POLITECNICO DI MILANO

School of industrial and information engineering

Master in automation and control engineering

RESOURCE AND LOAD ALLOCATION VIA LINEAR MULTIAGENT OPTIMIZATION: PROBABILISTIC CERTIFICATES OF SOLUTION STABILITY

Supervisor: Prof. Simone Garatti

Co-supervisors: Prof. Maria Prandini, Prof. Alessandro Falsone

M.Sc thesis by Jacopo Zizzo

Student ID 898344

Academic year 2018/2019

Contents

Abstract

Many engineering problems feature different units (machines, items or people) and a resource to be shared among them in such a way to maximize (or minimize) a certain profit or performance index. Examples of such problems can be a power company that has to fulfill a required load and has many plants available, or a warehouse with limited capacity that has to accommodate different goods in different quantities.

The previous work was focused on linear programming modeling of such problems, in which each agent is represented by a decision variable, contributes to a cost function with its own profit coefficient and is bound together with the other users by means of one or more coupling equality constraints.

Specifically the stability of the solution to such linear problems was assessed by using results coming from the scenario approach theory for stochastic optimization which provide a metric for the probability of the original solution changing if a new agent is added to the problem.

This research work aims to extend the achieved results to problems in which the amount of resource dedicated to each agent is subject to an upper bound and also to problems whose coupling constraints include also inequalities.

To better visualize the results, two realistic case studies are included along with their modeling and numerical simulation.

Sommario

In molti problemi ingegneristici viene affrontata la distribuzione ottimale di una risorsa tra diversi agenti (macchine o persone) in modo tale da massimizzare (o minimizzare) un certo indice di prestazioni. Esempi di suddetti problemi includono una società di distribuzione di energia elettrica che deve soddisfare una determinata domanda e possiede diversi impianti, oppure un magazzino con capacità limitata in cui devono essere conservati diversi beni in diverse quantità. Il lavoro precedente da cui parte questa tesi era incentrato su modelli di programmazione lineare di problemi di questo tipo, nei quali ciascun agente è rappresentato da una variabile decisionale , contribuisce a una funzione di costo con il suo coefficiente di profitto ed è legato agli altri agenti attraverso uno o più vincoli di accoppiamento. In particolare, la stabilità della soluzione di questi problemi lineari era valutata tramite risultati provenienti dalla teoria dell'approccio a scenario per l'ottimizzazione stocastica, la quale fornisce un metodo per misurare la probabilità di cambiamento della soluzione originale nel caso in cui un nuovo agente venga inserito nel problema. Lo scopo di questa tesi è di estendere i risultati ottenuti in precedenza a problemi in cui la quantità di risorsa dedicata a ciascun agente è soggetta a un limite superiore ed anche a problemi in cui i vincoli di accoppiamento includono disuguaglianze. Per visualizzare meglio i risultati, sono inclusi due casi di studio realistici.

1 Introduction

The topic of multi-agent games is well studied and documented in engineering and game theory. While the majority of research in this field is directed towards the computation of optimal strategies for each agent through iterative and distributed algorithms, this thesis will focus on linear programming models of problems in which a resource is to be shared among multiple users to the benefit of an external entity. In particular, rather than maximizing the utility functions of each (or some) agent, the global utility score, represented by the cost function of the problem, is maximized (or minimized) considering an initial pool of agents, then the stability of the retrieved solution is tested against new agent arrivals that may lead to an improvement in the cost function value. Examples of problems of this kind are found in every situation in which there is a planning phase to allocate space or distribute a task. For example a first case study which is exactly in this framework and will be developed in the thesis is about an energy producer that has to split a required power load among many power plants with different characteristics aiming to reduce fuel costs. A second example, also developed in the thesis regards an export company which has to make the best use possible of the limited space in one of its aircrafts to fit the best mix of products in order to maximize profit.

Once the optimal resource allocation problem is solved with a given pool of agents, it may be that just a part of the agents concur in the determination of the solution, while the others are not employed because it would disadvantageous to. When a new agent is added to the original pool and the solution is recomputed, it may either happen that the solution changes, in which case, the newly arrived agent must take part in it, or that the original solution remains unchanged, which means that the newly arrived agent becomes part of the unemployed ones. Stability of the solution with respect to the original pool of agents refers to the tendency of the solution to not change in presence of a new agent and the main point of the thesis is to develop results in the vein of the scenario approach theory [1] [2] [3] [4] to estimate the probability of the initial solution

changing if a new and unseen agent is included in the original optimization problem. This probability of improving the previous solution, also called *stability index*, has an important meaning and can be an useful tool for the planner. In the first of the two examples, an operator working for the energy producer has access to data only about a portion of all the power plants that can be employed. He/She can solve the allocation problem using only the available data or can request a new poll to get information about other plants the company owns. Since this polling operation takes time and money, it may not be the best choice to request it, even more if the newly acquired data do not bring an improvement to the solution obtained with only the previously available information. The aforementioned scenario approach theory will be exploited in order to obtain a metric to decide if the solution found is good enough or if it can still be improved with reasonable effort in searching for a new agent.
The exact same reasoning can be applied to the second example in which an air cargo company charges clients to carry their merch. Waiting more time and accepting more requests can be advantageous if clients willing to pay more to have their goods delivered in time show up but if they do not the air cargo company loses time and possibly money if the delivery service is delayed because of this. Again estimating the probability of change delivers precious information to decide whether it is better to search for new clients or go on with the present solution as it is unlikely to improve.

1.1 Mathematical formulation

From a mathematical standpoint, it is assumed that the generic multi-agent resource sharing problem is described by a linear program in one of the following three forms that model slightly different situations:

$$\begin{aligned} \mathcal{P}_0 : \min_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i = b \\ & x^i \geq 0 \qquad i = 1, .., m, \end{aligned} \tag{1}$$

$$\begin{array}{llll} \mathcal{P}_1: & \min\limits_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m} & \sum_{i=1}^m (c^i)^T x^i & \\ & \text{s.t.} & \sum_{i=1}^m A^i x^i = b & \\ & & 0 \leq x^i \leq d^i \quad i = 1,..,m, & \end{array} \tag{2}$$

$$\begin{array}{llll} \mathcal{P}_2: & \min\limits_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m} & \sum_{i=1}^m (c^i)^T x^i & \\ & \text{s.t.} & \sum_{i=1}^m A^i x^i \leq b & \\ & & 0 \leq x^i \leq d^i \quad i = 1,..,m, & \end{array} \tag{3}$$

In all of these problems there are m agents each one associated to a decision vector $x^i \in \mathbb{R}^{n_i}$ which is positive (≥ 0) component-wise, and a corresponding cost coefficient c^i. b $\in \mathbb{R}^p$ is a vector representing the shared resource and in all problems the agents compete for it, while $A^i \in \mathbb{R}^{p \times n_i}$ represents the utilization of the resource b by agent i that is required to implement an unitary decision vector ($A^i x^i$ is the utilization of b to implement x^i). The amount of b that is allocated to the single agent x_i can be unlimited (problem $\mathcal{P}_0$) or limited (problems $\mathcal{P}_1$ and $\mathcal{P}_2$) by means of the additional constraint on the maximal value of x^i given by $d^i \in \mathbb{R}^{n_i}$. Moreover, the resource b may be left partially unallocated as in $\mathcal{P}_2$, where the resource allocation constraint is posed as an inequality. Reasons for introducing this latter possibility are either because the agents are unable to fully utilize the resource due to their limits d^i or, in case b is a vector, it may be difficult or impossible to find a situation where all components of b are fully used. The example of Chapter 6 will provide a situation of this type.

Each one of the m agents is associated with a tuple $\delta^i = (n_i, c^i, A^i, d^i)$, and the mechanism through which the agents become available and take part in the decision problem is assumed to be stochastic. Specifically the $\{\delta^i\}_{i=1}^m$ is assumed to be a sample of a random variable $\delta = (n, c, A, d)$ defined over a generic probability space $(\Delta, \mathcal{D}, \mathbb{P})$. The instances in the sample are assumed to be independent and identically distributed (i.i.d.), so the collection $\{\delta^i\}_{i=1}^m$ is distributed according to the product probability measure $\mathbb{P}^m$. In other words, the problems $\mathcal{P}_0$, $\mathcal{P}_1$ and $\mathcal{P}_2$ are random linear problems over m realizations of δ and each one of them is furthermore assumed to have a unique minimizer with probability 1.

