y_2 it follows that L_1 is smaller than L_2 and this implies that $a_y < 0$ and that is not a coordination game.

  An example is:

$$\theta = \begin{bmatrix} 0.38 & 0.45 \end{bmatrix}, bgt = \begin{bmatrix} 0.7263 & 0.7133 \end{bmatrix}, q = \begin{bmatrix} 0.0412 & 0.0476 \end{bmatrix}, T = 100$$

	0.01	0.1
0.25	0.0091,-0.0030	0,0
0.03	0,0	-0.0047,0.0032

Table 3.12: GSP N slot: Player A wins 3 times.

- ***Player A always wins choosing an action and loses choosing the other***

With N slot a_x is positive if the value in cell 1 is greater than the value in cell 3. This happens if $q_1 x_1 \lambda_1 - q_2 y_1 > q_1 x_2$ is satisfied.

$b_x > 0$ if $L_4 > W_2$, this imply that $q_1 x_2 \lambda_2 > q_1 x_1 \lambda_1 - q_2 y_2$. These two conditions depend on the combination of the values of x_1, x_2, y_1, y_2 and q_1, q_2. If both are satisfied (otherwise it is not a coordination game) the sign of a_y and b_y must be control.

Suppose that y_1 is greater than y_2 then $L_1 > L_2$ and this entails that a_y is positive while $W_3 > W_4$ leads to a negative value of b_y.

If the value of y_1 is smaller than y_2 then $L_2 > L_1$ and $W_4 > W_3$, it follows that a_y is negative while b_y is positive, thus this case does not lead to a coordination game.

An example is:

$$\theta = \begin{bmatrix} 0.71 & 0.78 \end{bmatrix}, bgt = \begin{bmatrix} 0.0616 & 0.7802 \end{bmatrix}, q = \begin{bmatrix} 0.0526 & 0.0730 \end{bmatrix}, T = 100$$

	0.55	0.39
0.69	-0.0143,0.0083	0,0
0.28	0,0	0.0026,-0.0117

Table 3.13: GSP N slot: Player A always win choosing one action and lose choosing the other.

- ***Player A always wins against one action of player B and lose against the other***

  With multiple slots if the value of x_1 is greater than the value of x_2 then W_1 is grater than W_3 because in cell 1 and in cell 3 player A will pay the same amount for the allocation, this imply that a_x is positive. Player A lose against the second action of player B and he will be allocated in the second slot without paying, as $x_1 > x_2$ then the value in cell 2 (L_2) is greater than the value in cell 4 (L_4) then b_x is negative.

  Now suppose that x_1 is smaller than x_2, repeating the same reasoning it follows that a_x is negative while b_x is positive.

  In both cases coordination game conditions are not satisfied.

  An example is:

$$\theta = \begin{bmatrix} 0.91 & 0.86 \end{bmatrix}, bgt = \begin{bmatrix} 0.6646 & 0.9169 \end{bmatrix}, q = \begin{bmatrix} 0.0898 & 0.0836 \end{bmatrix}, T = 100$$

	0.01	0.37
0.17	0.0063,-0.0151	0,0
0.1	0,0	-0.0054,0.0214

Table 3.14: GSP N slot: Player A always win against one action and lose against the other.

□

3.3.3 Repeated GSP with budget constraint

Now is considered the case in which GSP auction is repeated multiple time (T).

Each player has a fixed initial budget (bgt) and he takes part in the auction until his residual budget (res) is non negative.

Player are ranked in decreasing order according to their bid values. The winner gets a utility equal to

$$\sum_T q_w b_w \lambda_1 - \sum_T q_l b_l + bgt_w \tag{3.6}$$

until his residual budget is positive. As soon as the residual budget becomes negative, he leaves the auction and his utility will remain constant. The residual budget is updated after each auction

$$res_t = res_{t-1} - q_l b_l$$

where $q_l b_l$ represent the payment for the allocation.

The loser player is allocated in the second slot without paying and his utility is

$$\sum_T q_l b_l \lambda_2 + bgt_l$$

If the winner player leaves the auction the loser will be allocated in the first slot without paying.

Theorem 5. *A 2×2 repeated GSP auction mechanism can be traced to a coordination game only if the two player have the same bid values and parity is broken randomly.*

Proof. In the following analysis to the same scenarios seen for the GSP are discussed.

□

- ***Player A always win***

  In cell 1 and cell 3 and in cell 2 and cell 4 player A will pay the same amount for the allocation, the residual budget is the same and the Player will run out in the same auction repetition.

  Suppose that the value of x_1 is greater than the value of x_2 then $W_1 > W_3$ and $W_2 > W_4$ follows that a_x is positive while b_x is negative. Instead if the value of x_1 is smaller than the value of x_2 then $W_1 < W_3$ and $W_2 < W_4$, this imply a negative value for a_x and a positive value for b_x.

  In both situation we do not have a coordination game.

  An example is:

  $$\theta = \begin{bmatrix}0.78 & 0.71\end{bmatrix}, bgt = \begin{bmatrix}0.7441 & 0.5\end{bmatrix}, q = \begin{bmatrix}0.0654 & 0.0494\end{bmatrix}, T = 100$$

	0.19	0.02
0.33	-1.9358,0.6507	0,0
0.7	0.0	2.4198,-0.6507

Table 3.15: Repeated GSP: Player A always wins.

- ***Player A wins three times***

  Repeating the auction in cell 1 and cell 3 player A always wins and pays the same amount therefore he runs out of budget at the same time t and from time $t+1$ his utility remain constants. Since the value of x_1 is greater than x_2 the difference between W_1 and W_3 is always greater than zero then $a_x > 0$.

  In cell 2 player A sooner or later will consume his budget and his utility will remain constant while in cell 4 he doesn't pay for the allocation and he will never leave the auction this means that L_4 continues to grow. At the end of the game can happen that the value in cell 4 is greater than the value in cell 2, this imply that b_x is positive. If the value in cell 4 is smaller than the value in cell 2 then b_x is negative and this doesn't lead to a coordination game.

  Supposing that $a_x, b_x > 0$ the sign of a_y and b_y must be controlled.

  In cell 1 and cell 2 player B always lose, he doesn't pay for the allocation and he always remain in the auction then both L_1 and L_2 continue to grow. Since the value of y_1 is smaller than the value of y_2 it follows that $L_1 < L_2$ and this imply that a_y is negative. L_3 grows over time while W_4 will remain constant if player B leaves the auction. It is possible to have that $L_3 > W_4$ and in this case $b_y < 0$ or $L_3 < W_4$ and therefore $b_y > 0$, both cases don't lead to a coordination game.

An example is:

$$\theta = \begin{bmatrix} 0.38 & 0.45 \end{bmatrix}, bgt = \begin{bmatrix} 0.7263 & 0.7133 \end{bmatrix}, q = \begin{bmatrix} 0.0412 & 0.0476 \end{bmatrix}, T = 100$$

	0.03	0.25
0.01	-0.2397,-1.0480	0,0
0.1	0.0	0.2631,0.6776

Table 3.16: Repeated GSP: Player A wins 3 times.

- ***Player A always wins choosing an action and lose choosing the other***

  As already explained in the case of GSP with a single auction repetition we are not able to say if $W_1 - L_3 > 0$ or $W_1 - L_3 < 0$. Repeating the auction multiple time in cell 1 player A may terminate his budget and W_1 become constant while L_3 increase. It can happen that $L_3 > W_1 \to a_x < 0$ otherwise $a_x > 0$, the same considerations can also be applied for W_2 and L_4. If $a_x < 0$ or $b_x < 0$ it is not a coordination game, while, if $a_x, b_x > 0$ the sign of a_y and b_y must be controlled.

  Suppose $y_1 > y_2$ then until player A remain in the auction $L_1 > L_2$. In cell 1 player A will leave the game before than in 2 because for the allocation he pays more, therefore it can never happen that $L_2 > L_1$ then a_y is positive. In cell 3 and cell 4 player B pays the same amount hence $W_3 > W_4$ implies $b_y < 0$.
  If $y_2 > y_1$ for the same reasoning as before $L_2 > L_1$ entails $a_y < 0$ and $W_4 > W_3$ implies $b_y > 0$.

  An example is:

$$\theta = \begin{bmatrix} 0.71 & 0.78 \end{bmatrix}, bgt = \begin{bmatrix} 0.0616 & 0.7802 \end{bmatrix}, q = \begin{bmatrix} 0.0526 & 0.0730 \end{bmatrix}, T = 100$$

	0.55	0.39
0.28	1.2541,0.619	0,0
0.69	0.0	-1.223,-1.1695

Table 3.17: Repeated GSP: Player A always win choosing one action and lose choosing the other.

- ***Player A always wins against one action of player B and always lose against the other***

  From player A point of view, we are interested in the difference between value in cell 1 and 3 and the difference between value in cell 2 and 4.

  In cell 1 and cell 3 player A win and pay the same amount given by the bid value of player B and therefore if he exhausts his budget it will happen at the same time t.

  Suppose that $x_1 > x_2$ then the utility in cell 1 is greater than the utility in cell 3, this implies a positive value for a_x while $L_2 > L_4$ entails a negative value for b_x and this is not a coordination game.

  If $x_1 < x_2$ then $W_1 < W_3$ implies $a_x < 0$ while $L_2 < L_4$ implies $b_x > 0$ and this is not a coordination game.

  An example is:

  $$\theta = \begin{bmatrix}0.55 & 0.92\end{bmatrix}, bgt = \begin{bmatrix}0.054 & 0.5308\end{bmatrix}, q = \begin{bmatrix}0.0831 & 0.0585\end{bmatrix}, T = 100$$

	0.56	0.06
0.28	-0.8305,-0.1164	0,0
0.38	0.0	0.133,0.3146

Table 3.18: Repeated GSP: Player A always win against one action and lose against the other.