To formalize the concept of stability, consider now a situation where one may want to introduce new agents in order to see if they can improve the resource exploitation. The agents in Δ may be infinite so exploring or waiting for them all to show up is not an option. Even the exploration can be costly and time consuming and for these reasons one would like to draw some conclusions on the possibility of improving the initial solution based on the first batch of m observed agents $\{\delta^i\}_{i=1}^m$. To be precise, suppose a new agent $y \in \mathbb{R}^{n_y}$ is observed alongside its tuple $\delta^y = (n_y, c^y, A^y, d^y)$. The starting resource allocation problem is now modified and defined over $m+1$ agents instead of m, and, depending on whether the initial problem were $\mathcal{P}_0$, $\mathcal{P}_1$ or $\mathcal{P}_2$, it becomes either:

$$\begin{aligned} \mathcal{P}_{0_+}: \quad \min_{\{x^i\in\mathbb{R}^{n_i}\}_{i=1}^m, y\in\mathbb{R}^{n_y}} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i + (c^y)^T y \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i + A^y y = b \\ & x^i \geq 0 \qquad i = 1, ..., m \\ & y \geq 0 \end{aligned} \tag{4}$$

or

$$\begin{aligned} \mathcal{P}_{1_+}: \quad \min_{\{x^i\in\mathbb{R}^{n_i}\}_{i=1}^m, y\in\mathbb{R}^{n_y}} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i + (c^y)^T y \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i + A^y y = b \\ & 0 \leq x^i \leq d^i \qquad i = 1, ..., m \\ & 0 \leq y \leq d^y \end{aligned} \tag{5}$$

or

$$\begin{aligned} \mathcal{P}_{2_+}: \quad \max_{\{x^i\in\mathbb{R}^{n_i}\}_{i=1}^m, y\in\mathbb{R}^{n_y}} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i + (c^y)^T y \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i + A^y y \leq b \\ & 0 \leq x^i \leq d^i \qquad i = 1, ..., m \\ & 0 \leq y \leq d^y \end{aligned} \tag{6}$$

Before continuing, a a remark on the notation used has to be made: a superscript as in x^i denotes that a vector is associate with agent i, while a subscript as in x_i generally denotes the $i-th$ element of a vector.

The decision vectors for $\mathcal{P}_{0,1,2}$ and $\mathcal{P}_{0_+,1_+,2_+}$ are $x = [x_1, \ldots, x_m]$ and $x_+ = [x_1, \ldots, x_m, y]$, respectively, and let us call the optimal solutions for these problems x^* and $x_+^\star = (x^\star, y^\star)$.

The newly added agent y can only improve the original solutions, since the costs of the solutions to $\mathcal{P}_{0_+,1_+}$ will be lesser than or equal than the ones of $\mathcal{P}_{0,1}$ or in the case of $\mathcal{P}_{2_+}$ and $\mathcal{P}_{2_+}$, greater than or equal. If the costs of x^* and $x_+^\star$ are equal, then certainly $y^\star = 0$ and $x_+^\star = (x^*, 0)$. If instead y brings improvement over the original solutions x^*, it is certainly assigned a share of b, effectively rendering $x_+^\star \neq (x^*, 0)$. The probability of this event happening is called *stability index* and is formally defined below:

Definition 1.1. - Stability index Consider the solution to $\mathcal{P}_0, 1, 2$ x^* and the new agent y with its tuple $\delta(c_i, A_i, d_i)$ giving rise to $\mathcal{P}_{0_+,1_+,2_+}$. The stability index $V(x^*)$ is the probability of x^* changing because of the addition of δ_y:

$$V(x^*) = \mathbb{P}\big\{\delta^y = (c^y, A^y, d^y) \in \Delta : \; x_+^\star \neq (x^*, 0)\big\}, \tag{7}$$

Since x^* depends on $\{\delta^i\}_{i=1}^m$, it is a random vector defined over $(\Delta^m, \mathbb{P}^m)$, where $\mathbb{P}^m$ represents a product probability since all δ_i are i.i.d and so $V(x^*)$, being a function of a random vector is itself a random variable over $(\Delta^m, \mathbb{P}^m)$, and as such any statement concerning $V(x^*)$ will be true with a certain confidence $\beta \in (0, 1)$ with respect to $\mathbb{P}^m$. The problem is that both the distribution and the mechanism through which the agents are observed are usually not known giving rise to a particularly complex problem that will be confronted throughout this research.

1.2 Contributions and thesis organization

The goal of this thesis is to characterize $V(x^*)$ as a tool useful to take decisions in situations where waiting for new agents or polling them may result or not in improvements over a previously computed solution. In particular:

- In Chapter 2 a recap of the main results achieved previously [5] for problems like $\mathcal{P}_0$ through tools from the classic *scenario approach* [1] [2] [3] theory is presented.

- In Chapters 3 and 4 the main theoretical contribution of this thesis is presented. A more recent development of the *scenario approach* theory [4] is employed in order give a characterization of the *stability index* for problems in non-standard form since what is achieved in Chapter 2 does not stand valid for problems of the form of $\mathcal{P}_1$ and $\mathcal{P}_2$.

- In Chapter 5 and 6 the two case studies on economic dispatch and air cargo loading problems, introduced in the previous paragraph, are more detailedly described and formulated in an appropriate way for numerical simulations. The two examples will serve to validate the results of Chapter 3 and 4.

- In Chapter 7 general conclusions on both the theoretical and the numerical results achieved in this research work are drawn, while in the Appendix more insights on the *scenario approach* theory and also the MATLAB code used for the numerical simulations can be found.

2 Previous results, problems in standard form

Let us recall problem $\mathcal{P}_0$:

$$\begin{aligned} \mathcal{P}_0 : \min_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i = b \\ & x^i \geq 0 \qquad i = 1, .., m, \end{aligned} \tag{8}$$

with $n = \sum_{i=1}^m n_i$ the total number of decision variables in $\mathcal{P}_0$, $x = [(x^1)^T, \dots, (x^m)^T]^T \in \mathbb{R}^n$, $A = [A^1 ... A^m] \in \mathbb{R}^{p \times n}$, $c = [(c^1)^T ... (c^m)^T]^T \in \mathbb{R}^n$,where only m agents drawn independently from a generic probability distribution are considered and denote as x^* its optimal solution. This chapter will be a recap of the main results of [5], providing means to characterize the random variable $V(x^*)$ as defined in Chapter 1, the probability that the optimal solution of $\mathcal{P}_0$ will change upon the arrival of a new agent y, associated with the tuple $\delta^y = (n_y, c^y, A^y)$, as in $\mathcal{P}_{0_+}$:

$$\begin{aligned} \mathcal{P}_{0_+} : \min_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m, y \in \mathbb{R}^{n_y}} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i + (c^y)^T y \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i + A^y y = b \\ & x^i \geq 0 \qquad i = 1, ..., m \\ & y \geq 0, \end{aligned} \tag{9}$$

whose solution is $x_+^\star = (x^\star, y^\star)$.

The main result is provided below, along with some comments on the notation and on its meaning:

Theorem 2.1. Let $\beta \in (0, 1 =)$ and $\varepsilon \in (0, 1)$ such that the following relation holds:

$$\sum_{k=0}^{p-1} \binom{m}{k} \varepsilon^k (1-\varepsilon)^{m-k} \leq \beta, \tag{10}$$

then it holds true that:

$$\mathbb{P}^m\big\{(\delta^1, \dots, \delta^m) \in \Delta^m : V(x^*) < \varepsilon\big\} \geq 1 - \beta, \tag{11}$$

where p is the row-rank of A in $\mathcal{P}_0$, $\varepsilon \in (0,1)$ represents an upper bound for the stability index and $\beta \in (0,1)$ is the confidence with which the above statement is true with respect to $\mathbb{P}^m$.

Usually β is fixed by the user and the bound ε for the satbility index is retrieved from the equation.
Setting a very small β, for example 10^{-7} ensures that $V(x^*) < \varepsilon$ is verified almost surely (apart from a case in 10 million) regardless of the particular $(\delta^1, \dots, \delta^m)$ extracted.
Concerning ε, having $V(x^*) < 0.1$ means that the probability that a new agent could bring improvements over x^* cannot be very high. It may take much effort, time or simply sampling many agents before finding it, while if $V(x^*) < 0.9$ the probability of improvement may be much larger than the previous case, requiring less time or much less agents to examine before finding the special one.
The rest of this chapter is dedicated to the proof of Theorem 2.1 and is organized as follows:

- In Section 2.1 a characterization of the optimal solution for problems of the type of $\mathcal{P}_0$ is provided.

- In Section 2.2 the dual of $\mathcal{P}_0$, $\mathcal{D}_0$ is introduced and a connection between their optimal solutions is established.

- In Section 2.3 the scenario optimization theory is applied on $\mathcal{D}_0$ and then mapped back to $\mathcal{P}_0$ to prove the main result.

2.1 Optimality conditions for a standard form problem

This section will deal with the characterization of the optimal solution x^* of $\mathcal{P}_0$:

$$\begin{aligned} \mathcal{P}_0 : \min_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i = b \\ & x^i \geq 0 \qquad i = 1, .., m, \end{aligned} \tag{12}$$

where $n = \sum_{i=1}^m n_i$ the total number of decision variables in (12), $x = [(x^1)^T, \ldots, (x^m)^T]^T \in \mathbb{R}^n$, $A = [A^1 ... A^m] \in \mathbb{R}^{p \times n}$, $c = [(c^1)^T ... (c^m)^T]^T \in \mathbb{R}^n$ and assume that A is full row rank. A superscript as in x^i indicates an association to agent i, while a subscript as in x_i denotes the $i - th$ component of such vector.
In order to characterize x^*, the concept of basic solution has to be introduced:

Definition 2.1. **-Basic solution** Consider a polyhedron P and a vector $x^v \in \mathbb{R}^n$. x^v is a **basic solution** if at x^v all equality constraints are active and n linearly independent constraints are active at x^v in total.
If x^v satisfies all the constraints., then it is also a **basic feasible solution**. According to theorem 2.3 of [6], basic solution $\iff$ vertex of a polyhedron.

Moreover, another definition is introduced that will help with defining the optimality conditions for x^*:

Definition 2.2. **-Degenerate basic solution** If more than n constraints are active at a basic solution $x^* \in \mathbb{R}^n$, it is said to be **degenerate**.

Since A is full row rank, theorems 2.3-2.4 of [6] can be applied and thus any basic feasible solution x^v of (12) is partitioned in two sets of variables: basic (x^v_B) and non-basic (x^v_N).
Denote with A_B the partition of A associated with the same indexes of the variables in x^v_B and with A_N the one corresponding with x^v_N. Since x^v is feasible for (12), it is true that:

$$Ax^v = A_B x^v_B + A_N x^v_N = b. \tag{13}$$

but remembering that $x_N^v = 0$ (theorem 2.4 of [6]) and since A_B is a full rank square matrix, this results in $x_B^v = A_B^{-1} b$.
By calling c_B and c_N the partitions of vector c corresponding respectively to basic and non-basic variables and also assuming that (12) admits a **non-degenerate** minimizer, theorem 3.1 of [6] guarantees that x^v is the optimal solution x^* for (12) *if and only if*:

$$c_N^T - c_B^T A_B^{-1} A_N \geq 0, \tag{14}$$

The quantity on the left hand sign of (14) is called reduced cost, $\bar{c}$.

2.2 Dual problem

The proof of Theorem 2.1 requires information derived from the optimal solution of the dual (see Chapter 4 of [7] or [6]) of (12), $\mathcal{D}_0$. In this subsection connections are drawn between the primal and the dual problem, here presented:

$$\begin{aligned} \mathcal{D}_0 : \max_{\lambda} \qquad & -\lambda^T b \\ \text{s.t.} \qquad & \lambda^T A^i + c^i \geq 0 \end{aligned} \tag{15}$$

where $\lambda \in \mathbb{R}^p$ is the vector of dual variables, associated with the equality constraints of $\mathcal{P}_0$. Let us denote λ^* as the optimal solution for (15).
The following paragraph will go on with the proof of the proposition:

Proposition 2.1. - Characterization of the dual solution The optimal solution to (15) is $\lambda^* = -(c_B^T A_B^{-1})^T$.

Proof: The feasibility of λ^* will be checked first, then the optimality. By evaluating the constraints of $\mathcal{D}_0$ at λ^* one obtains:

$$\begin{aligned} & c_i + (\lambda^*)^T A_i \geq 0 \\ & c_i - c_B A_B^{-1} A_i \geq 0 \end{aligned} \tag{16}$$

By considering the basic and non-basic partitions:

$$\begin{aligned} & c_N - c_B A_B^{-1} A_N \geq 0 \qquad \text{because of (14)} \\ & c_B - c_B A_B^{-1} A_B = 0 \end{aligned} \tag{17}$$

So λ^* is feasible. Concerning its optimality, consider the dual cost function evaluated at λ^*:

$$\begin{aligned} -(\lambda^*)^T b &= c_B^\top A_B^{-1} b \\ &= c_B^\top x_B^\star \qquad \text{(remember that } x_B^* = A_B^{-1} b) \\ &= c_B^\top x_B^\star + c_N^\top x_N^\star \\ &= c^\top x^*. \qquad \text{(and } x_N^* = 0) \end{aligned}$$

Since the values of the primal and dual cost functions are equal, the optimality of λ^* is proven by strong duality (theorem 4.4 of [6]).

2.3 New agent arrival: scenario approach

In order to obtain a practical measure of the probability of the original solution of $\mathcal{P}_0$ changing upon the arrival of a new agent, the scenario approach theory for random convex problems [1], [2], [3], is hereby introduced. The theory will be applied to the dual of $\mathcal{P}_0$ since it has the required structure and the results will then be mapped back to $\mathcal{P}_0$.

Consider $\mathcal{D}_0$ and $\lambda^* = -(c_B^T A_B^{-1})^T$. By taking $\varepsilon, \beta \in (0,1)$ such that:

$$\sum_{k=0}^{p-1} \binom{m}{k} \varepsilon^k (1-\varepsilon)^{m-k} \leq \beta. \tag{18}$$

and assuming P_0 is feasible and admits a unique minimizer, Theorem 1 of [2] guarantees that:

$$\mathbb{P}^m \big\{ (\delta^1, \dots, \delta^m) \in \Delta^m : \mathbb{P}\big\{ \delta^y = (n_y, c^y, A^y) \in \Delta : \; c_y^\top + (\lambda^*)^\top A^y \geq 0 \big\} \geq 1 - \varepsilon \big\} \geq 1 - \beta, \tag{19}$$

i.e., with confidence at least $1-\beta$ (measured with respect to $\mathbb{P}^m$), the optimal solution $\lambda^\star$ of $\mathcal{D}_0$ remains feasible for a constraint generated by a new extraction $\delta^y = (n_y, c^y, A^y)$ with probability at least $1-\varepsilon$.

Now consider $\mathcal{P}_0$ and $\mathcal{P}_{0_+}$, namely the problem that arises upon the arrival of a new agent y characterized by $\delta_y = (n_y, c^y, A^y)$:

$$\begin{aligned} \mathcal{P}_{0_+}: \quad \min_{\{x^i\in\mathbb{R}^{n_i}\}_{i=1}^m, y\in\mathbb{R}^{n_y}} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i + (c^y)^T y \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i + A^y y = b \\ & x^i \geq 0 \qquad i = 1, ..., m \\ & y \geq 0 \end{aligned} \tag{20}$$

Recalling that x^* and $x_+^\star = (x^\star, y^\star)$ denote their respective solutions, let us restate Theorem 2.1:

Theorem 2.1 Fix any $m \in \mathbb{N}$. Fix $\varepsilon, \beta \in (0,1)$ such that (18) holds.

Assuming $\mathcal{P}_0$ is feasible and admits a unique minimizer,

$$\mathbb{P}^m\big\{(\delta^1,\ldots,\delta^m) \in \Delta^m: \ \mathbb{P}\big\{\delta^y = (n_y, c^y, A^y) \in \Delta: \ x_+^\star = (x^*, 0)\big\} \geq 1-\varepsilon\big\} \geq 1-\beta, \tag{21}$$

i.e., with confidence at least $1-\beta$, $x_+^\star = (x^*, 0)$ with probability at least $1-\varepsilon$.

Proof: Fix $\{\delta^i\}_{i=1}^m$ and consider $\mathcal{P}_{0_+}$. Take $x_+^\star = (x^*, 0)$, which is certainly a basic feasible solution for $\mathcal{P}_{0_+}$. Since $y^\star = 0$, the new agent y is a nonbasic component of $x_+^\star$. Recalling the definition of *reduced cost* in (14) for $\mathcal{P}_0$, it is true that $x_+^\star = (x^*, 0)$ is optimal for $\mathcal{P}_{0_+}$ if and only if:

$$(c^y)^T - c_B A_B^{-1} A^y \geq 0 \tag{22}$$

But since the optimal solution for $\mathcal{D}_0$ $\lambda^* = -(c_B^T A_B^{-1})^T$, (22) is equal to:

$$(c^y)^T - (\lambda^*)^T A^y \geq 0 \tag{23}$$

So in conclusion $x_+^\star = (x^*, 0)$ is optimal for $\mathcal{P}_{0_+}$ if and only if (23) holds.
Therefore,

$$\begin{aligned}
&\mathbb{P}\big\{\delta^y = (n_y, c^y, A^y) \in \Delta :\ x_+^\star = (x^*, 0)\big\} = \qquad (24)\\
&\mathbb{P}\big\{\delta^y = (n_y, c^y, A^y) \in \Delta :\ (c^y)^\top + (\lambda^*)^\top A^y \geq 0\big\}.
\end{aligned}$$

For any $\varepsilon, \beta \in (0, 1)$ such that (18) holds.
The right hand side of (24) is actually equal to 1 minus the stability index of x^* and as such has to be greater than $1 - \varepsilon$, keeping the same meaning of $V(x^*) < \varepsilon$. Since the scenario approach is applied to the dual of $\mathcal{P}_0$, $\mathcal{D}_0$, the number of scenarios N (here equal to m), corresponding to N extraction of constraints of $\mathcal{D}_0$, is equivalent to the number of agents m that are being extracted (also remembering that each decision variable in the primal problem is mapped to a constraint in the dual problem), while p, the number of coupling constraints and equal to the number of decision variables in $\mathcal{D}_0$, corresponds to the number of *support constraints* [see chapter 3, definition 3.1] for the dual optimal solution.