- ***Players with the same bid values***

  If the bid values are generated randomly the probability that two players bid the same value is zero. However, in practice this event may occurs. In this case it is necessary to define a mechanism to break the parity.

  Parity can be broken randomly and in this case, no predictions on the winning player can be made, or it may be broken in lexicographic order and in this case player A always win.

  Breaking parity in lexicographic order leads to the cases already analyzed therefore we do not have a coordination game.

  An example is:

  $$\theta = \begin{bmatrix}0.71 & 0.78\end{bmatrix}, bgt = \begin{bmatrix}0.7441 & 0.5\end{bmatrix}, q = \begin{bmatrix}0.0494 & 0.0654\end{bmatrix}, T = 100$$

	0.33	0.7
0.7	1.8278,0	0,0
0.33	0.0	-3.4580,4.5780

Table 3.19: Repeated GSP: equal bid values with parity breakage in favour of player A.

If the parity is randomly broken and if $x_1 = y_2$ and $x_2 = y_1$ the condition of coordination game can be satisfied.

$$A = \begin{pmatrix} W_1 & W_2/L_2 \\ W_3/L_3 & L_4 \end{pmatrix}, B = \begin{pmatrix} L_1 & W_2/L_2 \\ W_3/L_3 & W_4 \end{pmatrix}$$

In cell 2 and cell 3 player A sometimes wins and sometimes loses, since is not possible to know a priori how many times parity will be break in favour of player A, there is no way to say if $a_x > 0$ or $a_x < 0$.

Below two simulations with random parity breakage of the same problem where the first one is a coordination game while the second not.

	0.33	0.7
0.7	0.3619,1.3811	0,0
0.33	0.0	1.4040,0.2445

Table 3.20: Repeated GSP: equal bid values with random parity breakage (coordination game).

	0.33	0.7
0.7	0.3619,-1.9982	0,0
0.33	0.0	1.4040,-0.6341

Table 3.21: Repeated GSP: equal bid values with random parity breakage (non coordination game).

3.3.4 Extension to the case n bid values per player

Now the analysis is extended to games with two players and n bid values per player $\{x_1, x_2, ..., x_n\}$ and $\{y_1, y_2, ..., y_n\}$.

$$A = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ \vdots & \vdots & ... & \vdots \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} & ... & b_{1n} \\ b_{21} & b_{22} & ... & b_{2n} \\ \vdots & \vdots & ... & \vdots \\ b_{n1} & b_{n2} & ... & b_{nn} \end{pmatrix}$$

The diagonal shape of these utility matrix is a coordination game if the values along the principal diagonal are all positive.

Corollary 5.1. *An auction mechanism with N players and n bid values is a coordination game if all the 2×2 submatrices satisfy the conditions of coordination game.*

Proof. Instead of analyzing this problem directly it can be broken down into 2×2 subproblems and examine if these can be traced to a coordination game.

Let's extract all possible 2×2 matrices failing on the principal diagonal, if all of them satisfy the coordination game conditions than is possible to conclude that also the original problem is a coordination game.

It has been proven that in 2×2 games there is no way to assign bid values in such a way as to obtain a coordination game, for this reason also the extended problem is not a coordination game. □

3.4 Remarks

The analysis carried out in the previous sections shows that the auction problems studied do not allow coordination game structures. For this reason it is not possible to use the hysteresis mechanism with the aim of increasing social welfare.

Chapter 4

Bifurcation Analysis

In this chapter, the basic concepts of non-linear dynamic systems, stability and bifurcations will be presented.

For this analysis games with two players and two actions per player were considered.

4.1 Basic notions of Dynamics Systems and Bifurcation Theory

A dynamic system is defined as a set of differential equations that evolve over time

$$\dot{x}(t) = f(x(t)) \tag{4.1}$$

x and $\dot{x}$ are n-dimensional vector (the state vector and its time derivative). Given the initial state $x(0)$, the state equations uniquely define a trajectory of the system, i.e., the state vector $x(t)$ for all $t \geq 0$. Trajectories can be found through simulations and are represented in the space as curves starting in $x(0)$ and vector $\dot{x(t)}$ is tangent to the curve at $x(t)$.

One of the most important properties in the study of dynamic systems is stability. In the case of non-linear dynamic systems, stability is studied by linearization approximating the behaviour of the system in the neighborhood of an equilibrium $\bar{x}$.

$$\dot{\delta x}(t) = \frac{\partial f}{\partial x}\Big|_{x=\bar{x}} \delta x(t) \tag{4.2}$$

Stability can be studied looking at the eigenvalues of the Jacobian matrix.

$$J = \frac{\partial f}{\partial x}\Big|_{x=\bar{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \ddots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x=\bar{x}} \tag{4.3}$$

If all eigenvalues $\lambda_i, i = 1, ..., n$ of the Jacobian matrix have negative real part the dynamical system is stable, otherwise if there is at least one eigenvalues with positive real part the dynamical system is unstable.

Bifurcation theory, widely used in the study of dynamic systems, aims at studying of qualitative changes in system dynamics produced by varying parameters.

Definition 17. *In dynamical systems, a bifurcation occurs when a small smooth change of the parameter values (the bifurcation parameters) of a system causes a sudden "qualitative" or topological change in its behaviour. Generally, at a bifurcation, the local stability properties of equilibria, periodic orbits or other invariant sets changes. See Faye [2011] for more details.*

The study of bifurcations is divided into two main classes:

- Local Bifurcation: it involves degeneracy of some eigenvalues of Jacobians associated with equilibria or cycles;
- Global Bifurcation: it cannot be revealed by eigenvalue degeneracies.

The aim of the study of local bifurcation is to analyse the effect of parameter changes on system stability. In this case, the eigenvalues of Jacobian matrix are studied, in particular the sign of their real part.

The *saddle-node* bifurcation is a local bifurcation in which varying a parameter there is the creation or the disappearance of equilibrium points. This phenomenon is also called *fold* or *limit point*.

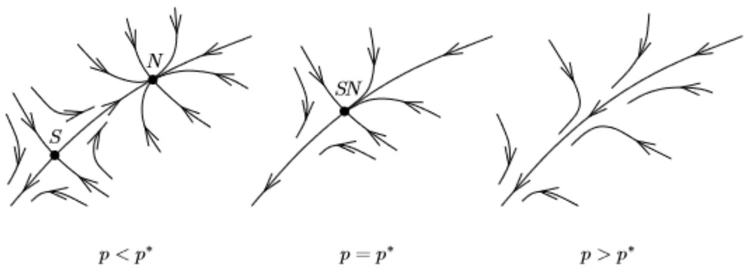


Figure 4.1: Example of local bifurcation: saddle-node bifurcation Dercole and Rinaldi [2011].

This bifurcation can be seen as a collision of two equilibria at $p = p^*$: for $p < p^*$ the two equilibria are distinct, one is stable (node N) and the other is unstable (saddle S). Then, as p increases, the two equilibria approach one each other and finally collide when $p = p^*$.

The eigenvalues evaluated at the saddle are one positive and one negative while the eigenvalues at the node are both negative therefore when they collide one of the two eigenvalues must be equal to 0. For $p > p^*$ no equilibrium is present.

In other words a saddle-node bifurcation can be identified as a change in the sign of one eigenvalues when p varies.

4.2 Application to Auctions

The results of the bifurcation study applied to the auction mechanism are shown below.

The dynamics that are considered are Q-Leaning, FAQ and gradient ascent. Depending on the dynamics, it is possible to vary the values of different parameters.

- Q-Learning: T_x and T_y;
- FAQ: τ and α;
- Gradient Ascent: α.

The standard replicator dynamics (equation 2.4) has not been considered because it does not have parameters that can vary.

The behaviour of the system of dynamic equations as each of the above parameters varies has been analized below.

The dynamic of Q-Learning at varying the temperatures shows different behaviours depending on whether the matrices A and B are in the form of a coordination game or not. The same behaviour is found for the FAQ dynamics. When the value of α changes, the speed at which the system reaches the equilibrium changes without affecting the stability.

Then the system has been analysed varying player's payoff.

4.2.1 Temperature parameter

The behaviour of the system when the temperature varies depends on the structure of the matrices **A** and **B**. If the conditions for a coordination game are not met, the dynamical system remains stable, it has only one equilibrium point and does not show bifurcations regardless of the mechanism by which these matrices were extracted.

Results

- $bid = \begin{bmatrix} 0.08 & 0.4 \\ 0.72 & 0.35 \end{bmatrix}$,
- $q = \begin{bmatrix} 0.0616 & 0.0513 \end{bmatrix}$,
- $bgt = \begin{bmatrix} 0.9744 & 0.3739 \end{bmatrix}$,
- $T = 500$,
- $\mathbf{A} = \begin{bmatrix} -9.8559 & 0 \\ 0 & -1.9884 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1.4428 & 0 \\ 0 & 8.4958 \end{bmatrix}$,
- $\alpha = 0.1$,
- FAQ temperature $\tau = 1$,
- Q-Learning temperature $temp = \begin{bmatrix} 1 & 2 \end{bmatrix}$.