3 Problems with upper bounds

This chapter will deal with problem $\mathcal{P}_1$:

$$\begin{aligned} \mathcal{P}_1 : \min_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i = b \\ & 0 \leq x^i \leq d^i \quad i = 1,..,m, \end{aligned} \tag{25}$$

which, differently from $\mathcal{P}_0$, accounts for upper bounds d^i to the agents x^i, resulting in the fact that the m agents are subject to a limit on the quantity of the shared resource b that can be assigned to them.
Recall that a superscript as in x^i indicates an association to agent i, while a subscript as in x_i denotes the $i - th$ component of the total decision vector $x = [x^1, x^2, \dots x^m]^T$
Similarly to the previous chapter, the objective is to provide a probabilistic characterization of the stability index $V(x^*)$ in the context of $\mathcal{P}_1$ (where x^* denotes the optimal solution to (25)) and, as in the previous chapter, it is worth to point out that $V(x^*)$ is a random variable since x^* is random. To this purpose, it is interesting to note that the previous results of Chapter 2 cannot be used. This means that

$$\mathbb{P}^m\big\{(\delta^1, \dots, \delta^m) \in \Delta^m : V(x^*) < \varepsilon\big\} \geq 1 - \beta, \tag{26}$$

where β and ε satisfy (18), is not a valid statement. More in detail, $\mathbb{P}^m\big\{(\delta^1, \dots, \delta^m) \in \Delta^m : V(x^*) < \varepsilon\big\}$, seen as a function of ε, represents the cumulative distribution function (CDF) of $V(x^*)$ and according to (26) should always be above $1 - \beta$ where β is given by (18). This yields the following stochastic dominance relation:

$$\mathbb{P}^m\big\{(\delta^1, \dots, \delta^m) \in \Delta^m : V(x^*) < \varepsilon\big\} \geq 1 - \sum_{k=0}^{p-1} \binom{m}{k} \varepsilon^k (1-\varepsilon)^{m-k}, \tag{27}$$

but it will be shown by means of a practical counterexample that it is false in the context of $\mathcal{P}_1$.

Example 3.1 - Consider the auxiliary problem $\mathcal{P}_a$, which is of the same type as $\mathcal{P}_1$:

$$\begin{aligned}\mathcal{P}_a: \quad \min_{x^i} \quad & \textstyle\sum_{i=1}^{110} c^i x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{m} x^i = 100 \\ & 0 \leq x^i \leq d^i \qquad i = 1, ..., 110,\end{aligned} \tag{28}$$

where $x^i \in \mathbb{R}$, $c^i \in \mathbb{R}$, $d^i \in \mathbb{R}$ and A^i is equal to 1 for all agents, implying that $p = 1$. The mechanism through which the agents are generated is as follows: the d^i are extracted from a uniform distribution between 0 and 30 and the c^i are extracted from a uniform distribution too but between 0 and 1. A numerical simulation was run in order to obtain an empirical estimate of the CDF of $V_a(x^*)$, the stability index of the optimal solution of $\mathcal{P}_a$: at first $m = 110$ agents with their tuples $\delta^i = (c^i, d^i)$ were extracted and the solution x^* retrieved, then another 5500 agents were extracted from the same distributions and one by one they were added to the initial pool of m agents, each time turning $\mathcal{P}_a$, into:

$$\begin{aligned}\mathcal{P}_{a_+}: \quad \min_{x^i, y} \quad & \textstyle\sum_{i=1}^{110} c^i x^i + c^y y \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{m} x^i + y = 100 \\ & 0 \leq x^i \leq d^i \qquad i = 1, ..., 110 \\ & 0 \leq y \leq d^y,\end{aligned} \tag{29}$$

where y represents the newly extracted agent that is being added to $\mathcal{P}_a$ alongside its tuple $\delta^y = (c^y, d^y)$. The solution to $\mathcal{P}_{a_+}$, $x^\star_+ = (x^\star, y^\star)$ was then obtained and compared to x^*: in particular it was checked if the value of $y^\star$ in $x^\star_+$ was different from zero, i.e. the solution had changed.

To obtain an empirical estimate of $V_a(x^*)$ the number of times, out of 5500, $x^\star_+$ was different from x^* was counted and divided by the total number of new agents 5500. This procedure was repeated 100 times, obtaining 100 different values for $V_a(x^*)$ corresponding to different realizations of the agents. The empirical CDF was then retrieved from this values.

Considering now $1-\sum_{k=0}^{p-1}\binom{m}{k}\varepsilon^k(1-\varepsilon)^{m-k}$, where $p=1$, $m=110$ and $\varepsilon\in(0,1)$, the empirical CDF of $V_a(x^*)$ should always be above it (recall (27)). A plot of the empirical CDF of $V_a(x^*)$ and of $1-\sum_{k=0}^{p-1}\binom{m}{k}\varepsilon^k(1-\varepsilon)^{m-k}$ is given in Figure 1 and, as it can be seen the CDF of $V_a(x^*)$ is entirely below $1-\sum_{k=0}^{p-1}\binom{m}{k}\varepsilon^k(1-\varepsilon)^{m-k}$, thus showing that the previous result of Chapter 2 is unusable for problems like $\mathcal{P}_1$.

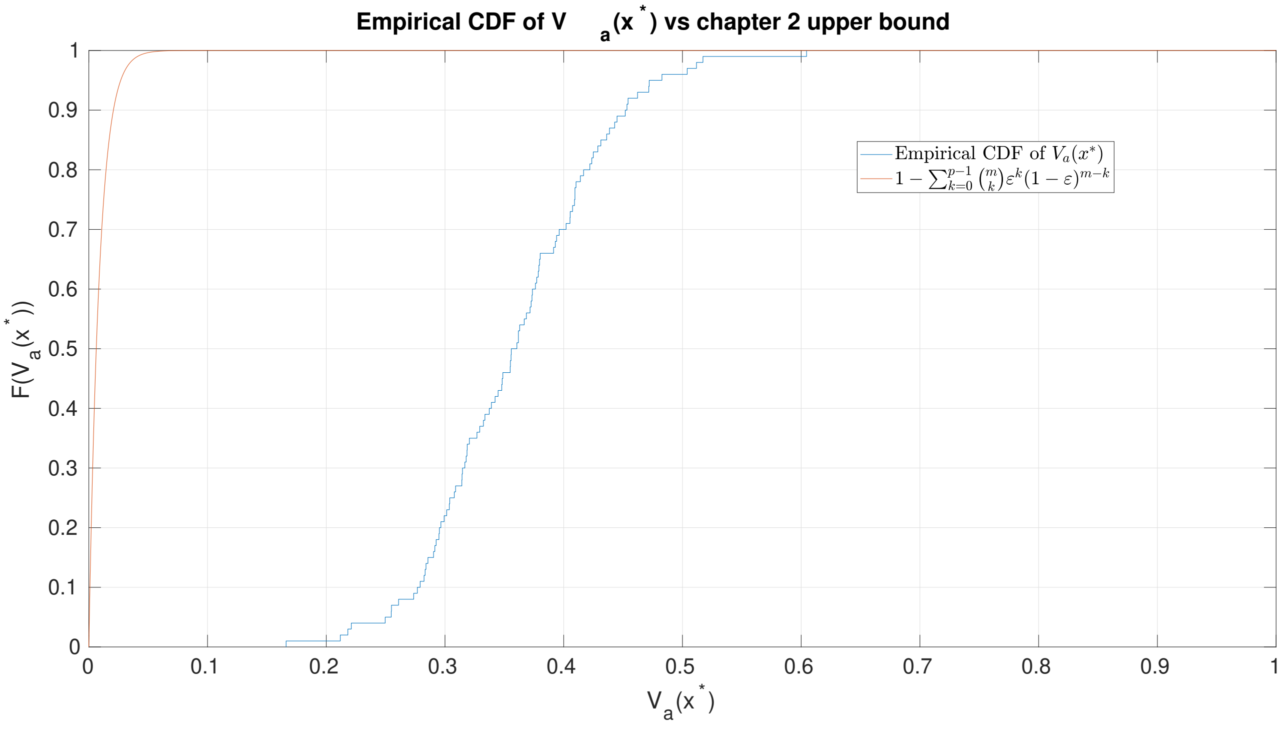

Figure 1: The CDF of $V_a(x^*)$ is always under the bound introduced in chapter 2.

Example 3.1 shows that it is not possible to provide a characterization of $V(x^*)$ in the same vein as in Theorem X in Chapter 2. The following perspective is inspired by some recent result in the scenario optimization theory, called *wait-and-judge* approach [4]. Instead of looking for $a-priori$ bounds to the stability index $V(x^*)$ that hold with high confidence, $a-posteriori$ bounds to $V(x^*)$ are searched, meaning that the bound depends on some quantity that becomes observable after $\mathcal{P}_a$ is solved

The previous example suggests what can be observed in order to tightly bound $V(x^*)$. If the d_i are very small with respect to 100, many agents have to be used in order to satisfy the constraint $\sum x^i = 100$, even the ones with high cost coefficients c^i. Therefore, the probability of observing a new agent with a better cost coefficient than one of the already used ones is higher.

If instead the d^i are quite big with respect to 100, less agents have to be used in order to satisfy the constraint $\sum x^i = 100$, obviously the few with the lowest cost coefficients out of the 110 original ones. In this situation it would be harder to observe a better agent.

This reasoning suggests that a tight evaluation of $V(x^*)$ must be based on the number of agents effectively employed in the determination of the solution x^* and leads to the following key definition.

Definition 3.1. - Number of support agents Consider problem $\mathcal{P}_1$ and its solution $x^* = [x_1^*, x_2^*, \dots, x_m^*]$. An agent is said to be a support agent if its optimal decision vector x_i^* is different from zero (in case x_i is a vector $\in \mathbb{R}^{n_i}$, it will be considered different from zero if at least one of its subcomponents is different from zero). The number of $x_i^* \neq 0$ (i.e. the number of support agents) will be denoted by s^*.

As it will be shown later on, the concept of support agents is related to that of support constraints, which is the observable quantity used in the scenario optimization theory [4], [8]. Precisely it will be shown that the number of support agents is equal to the number of support constraints of the dual of problem $\mathcal{P}_1$. The main theoretical contribution of this thesis is given by the following theorem.

Theorem 3.1. Let $\beta \in (0,1)$ and let $\epsilon(k) \in (0,1)$ for $k = 0, 1, \dots, m$ be equal to $1 - t(k)$ where $t(k)$ for $k = 0, 1, \dots, m$ is the unique solution of the polynomial equation

$$\frac{\beta}{m+1} \sum_{i=k}^{m} \binom{i}{k} t^{i-k} - \binom{m}{k} t^{m-k} = 0, \tag{30}$$

in the interval (0,1) and $t(m) = 0$. Then it holds that:

$$\mathbb{P}^m\big\{(\delta^1, \dots, \delta^m) \in \Delta^m : V(x^*) < \epsilon(s^*)\big\} \geq 1 - \beta, \tag{31}$$

where x^* is the optimal solution to $\mathcal{P}_1$ and s^* is the number of support agents of x^*.

$\epsilon(k)$, as a function of k, is retrieved from (30) for fixed values of β prior to obtaining s^*. Then, after evaluating s^*, the corresponding value of $\epsilon(s^*)$ can be calculated.
Theorem 3.1 says that the statement $V(x^*) \leq \epsilon(s^*)$ is true with confidence $1-\beta$. If beta is chosen to be, for example, 10^{-7} it can be reasonably assumed as true always (in only 1 case out of 10 million it would result false), making $\epsilon(s^*)$ an accurate upper bound to $V(x^*)$.
$\epsilon(s^*)$ is not known beforehand because it depends on s^* which is however observable. The upper bound to $V(x^*)$, $\epsilon(s^*)$ depends on what is observed, providing estimates tuned on the obtained solution that can be considered tight.
In Figures 2 and 3 $\epsilon(k)$ is plotted as a function of m and β. As it can be seen, decreasing β below a certain threshold does not yield significant changes, so setting for example $\beta = 10^{-7}$ is enough to consider the bound $\epsilon(k)$ tight. Moreover, by increasing m, the difference between different values of β becomes less evident.

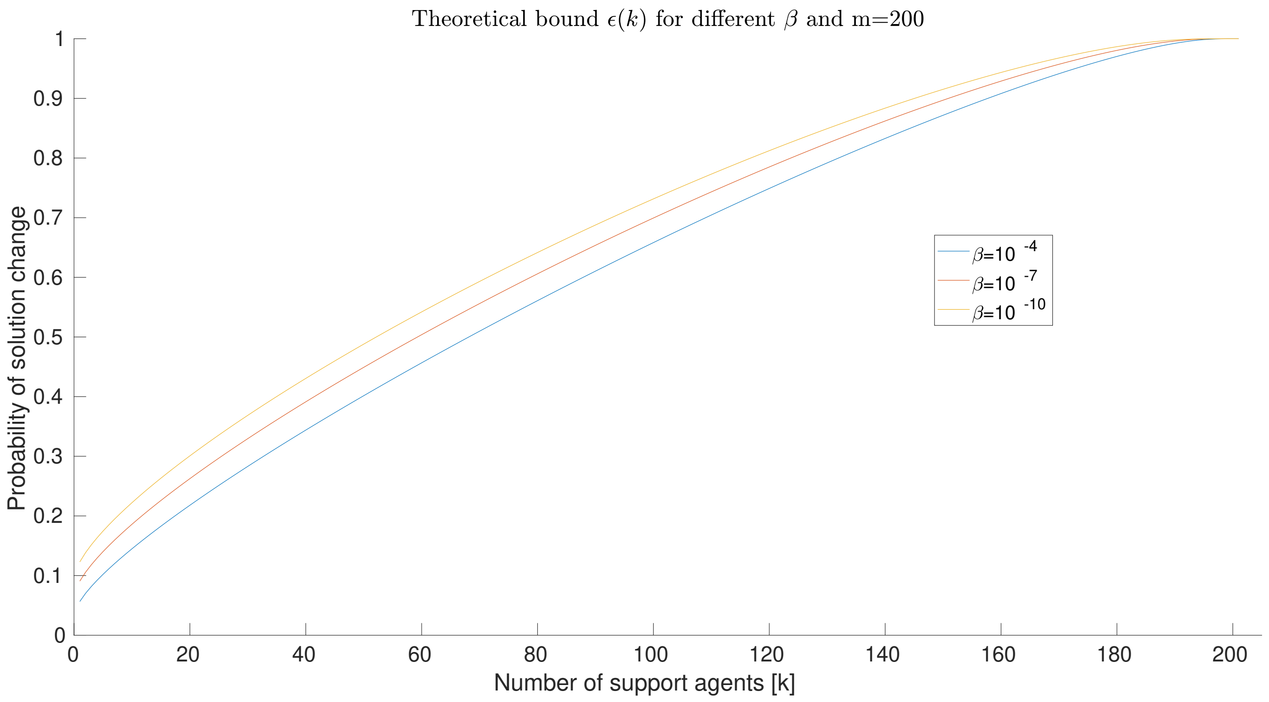

Figure 2: $\epsilon(k)$ for $m = 200$ and different values of β

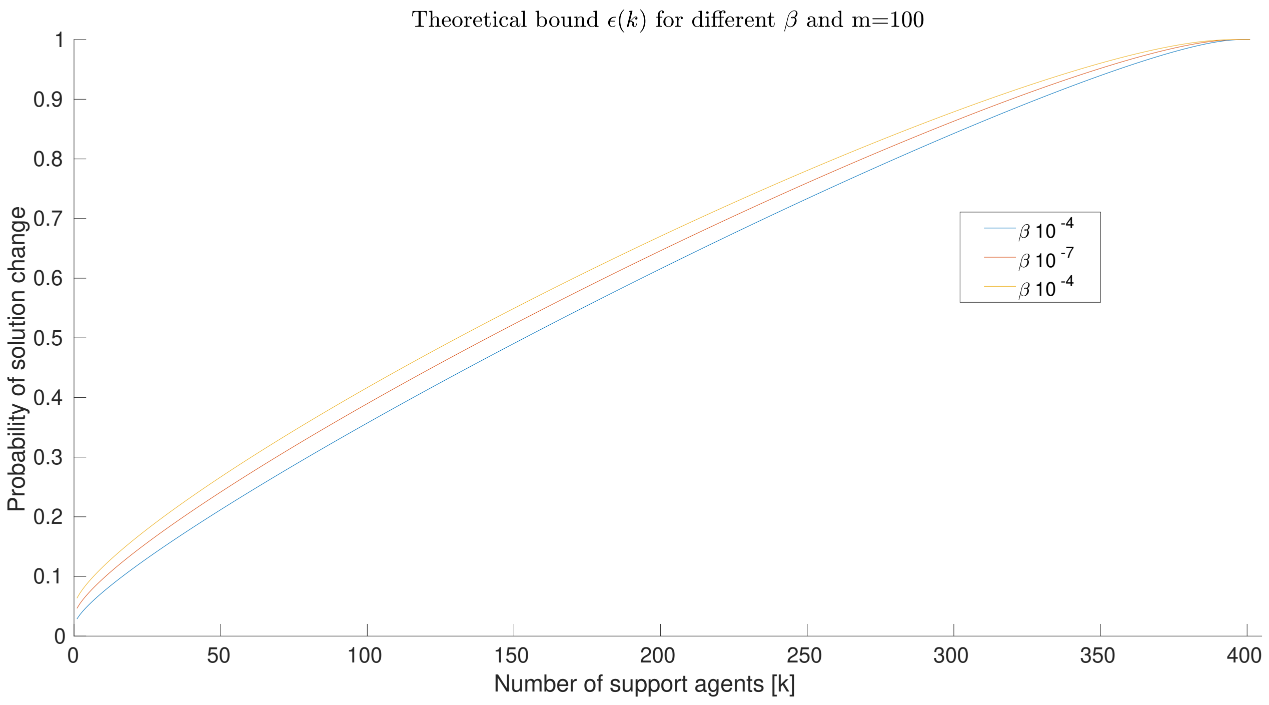

Figure 3: $\epsilon(k)$ for $m = 400$ and different values of β

By evaluating $\epsilon(k)$ at specific values of s^*, the number of support agents, it can be noted that the bound is lower for lower values of s^* and higher for higher values of s^*. This means that if many agents in the original problem contribute to its optimal solution x^* (high s^*), it is more likely to observe a new agent that brings improvements, vice-versa for a low number of support agents s^*: an higher s^* signifies that also agents with suboptimal cost coefficients among the initial m ones are used for finding x^* which are more likely to be replaced by a newly arrived one.