Evolutionary Dynamics

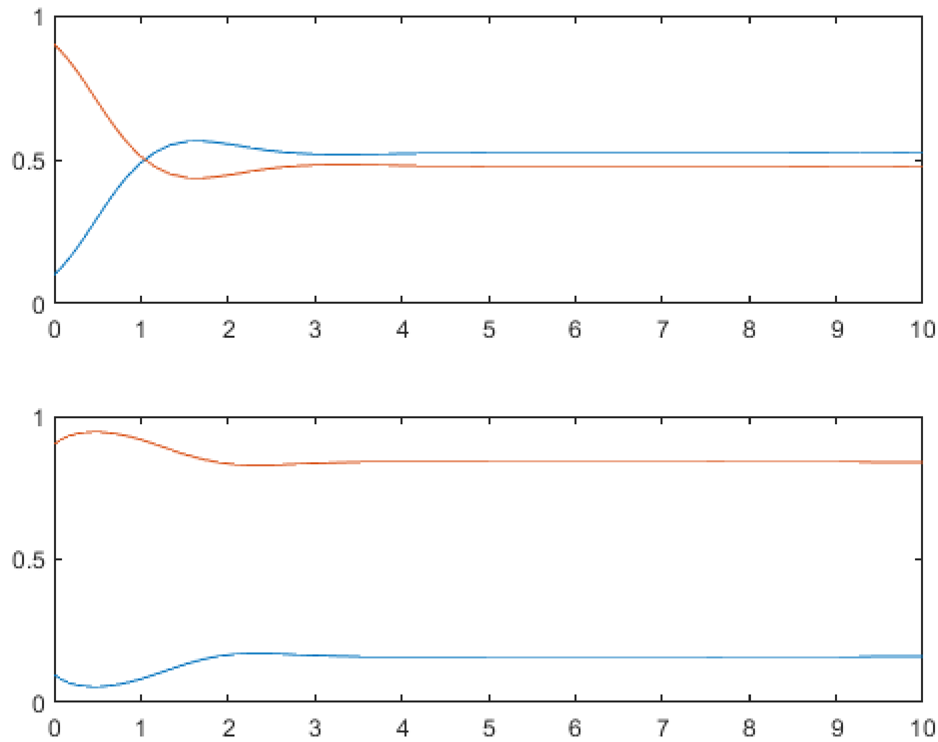


Figure 4.2: Q-Learning Evolutionary Dynamics.

As the dynamics shows the systems converges to $(0.52, 0.16)$. This point is selected for the computation of equilibrium curve (see Figure 4.4).

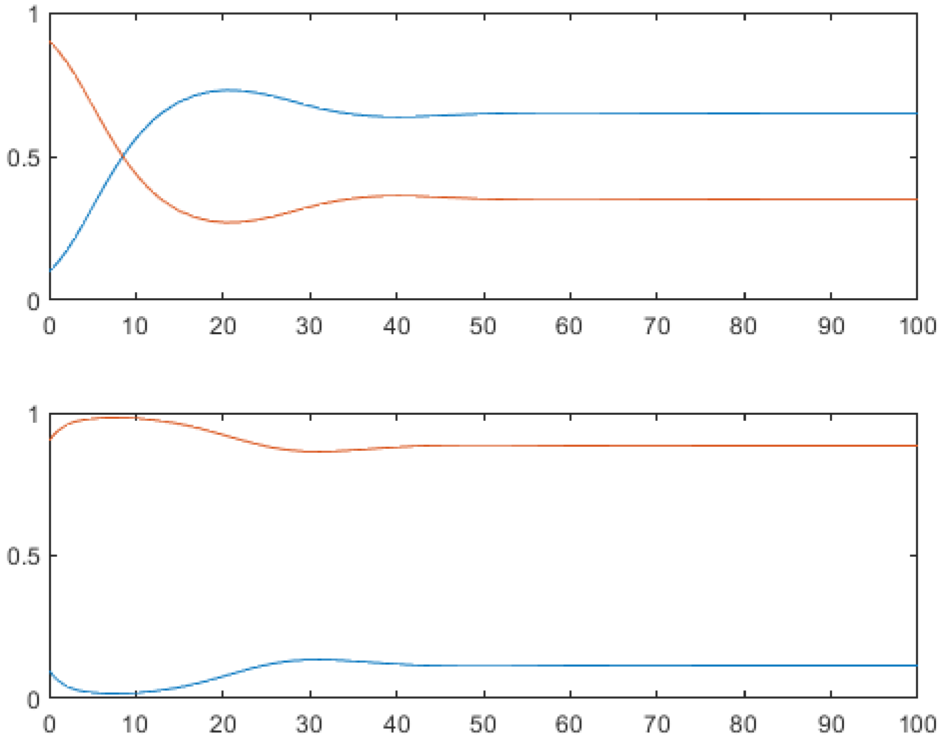


Figure 4.3: FAQ Evolutionary Dynamics for $\alpha = 0.1$.

Equilibrium is at $(0.65, 0.11)$.

Bifurcation

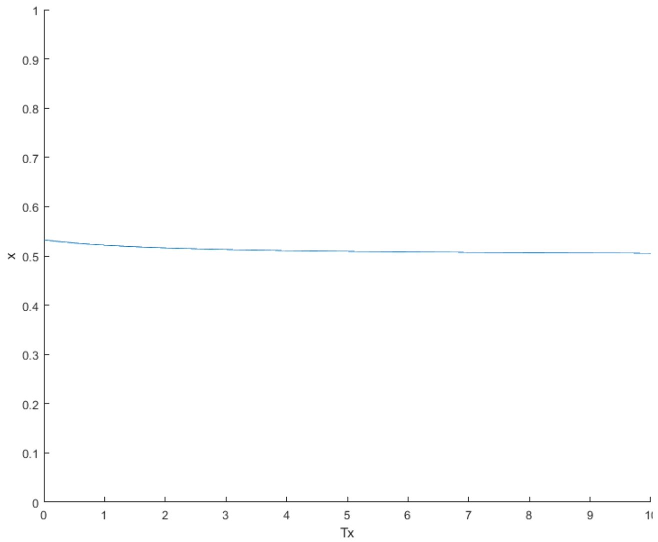


Figure 4.4: Q-Learning Bifurcation diagram for non coordination games.

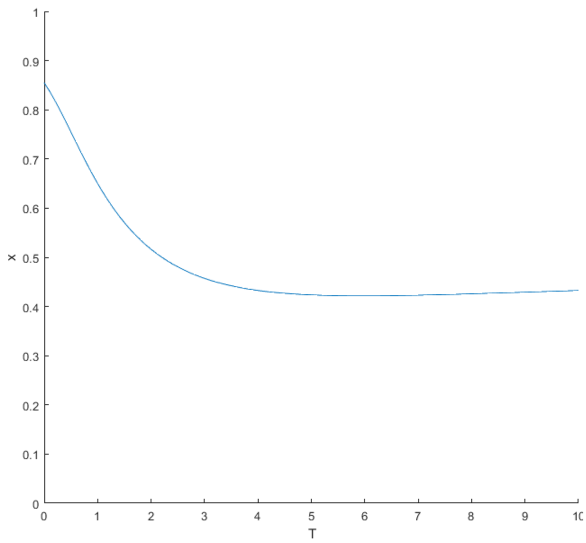


Figure 4.5: FAQ Bifurcation diagram for non coordination games.

The figures 4.4 and 4.5 show how the system reacts to temperature changes. In both cases this parameter has no effect on the stability of the system. Eigenvalues tend to be 0 without ever cancelling.

A different behaviour is obtained if $\mathbf{A}$ and $\mathbf{B}$ meet the conditions of having a coordination game. In the previous chapter we have shown that it is possible to have a coordination game only in the case where two players have equal bid values. Under these conditions, as the temperature changes, we see a change in the stability of the system reaching a *limit point*.

Results

- $bid = \begin{bmatrix} 0.18 & 0.52 \\ 0.52 & 0.18 \end{bmatrix}$,
- $q = \begin{bmatrix} 0.0491 & 0.0527 \end{bmatrix}$,
- $bgt = \begin{bmatrix} 7.399 & 6.576 \end{bmatrix}$,
- $\theta = \begin{bmatrix} 0.82 & 0.94 \end{bmatrix}$,
- $T = 2000$,
- $\alpha = 0.1$,
- FAQ temperature $\tau = 1$,
- Q-Learning temperature $temp = \begin{bmatrix} 1 & 2 \end{bmatrix}$.

$$\mathbf{A} = \begin{bmatrix} 12.0482 & 0 \\ 0 & 3.5887 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 8.4256 & 0 \\ 0 & 12.0822 \end{bmatrix}$$

In this situation the system shows two equilibrium points at $(0.0282, 0.0031)$ and $(0.999, 0.985)$, this suggests that bifurcation points may occurs.

Evolutionary Dynamics

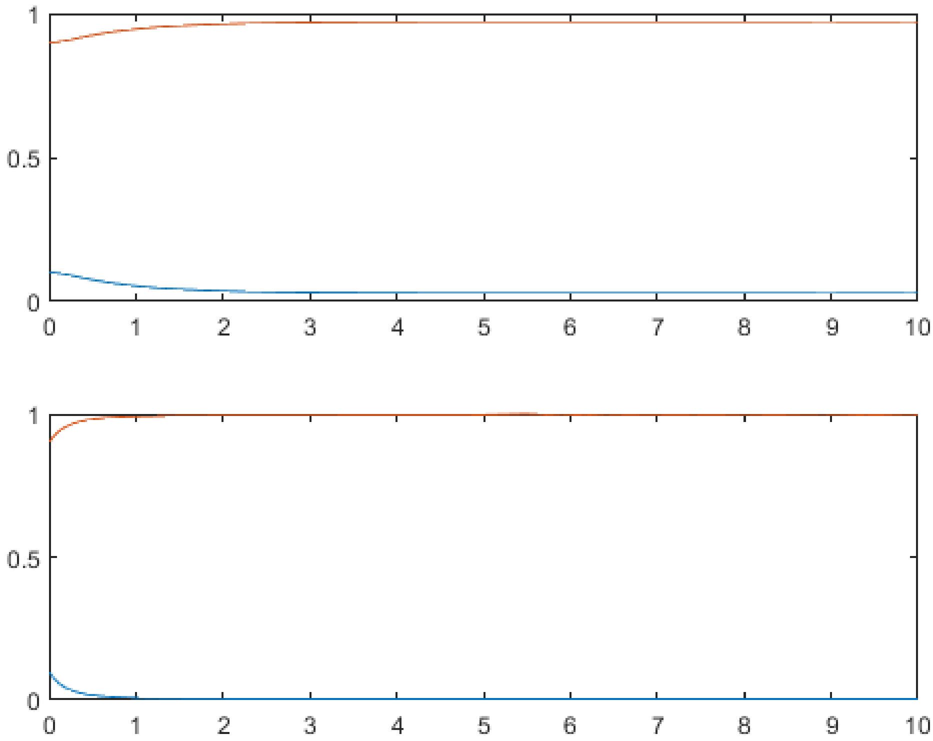


Figure 4.6: Q-Learning Evolutionary Dynamics for Coordination games.