As an additional remark it is worth noting that the results of the classic scenario theory (18) were valid in chapter 2 because $s^* \leq p$ always, where p represents the number of coupling constraints of the problem, or the row rank of A. Now because of the presence of the upper bounds on x^i, this is not always true. In fact consider a problem without upper bounds:

$$\begin{aligned} \mathcal{P}_0: \quad \min_{x^i} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i = b \\ & x^i \geq 0 \qquad i = 1, ..., m, \end{aligned} \tag{32}$$

where $x^i \in \mathbb{R}^m$ and $A \in \mathbb{R}^{p \times m}$. Recalling theorem 2.4 of [6], there cannot be more than p x^i different from zero at a vertex of the polyhedron $\{\mathbf{x} \in \mathbb{R}^m | Ax = b, x \geq 0\}$. Since the optimal solution x^* of (32) occurs surely at a vertex, the number of nonzero components of x^*, s^* must be less or equal than p.

Now consider again the auxiliary practical problem introduced here in chapter 3:

$$\begin{aligned} \mathcal{P}_a: \quad \min_{x^i} \quad & \textstyle\sum_{i=1}^{110} c^i x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m x^i = 100 \\ & 0 \leq x^i \leq d^i \qquad i = 1, ..., 110, \end{aligned} \tag{33}$$

where $x^i \in \mathbb{R}$, $c^i \in \mathbb{R}$, $d^i \in \mathbb{R}$ and A^i is equal to 1 for all agents, implying that $p = 1$. In particular, all $d_i \in (10, 30)$. It is clear that in order to satisfy the equality constraint more than one x_i^* (i-th element of the optimal solution x^* of (33)) will have to be different from zero because of the upper bounds d^i, each one far smaller than $b = 100$, while $p = 1$. This clearly result in $s^* > p$ making it not possible to reuse the *a priori* bound provided by (18).

The rest of this chapter is dedicated to the proof of the fundamental Theorem 3.1 and is organized as follows:

- In Section 3.1 a characterization of the optimal solution for problems of the type of $\mathcal{P}_1$ is provided.

- In Section 3.2 the dual of $\mathcal{P}_1$, $\mathcal{D}_1$ is introduced and a connection between their optimal solutions is established.

- In Section 3.3 the scenario optimization theory is applied on $\mathcal{D}_1$ and then mapped back to $\mathcal{P}_1$ to obtain the main result.

3.1 Extended basic solution and optimality condition

In this section, a characterization of the optimal solution x^* of:

$$\begin{aligned} \mathcal{P}_1 : \min_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i = b \\ & 0 \leq x^i \leq d^i \quad i = 1, .., m, \end{aligned} \tag{34}$$

is provided. Let $n = \sum_{i=1}^m n_i$ the total number of decision variables in $\mathcal{P}_1$, $x = [(x^1)^T, \ldots, (x^m)^T]^T \in \mathbb{R}^n$, $A = [A^1 ... A^m] \in \mathbb{R}^{p \times n}$, $c = [(c^1)^T ... (c^m)^T]^T \in \mathbb{R}^n$, $d = [(d^1)^T, \ldots, (d^m)]^T \in \mathbb{R}^n$ and without loss of generality assume that A is full row-rank with $n > p$. Also a superscript to a vector describes its association with a corresponding agent, while a subscript denotes a particular element in that vector. The equality constraints in $\mathcal{P}_1$ along with the upper bounds d_i define a polyhedron, let us call it $Q = \{0 \leq x \leq d : Ax = b\}$. In order to characterize the optimal solution of $\mathcal{P}_1$, the concept of basic solution (Definition 2.1) has to be recalled and extended in the current context:

Definition 3.2. Extended basic solution - $\overline{x} \in \mathbb{R}^n$ is said to be a basic solution associated with $\mathcal{P}_1$ if $A\overline{x} = b$ and n linearly independent constraints are active at $\overline{x}$. It is an **extended basic solution** of $\mathcal{P}_1$ if in addition $\overline{x} \in [0, d]$.

The following proposition is a modified version of Theorem 2.4 of [6]:

Proposition 3.1: Consider the constraints $Ax = b$ and $0 \leq x \leq d$ and assume that the $p \times n$ matrix A has linearly independent rows. A vector $\overline{x} \in \mathbb{R}^n$ is an **extended basic solution** if and only if we have $A\overline{x} = b$, and there exist p indices $B(1), ..., B(p) \subset \{1, \ldots, n\}$ such that:

1. The columns $A_{B(1)}, ..., A_{B(p)}$ are linearly independent;
2. If $i \neq B(1), ..., B(p)$, then either $\overline{x}_i = 0$ or its respective upper bound d_i.

Proof: $\Longleftarrow$:Suppose that $\overline{x}$ satisfies $A\overline{x} = b$ and both conditions (1) and (2). Condition (2) clearly implies that there are at least $n - p$ active constraints. Then, it is true that:

$$\begin{aligned} b = A\overline{x} &= \sum_{i=1}^{p} A_{B(i)}\overline{x}_{B(i)} + \sum_{i \neq B(1),...B(p)} A_i\overline{x}_i, \\ \text{from which} \quad \sum_{i=1}^{p} A_{B(i)}\overline{x}_{B(i)} &= b - \sum_{i \neq B(1),...B(p)} A_i\overline{x}_i \end{aligned} \tag{35}$$

Since the columns $A_{B(i)}$ $i = 1, ..., p$ are linearly independent, then the $\overline{x}_{B(i)}$ $i = 1, ..., p$ are uniquely determined. By Theorem 2.2 of [6], this implies that there are p linearly independent active constraints, besides the already existing $n - p$ and this implies that $\overline{x}$ is a basic solution.

$\Longrightarrow$: For the converse, let us assume $\overline{x}$ is a basic solution. Let $B(1), ..., B(k)$ all the indices such that $\overline{x}_{B(i)} \neq 0$ and $\overline{x}_{B(i)} \neq d_i$ for all $i = 1, ..., k$.

Since $\overline{x}$ is a basic solution, the system of equations formed by the active constraints among the equality constraints $\sum_{i=1}^{m} A_i\overline{x}_i = b$, $\overline{x}_i = 0$, $\overline{x}_i = d_i$ $i \neq B(1), ...B(k)$ has a unique solution (Theorem 2.2 of [6]).

This implies that the columns $A_{B(1)}, ..., A_{B(k)}$ are linearly independent and so $k \leq p$. Since $rank(A) = p$, we can choose $p - k$ more columns so that the columns $A_{B(1)}, ..., A_{B(p)}$ are linearly independent. Moreover if $i \neq B(1), ..., B(p)$, then it is also true that $i \neq B(1), ..., B(k)$, since $k \leq p$, and hence $\overline{x}_i = 0$ or $\overline{x}_i = d_i$.

The previous proposition implies that any extended basic solution can be partitioned in three subvectors: $\overline{x}_B$ corresponding to the indices $B(1), \ldots, B(p)$ (subvector of basic variables), $\overline{x}_{Nl}$ (the subvector of non-basic variables at lower bound) and $\overline{x}_{Nu}$ (the subvector of non-basic variables at upper bound). The same subscripts applied to A and c will denote the corresponding index partitions from now on.

Being at one vertex of Q, the optimal solution to $\mathcal{P}_1$ is always an extended basic solution. The previous definition of extended basic solution is now used to provide a characterization of x^*.

Proposition 3.2 - Optimality condition (extended) Assume that $\mathcal{P}_1$ is feasible and admits a unique **non degenerate** minimizer. An extended basic solution x^v is the optimal solution x^* of $\mathcal{P}_1$ if and only if:

- $c_{Nl}^T - c_B^T A_B^{-1} A_{Nl} \geq 0$
- $c_{Nu}^T - c_B^T A_B^{-1} A_{Nu} \leq 0,$

where B, Nu and Nl is the indices partition corresponding to x^v. In order to have $x^v = x^*$ the reduced cost for any non-basic variable at lower bound must be greater or equal than 0 and vice-versa for any variable at upper bound.

Proof: Consider an extended basic solution x^v and another feasible point $x \in Q$ and let $z = x - x^v$. Since x and x^v are feasible, $Ax^v = b = Ax$, so $Az = 0$, which is equal to writing $A_B z_B + A_N z_N = 0$ with the N subscript including both the indices of N_l and of N_u.