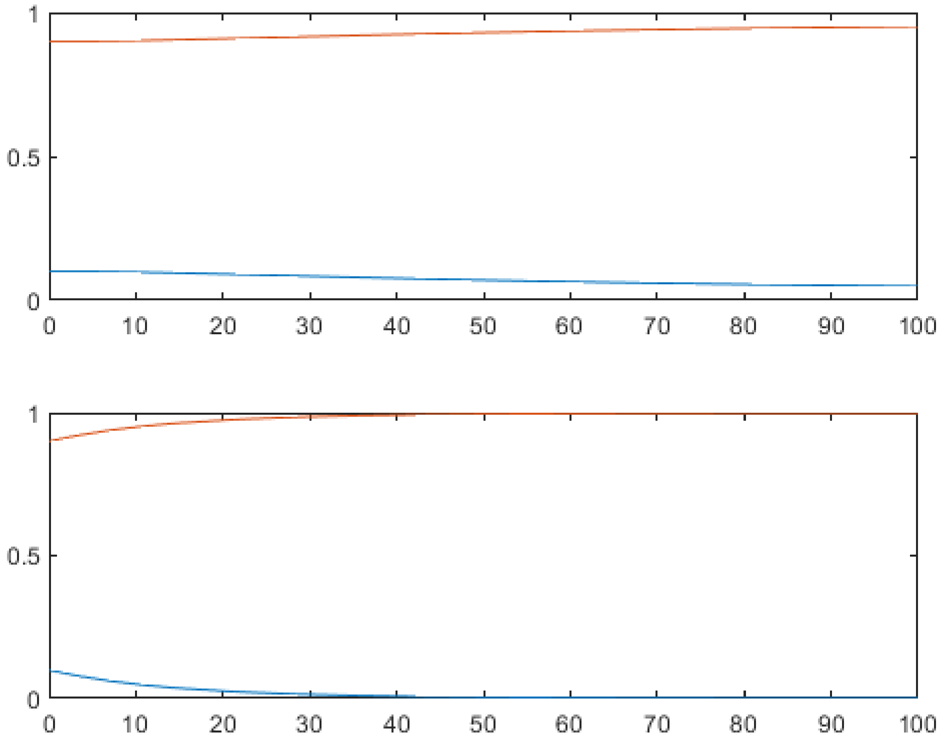


Figure 4.7: FAQ Learning Evolutionary Dynamics for Coordination games.

Bifurcation

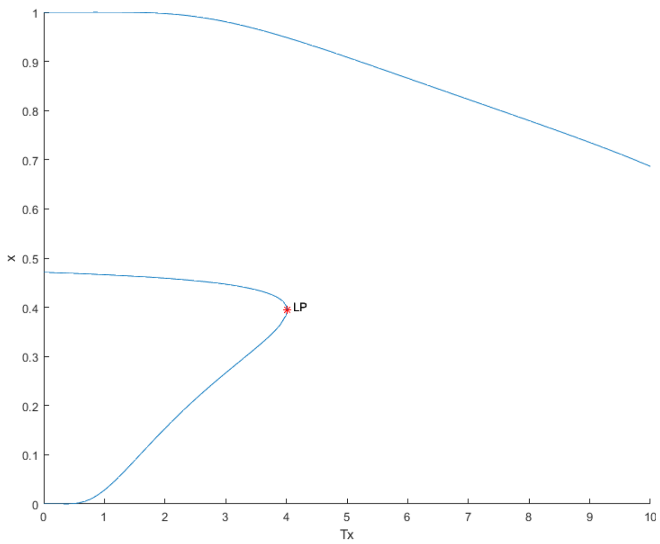


Figure 4.8: Q-Learning Bifurcation diagram for coordination games.

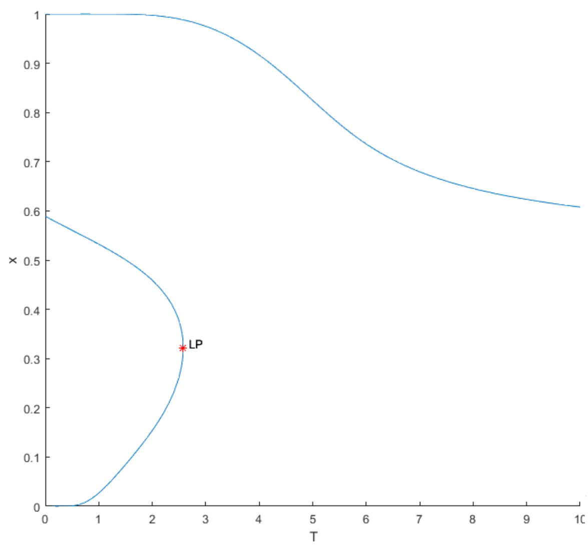


Figure 4.9: FAQ Learning Bifurcation diagram for coordination games.

Q-Learning dynamics shows a LP at $(0.39, 0.12)$ while FAQ dynamics at $(0.32, 0.1)$. Starting form the initial condition and increasing the temperature the system remains stable until the LP is reached, than it becomes unstable when temperature starts decreasing. The other branch is always stable.

The same behaviour occurs with variations of T_y.

4.2.2 Learning rate α

In FAQ dynamics (Equation 2.23) and Gradient Ascent (Equation 2.24) the learning rate can vary therefore the system may be affected by this parameter.

Observing the structure of this learning dynamics it is possible to derive that this parameter does not affect the stability of the system but only the speed with which it reaches the equilibrium. This behaviour occurs regardless of whether the matrices, obtained from the auction mechanism, under analysis meet the conditions for having a coordination game or not.

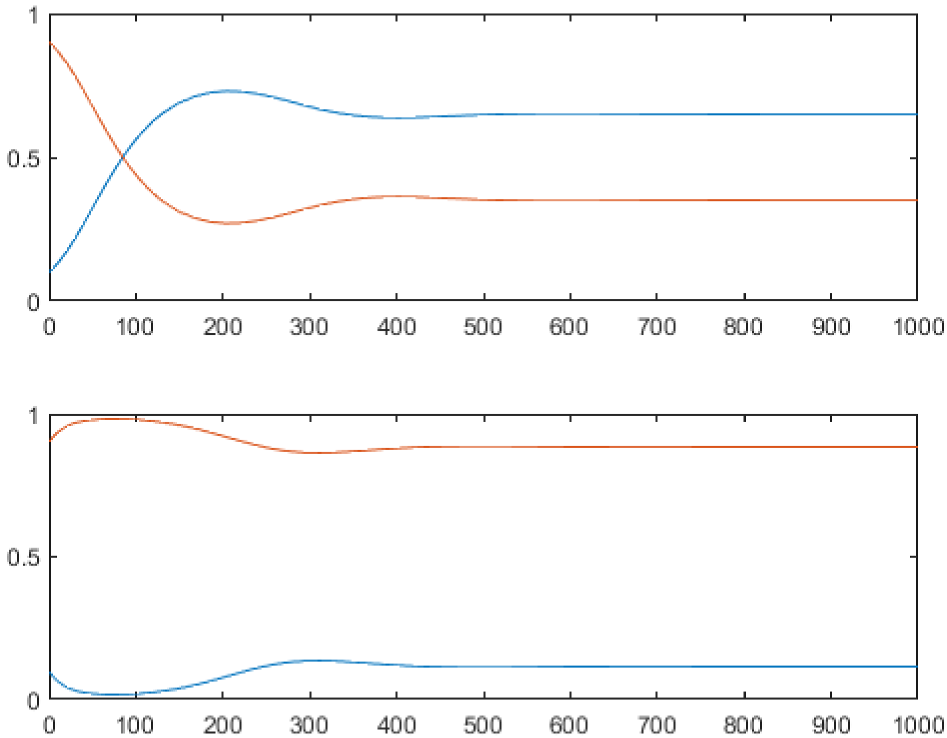


Figure 4.10: FAQ Evolutionary Dynamics for $\alpha = 0.01$.

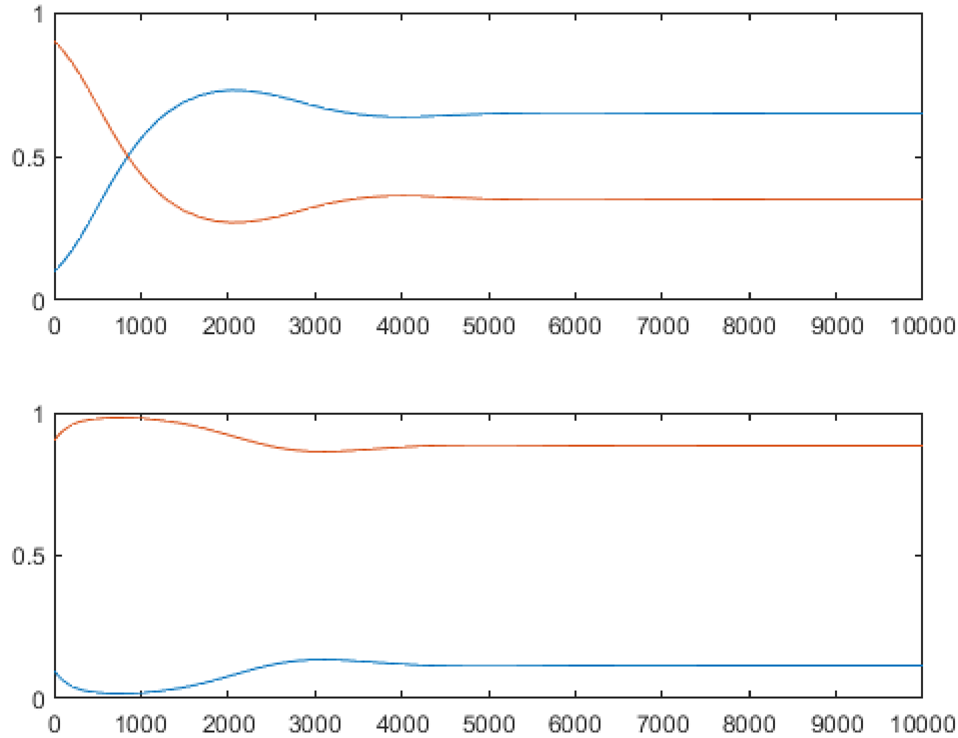


Figure 4.11: FAQ Evolutionary Dynamics for $\alpha = 0.001$.

From figure 4.3, 4.10 and 4.11 we can see how the dynamics is influenced by the learning rate.

Theorem 6. *In FAQ and Gradient Ascent dynamics when the learning rate changes it does not influence the stability of the system but only the speed with which it reaches the equilibrium.*

Proof. The set of differential equation is given by the pair:

$$\dot{x_i} = x_i \alpha f_i(\mathbf{x})$$

$$\dot{y_i} = y_i \alpha g_i(\mathbf{y})$$

The Jacobian matrices are

$$J_A = \frac{\partial f}{\partial x} = \begin{bmatrix} \alpha \frac{\partial f_1}{\partial x_1} & \alpha \frac{\partial f_1}{\partial x_2} \\ \alpha \frac{\partial f_2}{\partial x_1} & \alpha \frac{\partial f_2}{\partial x_2} \end{bmatrix} \qquad det(J_A) = \alpha^2(...) = 0 \tag{4.4}$$

$$J_B = \frac{\partial g}{\partial y} = \begin{bmatrix} \alpha\frac{\partial g_1}{\partial y_1} & \alpha\frac{\partial g_1}{\partial y_2} \\ \alpha\frac{\partial g_2}{\partial y_1} & \alpha\frac{\partial g_2}{\partial y_2} \end{bmatrix} \quad det(J_B) = \alpha^2(...) = 0 \tag{4.5}$$

Since α can assume a value between 0 and 1 from equations 4.4 and 4.5 it follows that the sign of the eigenvalues does not depend on this parameter so when α varies the system is not disrupted and it does not shows bifurcations.

□

Figure 4.12 shows the speed with which the system reaches equilibrium as α changes. The more α tends to 1 the faster the system reaches the equilibrium.

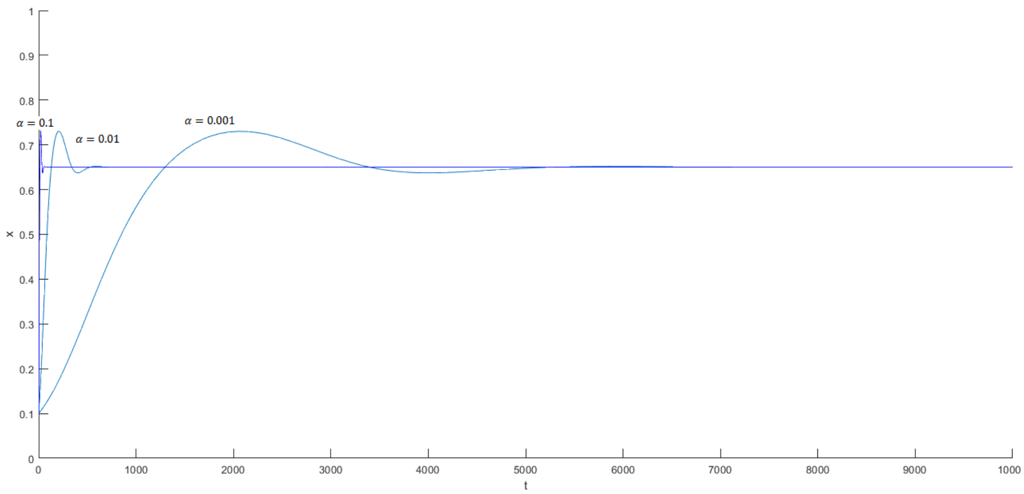


Figure 4.12: FAQ learning rate.

4.2.3 Payoff linear combination

This solution has been used for the particular case in which the two players have the same bid values. The system receives as input four matrices: the first couple $(\mathbf{m}, \mathbf{n})$ is obtained breaking the parity in favour of the first player while the second pair breaking the parity in favour of the seconds $(\mathbf{M}, \mathbf{N})$. Matrices $\mathbf{A}$ and $\mathbf{B}$ are obtained by a linear combination of the above:

$$\begin{aligned} \mathbf{A} &= \mathbf{m} + \epsilon\mathbf{M} \\ \mathbf{B} &= \mathbf{n} + \epsilon\mathbf{N} \end{aligned} \tag{4.6}$$

In this way applying the Q-Learning or FAQ dynamics in addition to the temperature, the bifurcation study can be carried out as the ϵ varies.

As showed in the previous chapter the only way to have a coordination game is when the two players have the same bid values and parity is randomly broken. Breaking it in favour of one of the two players falls into the scenario where a player wins three times, for this reason both pairs of matrices don't satisfy the conditions to have a coordination game.

Looking at the structure of these matrices it is possible to notice how changing the parity break in favour of one player rather than the other the matrices payoff change their signs.

In the linear combination, varying ϵ between 0 and 1 means that one of the two matrices have a lower weight, so that it does not cause any changes in the game structure.

The system remains stable and does not show any bifurcation points.

When the temperature varies, the observed behaviour is the same for non coordination games.

Results

- $bid = \begin{bmatrix} 0.7 & 0.33 \\ 0.33 & 0.7 \end{bmatrix}$,
- $\theta = \begin{bmatrix} 0.71 & 0.78 \end{bmatrix}$,
- $q = \begin{bmatrix} 0.0494 & 0.0654 \end{bmatrix}$,
- $bgt = \begin{bmatrix} 0.7441 & 0.5 \end{bmatrix}$,
- $T = 300$,
- Parity broken in favour of player A
  $\mathbf{m} = \begin{bmatrix} 0.6397 & 0 \\ 0 & 4.9344 \end{bmatrix} \quad \mathbf{n} = \begin{bmatrix} -7.3275 & 0 \\ 0 & -5.3417 \end{bmatrix}$,
- Parity broken in favour of player B
  $\mathbf{M} = \begin{bmatrix} -4.2891 & 0 \\ 0 & -5.4796 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 6.0875 & 0 \\ 0 & 0.7501 \end{bmatrix}$,
- $\alpha = 0.1$,
- FAQ temperature $\tau = 1$,
- Q-Learning temperature $temp = \begin{bmatrix} 1 & 2 \end{bmatrix}$.

Linear combination

$$\mathbf{A} = \begin{bmatrix} 0.6397 & 0 \\ 0 & 4.9344 \end{bmatrix} + \epsilon \begin{bmatrix} -4.2891 & 0 \\ 0 & -5.4796 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -7.3275 & 0 \\ 0 & -5.3417 \end{bmatrix} + \epsilon \begin{bmatrix} 6.0875 & 0 \\ 0 & 0.7501 \end{bmatrix}$$

ϵ can vary between 0 and 1, the value of 0.1 has been chosen as the initial condition.

$$\mathbf{A} = \begin{bmatrix} 0.2108 & 0 \\ 0 & 4.3864 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} -6.7188 & 0 \\ 0 & -5.2667 \end{bmatrix}$$

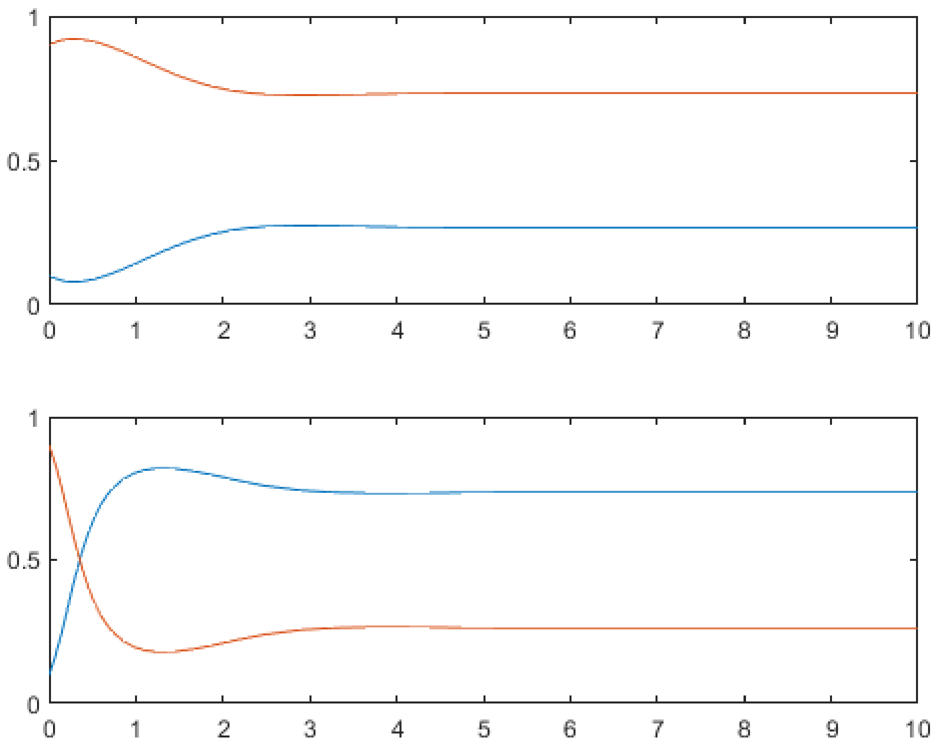


Figure 4.13: Dynamics.