Since A_B is a square matrix and its columns are linearly independent, it is true that $z_B = -A_B^{-1} A_N z_N$. Considering the cost value for z:

$$\begin{aligned}
&c^T z = c_N^T z_N + c_B^T z_B = \\
&c_N^T z_N - c_B^T A_B^{-1} A_N z_N = \\
&(c_N^T - c_B^T A_B^{-1} A_N) z_N = \\
&(c_N^T - c_B^T A_B^{-1} A_N)(x_N - x_N^v)
\end{aligned} \tag{36}$$

Now let us split the previous equation along the components Nl and Nu:

$$c^T z = (c_{Nu}^T - c_B^T A_B^{-1} A_{Nu})(x_{Nu} - x_{Nu}^v) + \tag{37}$$

$$+(c_{Nl}^T - c_B^T A_B^{-1} A_{Nl})(x_{Nl} - x_{Nl}^v) \tag{38}$$

Regarding (37) note that , $(x_{Nl} - x^v_{Nl}) \geq 0$ since x^v_{Nu} contains variables whose value is the lowest possible (x^v_{Nl} are at lower bound), while regarding (38) $(x_{Nu} - x^v_{Nu}) \leq 0$ since x^v_{Nu} contains variables whose value is the highest possible (x^v_{Nl} are at upper bound). In order for x^v to be optimal it must be that $c^T z \geq 0 \; \forall x \in Q$ (which means $c^T x - c^T x^v \geq 0$ for all x in Q). The sign conditions $(x_{Nl} - x^v_{Nl}) \geq 0$ and $(x_{Nu} - x^v_{Nu}) \leq 0$ along with the expression for $c^T z$ in (37) and (38) imply that it must be $(c^T_{Nu} - c^T_B A^{-1}_B A_{Nu}) \geq 0$ and $(c^T_{Nu} - c^T_B A^{-1}_B A_{Nu}) \leq 0$.

3.2 Dual problem

As anticipated in the introduction, the proof of Theorem 3.1 is based on inheriting some properties possessed by the solution of the dual problem of $\mathcal{P}_1$. In this subsection the connections between $\mathcal{P}_1$ and its dual problem are established. The dual of $\mathcal{P}_1$, $\mathcal{D}_1$ is:

$$\begin{array}{lll} \mathcal{D}_1 : \min\limits_{\lambda \in \mathbb{R}^p, \{\nu^i \in \mathbb{R}^{n_i}\}_{i=1}^m} & \lambda^T b + \sum_{i=1}^m (\nu^i)^T d^i & \\ \text{s.t.} & \lambda^T A^i + c^i \geq -\nu^i & i = 1, .., m \\ & \nu^i \geq 0 & i = 1, .., m \end{array} \tag{39}$$

where $\lambda \in \mathbb{R}^p$ is the dual vector associated to the resource allocation equality constraints, while $v^i \in \mathbb{R}^{n_i}$, $i = 1, \ldots, m$ are the dual vectors associated to the upper limit constraints on x^i.

Letting $\nu = [(\nu^1)^T, \ldots, (\nu^m)^T]$, the optimal solution to $\mathcal{D}_1$ is denoted by (λ^*, ν^*).

The main purpose of the following paragraph is to prove the proposition:

Proposition 3.3 - Characterization of the dual solution The optimal solution to $\mathcal{D}_1$ is (λ^*, ν^*) where:

- $\lambda^* = -(c^T_B A^{-1}_B)^T$

- $\nu^*_i = 0 \qquad$ if $i \in Nl$ or $i \in B$

- $\nu_i^* = -(c_i - c_B^T A_B^{-1} A_I) := -\overline{c_i} \qquad \text{if } i \in Nu$

Proof: Firstly, the feasibility of the solution will be proved, then the optimality. Considering the dual constraint at $\lambda^* = -(c_B^T A_B^{-1})^T$, $\lambda^{*T} A_i + c_i \geq -\nu_i$, it is satisfied for $\nu_i = 0$, $i \in N_l$, since $c_{Nl}^T - c_B^T A_B^{-1} A_{Nl} \geq 0$. The constraint also holds for $\nu_i = 0$, $i \in B$ since $c_{Nl}^T - c_B^T A_B^{-1} A_{Nl} = 0$. Lastly $\lambda^{*T} A_i + c_i \geq -\nu_i$ is certainly satisfied for $\nu_i = -\overline{c}_i$, $i \in Nu$, since it results in $c_{Nu}^T - c_B^T A_B^{-1} A_{Nu} \geq -\overline{c_i}$ where $-\overline{c}_i \geq 0$ for $i \in Nu$ (Recall Proposition 3.2). We then can select a suitable $\nu^*, \nu_i = 0 \ \forall i \in N_l$ such that the remaining constraints λ^*, $\lambda^{*T} A_i + c_i \geq -\nu_i, i \in N_u$ are satisfied. Now, to verify the optimality (λ^*, ν^*), it must satisfy together with the optimal solution of $\mathcal{P}_1$, x^* the following **complementary slackness conditions** (for upper bound problems [9]):

$$\begin{aligned} &x_i(\lambda^T A_i + \nu_i + c_i) = 0 \qquad i = 1, ..., n \\ &\nu_i(d_i - x_i) = 0 \qquad i = 1, ..., n \end{aligned} \tag{40}$$

If and only if these condition (40) are verified, the couple of solutions x^*, (λ^*, ν^*) is optimal. Given an extended basic feasible solution x^* that verifies the optimality conditions seen before and the previously introduced (λ^*, ν^*), they indeed do. $\nu_i(d_i - x_i) = 0$ is always verified since $\nu_i = 0$,$\forall i \in Nl \cup B$, corresponding to the x_i that are either 0 or a value between 0 and their upper bound d_i, while for $i \in Nu$ $x_i = d_i$.
$x_i(\lambda^T A_i + \nu_i + c_i) = 0$ is verified for $i \in Nl$ since $x_i = 0$ for $i \in Nl$.
For $i \in B$, substituting $\lambda^* = -(c_B^T A_B^{-1})^T$ and remembering that $c_{Nl}^T - c_B^T A_B^{-1} A_{Nl} = 0$, the first condition is again satisfied.
Lastly for $i \in Nu$ by substituting $\lambda^* = -(c_B^T A_B^{-1})^T$ and $\nu_i^* = -(c_{Nu} - c_B^T A_B^{-1} A_{Nu})$ in the first condition it becomes $x_{Nu}(-c_B^T A_B^{-1} A_{Nu} - c_{Nu} + c_B^T A_B^{-1} A_{Nu} + c_{Nu}) = 0$. This concludes the proof.
For the sake of completeness the optimality of (λ^*, ν^*) will be proven also by showing that the duality gap is equal to 0 for this solution.

Consider again the dual problem $\mathcal{D}_1$, rewritten to be a maximization one:

$$\begin{aligned} \mathcal{D}_1: \max_{\lambda\in\mathbb{R}^p,\{\nu^i\in\mathbb{R}^{n_i}\}_{i=1}^m} \quad & -\lambda^T b - \textstyle\sum_{i=1}^m (\nu^i)^T d^i \\ \text{s.t.} \quad & \lambda^T A^i + c^i \geq -\nu^i \qquad i=1,..,m \\ & \nu^i \geq 0 \qquad\qquad\quad i=1,..,m \end{aligned} \tag{41}$$

Remembering that $\nu^*_{Nu} = -\bar{c}_{Nu}$, $\nu^*_{Nl} = 0$ and $\nu^*_B = 0$, the cost function of the dual problem evaluated at (λ^*, ν^*) reads:

$$\begin{aligned} &(c_B^T A_B^{-1})b - (-\bar{c}_{Nu}^T d_{Nu}) = \\ &(c_B^T A_B^{-1})b + c_{Nu}^T d_{Nu} - c_B^T A_B^{-1} A_{Nu} d_{Nu} = \\ &c_B^T A_B^{-1}(b - A_{Nu} d_{Nu}) + c_{Nu}^T d_{Nu} \end{aligned} \tag{42}$$

Now consider the optimal solution of the primal problem x^* (remembering that $x^*_{Nl} = 0$ and $x^*_{Nu} = d_{Nu}$), since it is feasible it is true that:

$$\begin{aligned} Ax^* =& A_B x_B^* + A_{N_u} x^*_{N_u} = \\ & A_B x_B^* + A_{Nu} d_{Nu} = b \end{aligned} \tag{43}$$

From (43) an expression for x_B^* can be derived:

$$x_B^* = A_B^{-1}(b - A_{Nu} d_{Nu}), \tag{44}$$

and substituting (44) in (42), the dual cost associated to (λ^*, ν^*) can be rewritten as:

$$c_B^T A_B^{-1}(b - A_{Nu} d_{Nu}) + c_{Nu}^T d_{Nu} = c_B^T x_B^* + c_{Nu}^T d_{Nu} \tag{45}$$

The right-hand side of (45) is however equal to $c^T x^* = c_B^T x_B^* + c_{Nl}^T x_{Nl}^* + c_{Nu}^T x_{Nu}^* = c_B^T x_B^* + c_{Nu}^T d_{Nu}$ because $x_i^* = d_i$ for $i \in Nu$ and $x^*_{Nl} = 0$.
Thus, it is shown that the value of the cost function of the primal and the dual are the same and the optimality of (λ^*, ν^*) is proven by strong duality.