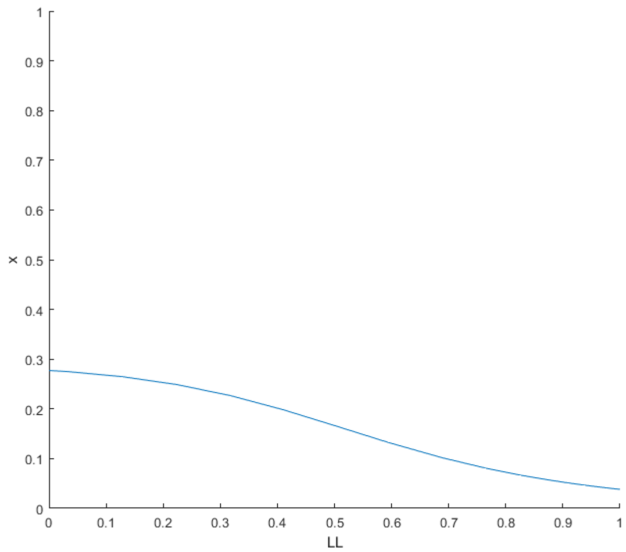


Figure 4.14: Bifurcation for ϵ.

4.2.4 Varying q in GSP Auction Mechanism

Using the GSP auction mechanism is possible to study the behaviour of the system as q varies. In this case inputs are the bid values of the two player (b_1, b_2 for players A and b_3, b_4 for player B), q_1 and q_2.

The aim of this study is observing how the dynamical system evolves as the payoff vary.

We consider now the same scenario considered in section 3.3.2. Attention is focused on how payoffs are computed instead of their values.

1. ***Player A always wins***

$$
\begin{aligned}
\mathbf{A} &= \begin{bmatrix} q_1b_1\lambda_1 + q_2b_3 & q_1b_1\lambda_1 + q_2b_4 \\ q_1b_2\lambda_1 + q_2b_3 & q_1b_2\lambda_1 + q_2b_4 \end{bmatrix} \\
&= \begin{bmatrix} q_1b_1\lambda_1 + q_2b_3 - q_1b_2\lambda_1 - q_2b_3 & 0 \\ 0 & q_1b_2\lambda_1 + q_2b_4 - q_1b_1\lambda_1 bq_2b_4 \end{bmatrix} \\
&= q_1\lambda_1 \begin{bmatrix} (b_1 - b_2) & 0 \\ & (b_3 - b_4) \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} q_2b_3\lambda_2 & q_2b_4\lambda_2 \\ q_2b_3\lambda_2 & q_2b_4\lambda_2 \end{bmatrix} \\
&= \begin{bmatrix} q_2b_3\lambda_2 - q_2b_4\lambda_2 & 0 \\ 0 & q_2b_4\lambda_2 - q_2b_3\lambda_2 \end{bmatrix} \\
&= q_2\lambda_2 \begin{bmatrix} (b_3 - b_4) & 0 \\ 0 & (b_4 - b_3) \end{bmatrix}
\end{aligned}
\tag{4.7}
$$

2. ***Player A wins two times***

   Suppose that $b_1 > b_3; b_1 > b_4$ e $b_2 < b_3; b_2 < b_4$ matrices payoff $\mathbf{A}$ e $\mathbf{B}$ will be:

$$
\begin{aligned}
\mathbf{A} &= \begin{bmatrix} q_1b_1\lambda_1 - q_2b_3 & q_1b_1\lambda_1 - q_2b_4 \\ q_1b_2\lambda_2 & q_1b_2\lambda_2 \end{bmatrix} \\
&= \begin{bmatrix} q_1(b_1\lambda_1 - b_2\lambda_2) - q_2b_3 & 0 \\ 0 & q_1(b_2\lambda_2 - b_1\lambda_1) - q_2b_4 \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} q_2b_3\lambda_2 & q_2b_4\lambda_2 \\ q_2b_3\lambda_1 - q_1b_2 & q_2b_4\lambda_1 - q_1b_2 \end{bmatrix} \\
&= q_2\lambda_2 \begin{bmatrix} b_3 - b_4 & 0 \\ 0 & b_4 - b_3 \end{bmatrix}
\end{aligned}
\tag{4.8}
$$

   If player A always wins against one action of player B and lose against the other:

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} q_1 b_1 \lambda_1 - q_2 b_3 & q_1 b_1 \lambda_2 \\ q_1 b_2 \lambda_1 - q_2 b_3 & q_1 b_2 \lambda_2 \end{bmatrix} \\
&= \begin{bmatrix} q_1 \lambda_1 (b_1 - b_2) & 0 \\ 0 & q_1 \lambda_2 (b_2 - b_1) \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} q_2 b_3 \lambda_2 & q_2 b_4 \lambda_1 - q_1 b_1 \\ q_2 b_3 \lambda_2 & q_2 b_4 \lambda_1 - q_1 b_2 \end{bmatrix} \\
&= \begin{bmatrix} q_2 (b_3 \lambda_2 - b_4) + q_1 b_1 & 0 \\ 0 & q_2 (b_4 \lambda_1 - b_3 \lambda_2) - q_1 b_2 \end{bmatrix}
\end{aligned} \tag{4.9}$$

3. ***Player A wins three times***

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} q_1 b_1 \lambda_1 - q_2 b_3 & q_1 b_1 \lambda_1 - q_2 b_4 \\ q_1 b_2 \lambda_1 - q_2 b_3 & q_1 b_2 \lambda_2 \end{bmatrix} \\
&= \begin{bmatrix} q_1 \lambda_1 (b_1 - b_2) & 0 \\ 0 & q_1 (b_2 \lambda_2 - b_1) - q_2 b_4 \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} q_2 b_3 \lambda_2 & q_2 b_4 \lambda_2 \\ q_2 b_3 \lambda_2 & q_2 b_4 \lambda_1 - q_1 b_2 \end{bmatrix} \\
&= \begin{bmatrix} q_2 \lambda_2 (b_3 - b_4) & 0 \\ 0 & q_2 (b_4 \lambda_1 - b_3 \lambda_2) - q_1 b_2 \end{bmatrix}
\end{aligned} \tag{4.10}$$

From matrices in equations 4.8, 4.9 and 4.10 is possible to notice that **A** and **B** show the same structure. Since q_1 and q_2 assume a value between 0 and 0.1 these parameters don't affect the stability of the system which maintains only one equilibrium point. For this reason, no bifurcation will be detected.

Chapter 5

Optimal Control Mechanism

The aim of this chapter is to illustrate a mechanism that improves the Social Welfare (SW) controlling the temperature parameters.

5.1 Nash Equilibria and Social Optimal

As already explained in section 2.1.3, the NE is the state where each player has no incentive to change his strategy.

In auctions problem NE may not coincide with the Social Optima (SO) which is the state where the Social Welfare is maximum.

$$SW(x,y) := \sum_{j=1}^{n} p_j(x,y) \tag{5.1}$$

For a 2×2 game with payoff matrices $\mathbf{A}, \mathbf{B}$, the social welfare is given by:

$$SW(x,y) = xy(a_{11}+b_{11})+x(1-y)(a_{12}+b_{21})+y(1-x)(a_{21}+b_{12})+(1-x)(1-x)(a_{22}+b_{22})$$

where x is the probability with which Player A plays his first action and y is the probability with which B plays his first action.

The social optimal state is given by the pair (x,y) which maximizes the social welfare $SW(x,y)$.

$$SO = \max_{(x,y)} SW(x,y) \tag{5.2}$$

Our goal is to control the parameters of the learning process to drive the system to a state (x,y) that is QRE-Achievable with a SW greather than the Nash.

$$SW(x,y) > SW(x_{NE}, y_{NE})$$

where x_{NE}, y_{NE} are the probabilities of being at the NE.

Definition 18. *A state $(x, y) \in [0,1]^2$ is a QRE-Achievable state if for every $\epsilon > 0$, there is a positive finite T_x and T_y and (x', y') such that $|(x', y') - (x, y)| < \epsilon$ and $(x', y') \in QRE(T_x, T_y)$. See Ger Yang and Piliouras [2018] for more details.*

If the Nash Equilibrium coincides with the Social Optimal starting from any initial state we are always able to reach the SO decreasing the temperature parameters to zero which means that both player are playing rationally. An example below:

Example 7.

$$\begin{bmatrix} -5.57e-04, -0.0406 & 0, 0 \\ 0, 0 & 5.57e-04, 0.046 \end{bmatrix}$$

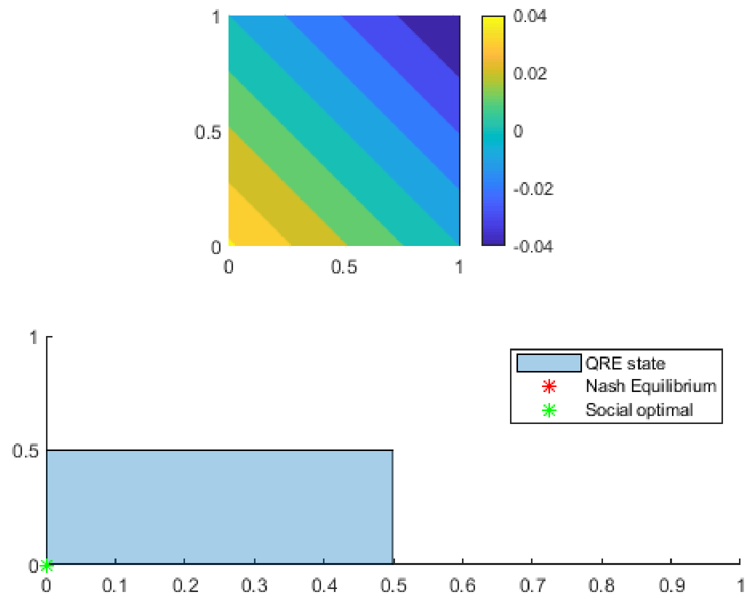


Figure 5.1: Social Optimal coincides with Nash Equilibrium.

On the top of figure 5.1 we can see the trend of the Social Welfare changing x and y. The bottom one shows the position of the NE and SO and the set of QRE-Achievable state.

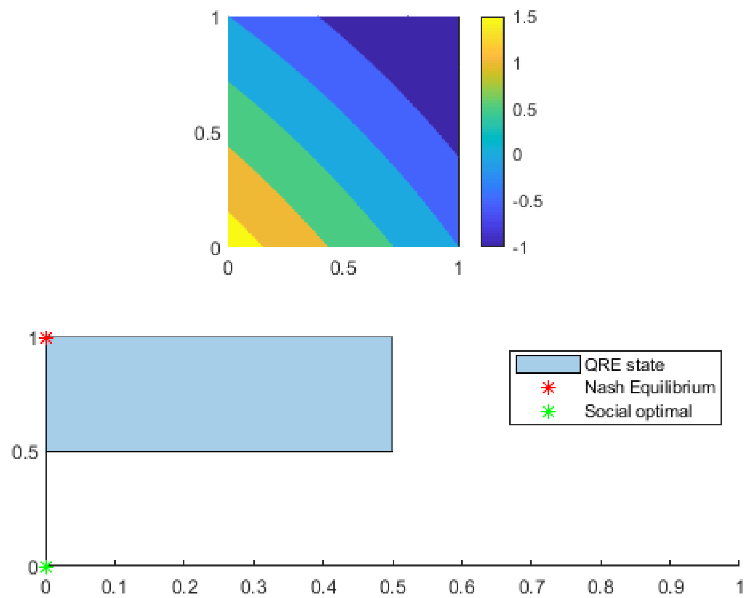
The situations in which SO and NE do not coincide are of greater interest, below we will analyze the different possible scenarios that will occur in auction problems.

5.2 Analysis

In the following analysis we will consider the scenario for which Nash Equilibrium does not coincide with Social Optimum. In all cases the starting equilibrium point is found for T_x and T_y equal to 0.

Case 1

$$\begin{bmatrix} -1.9358, 0.6507 & 0, 0 \\ 0, 0 & 2.4198, -0.6507 \end{bmatrix}$$



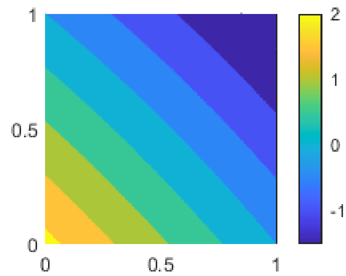
This game has one NE for $(0, 1)$ with a SW equal to 0. It does not coincide with the SO that is not QRE-Achievable. Increasing T_y it is possible to guide the system towards the point $(0, 0.5)$ which is QRE-Achievable with a SW equal to 0.8845 that is greater than the NE.

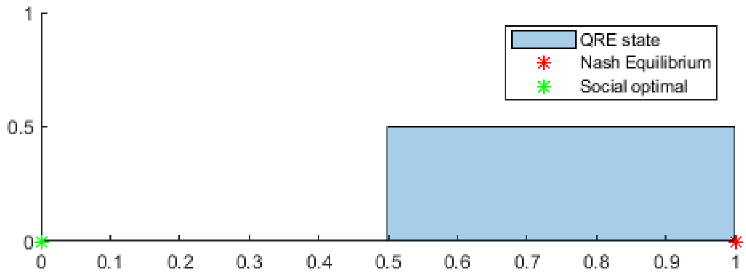
Theorem 7. *Given a 2×2 game if $a_x, b_y < 0$ and $a_y, b_x > 0$ the game has only one Nash equilibria and it is not the Social Optimal state than increasing*

T_y the mechanism allows to reach a state with an higher SW.

Case 2

$$\begin{bmatrix} 0.1021, -1.8664 & 0, 0 \\ 0, 0 & -0.1021, 2.2397 \end{bmatrix}$$



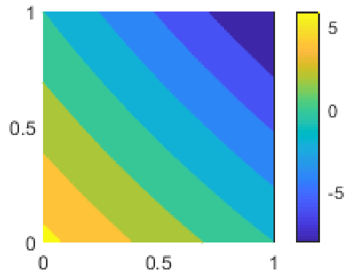


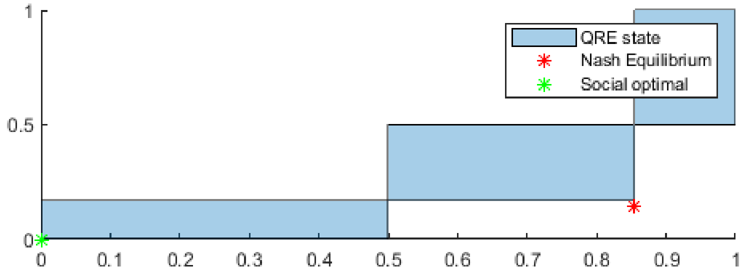
The game has only one NE for $(1, 0)$ with a SW equal to 0. The NE does not coincide with the SO which is not QRE-Achievable. Increasing T_x it is possible to guide the system towards the point $(0.5, 0)$ that is QRE-Achievable with a Social Welfare equal to 1.0688 greater than the NE.

Theorem 8. *Given a 2×2 game if $a_x, b_y > 0$ and $a_y, b_x < 0$ the game has only one Nash equilibria which is not the Social Optimal state. Increasing T_y the mechanism allows to reach a state with an higher SW.*

Case 3: no pure Nash Equilibria

$$\begin{bmatrix} -9.8559, 1.4428 & 0, 0 \\ 0, 0 & -1.9884, 8.4958 \end{bmatrix}$$





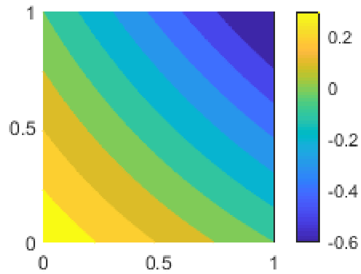
This game do not admits a pure Nash Equilibria but only a Mixed Nash Equilibria (MNE) for $(0.85, 0.15)$ as we can see form figure above the SO do not coincide with the MNE. Starting form any initial point and increasing the temperature of Player 2 the system reaches a state close to the SO.

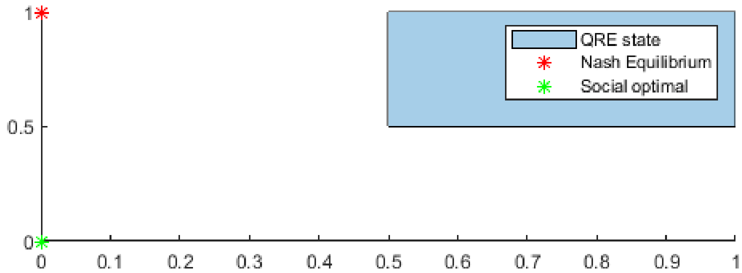
Theorem 9. *Given a* 2×2 *game with* $a_x, b_x < 0$ *and* $a_y, b_y > 0$*, if it has only a Mixed Nash Equilibria which is not the Social Optimal state, increasing* T_y *the mechanism allows to reach a state with an higher SW.*

If $a_x, b_x > 0$ and $a_y, b_y < 0$, increasing T_x the mechanism reaches a state with an higher SW with respect the NE.

Case 4: Nash Equilibrium does not belong to the set of QRE-Achievable state

$$\begin{bmatrix} -0.4819, -0.1853 & 0, 0 \\ 0, 0 & 0.4805, -0.0918 \end{bmatrix}$$





In this situation both Nash equilibrium and Social Optimal are not QRE-Achievable. Increasing the temperature parameters T_x and T_y there is no way to reach a state with an higher Social Welfare.

Theorem 10. *Given a* 2×2 *game if the Nash equilibrium do not coincide with the Social Optimal and both are not QRE-Achievable, increasing the temperature parameters there is no way to reach a state with an higher social welfare.*

Chapter 6

Conclusions

In this work we have presented a bifurcation studied applied to auction mechanism. First, we analyze the structure of the game obtained by simulation of the auction mechanism and we show that due to mechanism constraints we are not able to trace it back to a coordination game.

We conducted the bifurcation study by analyzing the dynamics of Q-Learning, FAQ and Gradient Ascent. The first two dynamics have a similar behaviour being the FAQ dynamics a variation of the Q-learning which uses a softmax activation function for policy-generation, and an update rule inversely proportional to x_i.

For non-coordination games, varying the temperature parameters the system remains stable and does not present any bifurcation point. An exception is when players have the same bid values and ties are broken randomly. In this situation, we can have a coordination game and the bifurcation study shows a bifurcation point where the system stability changes. For the GSP auction mechanism we also analyzed the behaviour of the system as it varied by q under the Q-learning dynamics. Analyzing the structure of the matrices we have noticed that they have similar structures between them and because q assumes a value between 0 and 0.1 the system turns out to be immune to the perturbations of this parameter. For the case in which the two players have the same bid values, a further analysis has been carried out through the linear combination of the matrices obtained by breaking the parity once in favour of Player 1 and once in favour of Player 2. In this case, we have analyzed the stability of the system to vary of ϵ. Since ϵ can assume a value between 0 and 1, this means that we're weighing one of the two matrices more heavily and for this reason the structure of the game do not change.

In the Gradient Ascent dynamics, the only parameter that can vary is the learning rate α, as we prove this parameter does not affect the stability of the system but only the speed at which it reaches equilibrium.

Finally, we designed an Optimal Control Mechanism to drive the system towards a state, which is not reachable by rational players, with Social Welfare higher than the Nash Equilibrium.

Bibliography

S. Abdallah and V. Lesser. A multiagent reinforcement learning algorithm with non-linear dynamics. *Journal of Artificial Intelligence Research*, 2008.

B. An, N. Gatti, and V. Lesser. Extending alternating-offers bargaining in one-to-many and many-to-many settings. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, pages 423–426. IEEE Computer Society, 2009.

B. An, N. Gatti, and V. Lesser. Bilateral bargaining with one-sided uncertain reserve prices. *Autonomous Agents and Multi-Agent Systems*, 26(3):420–455, 2013.

N. Basilico, G. De Nittis, and N. Gatti. A security game combining patrolling and alarm-triggered responses under spatial and detection uncertainties. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

N. Basilico, G. De Nittis, and N. Gatti. Adversarial patrolling with spatially uncertain alarm signals. *Artificial Intelligence*, 246:220–257, 2017.

M. O. Benjamin Edelman and M. Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review 97*, 2007.

M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 2002.

S. Ceppi, N. Gatti, G. Patrini, and M. Rocco. Local search techniques for computing equilibria in two-player general-sum strategic-form games. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 1469–1470. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

S. Ceppi, N. Gatti, and E. H. Gerding. Mechanism design for federated sponsored search auctions. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

S. Coniglio, N. Gatti, and A. Marchesi. Pessimistic leader-follower equilibria with multiple followers. 2017.

D. H. Daan Bloembergen, Karl Tuyls and M. Kaisers. Evolutionary dynamics of multi-agent learning: A survey. *Jurnal of Artificial Intelligence Research*, 2015.

M. K. Daan Bloembergen and K. Tuyls. Lenient frequency adjusted q-learning. *Artificial intelligence*, 2010.

F. Dercole and S. Rinaldi. *Analysis of Evolutionary Processes: The Adaptive Dynamics Approach and its Applications.* Princeton University Press, 2008.

F. Dercole and S. Rinaldi. Dynamical systems and their bifurcations. *Advanced Methods of Biomedical Signal Processing*, 2011.

A. Dhooge, W. Govaerts, Y. Kuznetsov, W. Mestrom, A. Riet, and B. Sautois. *MATCONT and CL_MATCONT: Continuation toolboxes in MATLAB.* 2010.

F. Di Giunta and N. Gatti. Bargaining over multiple issues in finite horizon alternating-offers protocol. *Annals of Mathematics and Artificial Intelligence*, 47(3-4):251–271, 2006.

D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World.* Cambridge University Press, 2010.

G. Faye. An introduction to bifurcation theory. 2011.

N. Gatti. Normal-form representation. 2018.

N. Gatti and M. Restelli. Sequence-form and evolutionary dynamics: realization equivalence to agent form and logit dynamics. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

N. Gatti, A. Lazaric, and F. Trovò. A truthful learning mechanism for contextual multi-slot sponsored search auctions with externalities. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 605–622. ACM, 2012.

N. Gatti, F. Panozzo, and M. Restelli. Efficient evolutionary dynamics with extensive-form games. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013a.

N. Gatti, M. Rocco, and T. Sandholm. On the verification and computation of strong nash equilibrium. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 723–730. International Foundation for Autonomous Agents and Multiagent Systems, 2013b.

N. Gatti, A. Lazaric, M. Rocco, and F. Trovò. Truthful learning mechanisms for multi-slot sponsored search auctions with externalities. *Artificial Intelligence*, 227:93–139, 2015.

D. B. Ger Yang and G. Piliouras. Bifurcation mechanism design—from optimal flat taxes to better cancer treatments. *Games*, 2018.

E. R. Gomes and R. Kowalczyk. Dynamic analysis of multiagent q-learning with ϵ-greedy exploration. *In Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

M. Kaisers. Replicator dynamics for multi-agent learning an orthogonal approach. *ALA*, 2009.

M. Kaisers. *Learning against Learning. Evolutionary dynamics of reinforcement learning algorithms in strategic interactions.* 2012.

M. Kaisers and K. Tuyls. Frequency adjusted multi-agent q-learning. *Autonomous Agents and Multiagent Systems*, 2010.

M. Kaisers and K. Tuyls. Faq-learning in matrix games: Demonstrating convergence near nash equilibria, and bifurcation of attractors in the battle of sexes. *AAAI*, 2011.

T. Klos, G. J. van Ahee, and K. Tuyls. Evolutionary dynamics of regret minimization. *In Machine Learning and Knowledge Discovery in Databases*, 2010.

Y. Kuznetsov. *Elements of Applied Bifurcation Theory.* Springer Nature, 2004.

Y. Kuznetsov. Computer session I: Using MATCONT for numerical integration of odes. 2009a.

Y. Kuznetsov. Computer session II: One-parameter bifurcation analysis of equilibria with MATCONT. 2009b.

Y. Kuznetsov. Computer session III: One-parameter bifurcation analysis of limit cycles with MATCONT. 2009c.

Y. Kuznetsov. Computer session IV: Two-parameter bifurcation analysis of equilibria and limit cycles with MATCONT. 2009d.

Y. Kuznetsov. Computer session V: Bifurcation analysis of maps with CL_MATCONTM. 2009e.

A. Lazaric, E. M. de Cote, F. Dercole, and m. Restelli. Bifurcation analysis of reinforcement learning agents in the selten's horse game. *Adaptive Agents and Multi-Agent Systems III*, 2008.

M. L. L. Leslie Pack Kaelbling and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 1996.

K. Leyton-brown, R. J. Brachman, and T. Dietterich. *Essentials of Game Theory*. Morgan & Claypool, 2008.

F. Lopes and H. Coelho. *Negotiation and argumentation in multi-agent systems: Fundamentals, theories, systems and applications*. Bentham Science Publishers, 2014.

R. Lucchetti. *A Primer in Game Theory*. Esculapio, 2011.

D. B. Michael Kaisers and K. Tuyls. A common gradient in multi-agent reinforcement learning. *Autonomous Agents and Multiagent Systems*, 2012.

E. S. Michael Maschler and T. Shmuel Zamir. *Game Theory*. Cambridge University Press, 2013.

S. S. Morris W. Hirsch and R. Devaney. *Differential equations, Dynamical systems, and an introduction to caos*. Academic Press, 2012.

E. Munoz de Cote, R. Stranders, N. Basilico, N. Gatti, and N. Jennings. Introducing alarms in adversarial patrolling games. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1275–1276. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Pres, 2007.

T. Börgers and R. Sarin. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 1997.

C. P. Panayotis Mertikopoulos and G. Piliouras. Cycles in adversarial regularized learning. *SODA*, 2018.

F. Panozzo, N. Gatti, and M. Restelli. Evolutionary dynamics of q-learning over the sequence form. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

L. Pavel. *Game Theory for Control of Optical Networks.* Birkhauser, 2012.

G. Piliouras, C. Nieto-Granda, H. I. Christensen, and J. S. Shamma. Persistent patterns: Multi-agent learning beyond equilibrium and utility. *Autonomous Agents and Multiagents Systems*, 2014.

S. M. Sai Ganesh Nagarajan and G. Piliouras. Three body problems in evolutionary game dynamics: Convergence,periodicity and limit cycles. *Autonomous Agents and Multiagents Systems*, 2018.

W. H. Sandholm. *Evolutionary Game Theory.* 2007.

W. H. Sandholm. *Population Games and Evolutionary Dynamics.* Mit Pr, 2010.

M. K. Satinder Singh and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. *UAI*, 2000.

D. C. P. Sébastien Lahaie and D. M. Pennock. An expressive auction design for online display advertising. *AAAI*, 2008.

Y. Shoham and K. Leyton-Brown. *Multiagent Systems. Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, 2010.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* Bradford Books, 1998.

S. Tadelis. *Game Theory: an introduction.* Princeton Univ Pr, 2013.

C. M. Tushar Vaidya and G. Piliouras. Learning agents in financial markets:consensus dynamics on volatility. *Autonomous Agents and Multiagents Systems*, 2018.

K. Tuyls and S. Parsons. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 2007.

K. Tuyls and G. Weiss. Multiagent learning: Basics, challenges, and prospects. *AI magazine*, 2012.

K. Tuyls, P. J. Hoen, and B. Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *Autonomous Agents and Multi-Agent Systems*, 2005.

V. C. Wassim M. Haddad. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach.* Princeton Univ Pr, 2008.

J. Weibull. *Evolutionary Game Theory.* Mit Pr, 1997.

M. Wooldridge. *An Introduction to MultiAgent Systems.* John Wiley & Sons Inc, 2009.

R. P. Yoav Shoham and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 2006.