3.3 New agent arrival

The results of the previous sections are now used to prove Theorem 3.1.

Consider now the problem $\mathcal{P}_{1_+}$:

$$\begin{aligned} \mathcal{P}_{1_+}: \min_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m, y \in \mathbb{R}^{n_y}} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i + (c^y)^T y \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i + A^y y = b \\ & 0 \leq x^i \leq d^i \qquad i = 1,..,m \\ & 0 \leq y \leq d^y, \end{aligned} \tag{46}$$

where the symbol x_+ denotes the augmented vector of decision variables (x, y).

Applying the results in Section 3.1 to $\mathcal{P}_{1_+}$, it holds that $x_+^\star = (x^*, 0)$, i.e. the optimal value for x in $\mathcal{P}_{1_+}$ coincides with the optimal solution to $\mathcal{P}_1$ and the optimal value of y is 0, if and only if all the reduced costs associated with the indices of the new variable y are ≥ 0 (see Proposition 3.2). That is, in compact notation:

$$\overline{c_y} = (c^y)^T - c_B^T A_B^{-1} A^y \geq 0. \tag{47}$$

On the other hand, recalling that $\lambda^* = -(c_B^T A_B^{-1})^T$ as given by Proposition 3.3, (47) is true if and only if

$$\lambda^{*T} A^y + c^y \geq 0, \tag{48}$$

i.e if the solution to the dual λ^* satisfies a new constraint associated to the new agent y.

In conclusion, it can be said that:

$$\begin{aligned} &\mathbb{P}\big\{\delta^y = (n_y, c^y, A^y, d^y) \in \Delta : \; x_+^\star = (x^*, 0)\big\} = \\ &\mathbb{P}\big\{\delta^y = (n_y, c^y, A^y, d^y) \in \Delta : \; (\lambda^*)^\top A^y + c^y \geq 0\big\}. \end{aligned} \tag{49}$$

The left-hand side of (49) is equal to 1 minus the stability index of x^* (1-$V(x^*)$), to which a bound has to be found, while the right-hand side is the probability of extracting a new constraint of the dual problem which is satisfied by the optimal dual solution λ^*. This probability (the right-hand side one)is itself a random variable because it varies with λ^*, which is random.

The scenario optimization theory deals with the problem of bounding $\mathbb{P}\big\{\delta^y = (n_y, c^y, A^y, u^y) \in \Delta : \ (\lambda^*)^\top A^y + c^y \geq 0\big\}$, that is why it was necessary to characterize the optimal solution of $\mathcal{D}_1$. More recent results from the *Wait-and-judge* scenario optimization [4] guarantee that by taking $\beta \in (0,1)$ the following polynomial equation has a unique solution in the variable t:

$$\frac{\beta}{m+1}\sum_{i=k}^{m}\binom{i}{k}t^{i-k} - \binom{m}{k}t^{m-k} = 0, \tag{50}$$

for $k = 0, \dots, n-1$ where n is the total number of decision variables of $\mathcal{P}_1$.

By considering the unique solution to (50) $t(k) \in (0,1)$, recalling (49) and letting $\epsilon(k) = 1 - t(k)$ [4], [8] guarantees that:

$$\begin{aligned}&\mathbb{P}^m\big\{(\delta^1, \dots, \delta^m) \in \Delta^m : \\ &\mathbb{P}\big\{\delta^y = (n_y, c^y, A^y, d^y) \in \Delta : \ (\lambda^*)^\top A^y + c^y \geq 0\big\} \geq 1 - \epsilon(s^*)\big\} \geq 1 - \beta,\end{aligned} \tag{51}$$

where λ^* is the solution to (39) , the dual problem of $\mathcal{P}_1$ $\delta^y = (n_y, c^y, A^y, d^y)$ is the tuple associated with the new agent y and $\epsilon(s^*)$ is the probability bound $\epsilon(k) = 1 - t(k)$ evaluated at the number of *support constraints* at λ^*, s^*, that is the number of constraints out of all of the dual problem $\mathcal{D}_1$ for which it does not hold $(\lambda^*)^T A^i + c^i > 0$, but rather $(\lambda^*)^T A^i + c^i = 0$ or $(\lambda^*)^T A^i + c^i < 0$ element-wise. This means that $\exists$ an index j such that

$$(\lambda^*)^T A_j + c_j < 0 \text{ or } = 0, \tag{52}$$

where A_j represents the $j-th$ column of A_i and c_j the $j-th$ element of c_i.

By recalling (49) and applying it to (51), it holds true that:

$$\begin{aligned}&\mathbb{P}^m\big\{(\delta^1,\dots,\delta^m)\in\Delta^m:\\&\mathbb{P}\big\{\delta^y=(n_y,c^y,A^y,d^y)\in\Delta:\ x_+^\star=(x^*,0)\big\}\geq 1-\epsilon(s^*)\big\}\geq 1-\beta,\end{aligned}\tag{53}$$

where $x_+^\star$ is the optimal solution of (46) and x^* of $\mathcal{P}_1$, while s^* still denotes the number of *support constraints* of $\mathcal{D}_1$.

The concept of *support constraints* defines the constraints that if removed bring an improvement to the optimal solution of the problem. The number of support agents is actually equal to the number of support constraints. To prove the previous statement it is sufficient to substitute $\lambda^*=-(c_B^TA_B^{-1})^T$ into $(\lambda^*)^TA^i+c^i$, obtaining $-c_B^TA_B^{-1}A_j+c_j=\bar{c}_i$ for $j=1,\dots,n_i$, the reduced cost for component $(x^i)^*$ of x^*. (52) holds true if and only if $-c_B^TA_B^{-1}A_j+c_j<0$ or $-c_B^TA_B^{-1}A_j+c_j=0$, meaning that either $j\in Nu$ or $j\in B$, respectively. By denoting with x_j^* the $j-th$ element of $(x^i)^*$, it holds true that $x_j^*=d_j$ if $j\in Nu$ or $x_j^*\neq d_j\geq 0$ if $j\in\ B$, and so according to Definition 3.1 $(x^i)^*\neq 0$ because at least one of its elements are different from zero. So in conclusion $(x^i)^*\neq 0$ if and only if the associated $(\lambda^*)^TA^i+c^i<0$ or 0, proving that indeed the number of support agents and the number of support constraints are equal.

Employing the result in (53) it is possible, after computing the optimal solution x^* of $\mathcal{P}_1$ and therefore s^* to say that the probability that x^* remains feasible upon the arrival of the new agent y is at least $1-\epsilon(s^*)$, with confidence of at least $1-\beta$, where β is user chosen and measure with respect to $\mathbb{P}^m$.

4 Inequality constrained problems

This chapter will consider problem $\mathcal{P}_2$:

$$\begin{aligned} \mathcal{P}_2 : \min_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i \leq b \\ & 0 \leq x^i \leq d^i \qquad i = 1, .., m, \end{aligned} \tag{54}$$

whose optimal solution is x^* and where the initial m agents are subject to an upper bound d_i to the quantity of b they can be assigned and moreover the shared resource b can also not be used fully. Again, the results of Chapter 2 are not valid in this situation for the same reasons expressed in Chapter 3 and $V(x^*)$, where x^* represents the optimal solution of $\mathcal{P}_2$ will be characterized by using results from the *Wait and judge* scenario approach theory.

The result of this chapter, proven in the later sections is really similar to the one of Chapter 3, stated in the following theorem:

Theorem 4.1. Let $\beta \in (0,1)$ and let $\epsilon(k) \in (0,1)$ for $k = 0, 1, \ldots, m$ be equal to $1 - t(k)$ where $t(k)$ for $k = 0, 1, \ldots, m$ is the unique solution of the polynomial equation

$$\frac{\beta}{m+1} \sum_{i=k}^{m} \binom{i}{k} t^{i-k} - \binom{m}{k} t^{m-k} = 0, \tag{55}$$

in the interval (0,1) and $t(m) = 0$. Then it holds that:

$$\mathbb{P}^m \big\{ (\delta^1, \ldots, \delta^m) \in \Delta^m : V(x^*) < \epsilon(s^*) \big\} \geq 1 - \beta, \tag{56}$$

where x^* is the optimal solution to $\mathcal{P}_1$ and s^* is the number of support agents of x^*.

where the number of support agents of x^*, s^* is defined in Definition 3.1. The rest of this chapter is structured in a similar way to Chapters 2 and 3, and will go on with the characterization of x^* in section 4.1, the primal dual connection in section 4.2 and finally recall the *Wait-and-judge* scenario optimization theory [4], [8] to provide a proof of Theorem 4.1.

4.1 Optimality conditions

The previous results of Chapter 3 can be easily extended to problems with inequality constraints.

Consider the problem:

$$\begin{aligned} \mathcal{P}_2 : \max_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i \leq b \\ & 0 \leq x^i \leq d^i \qquad i = 1, .., m, \end{aligned} \tag{57}$$

whose solution is x^* and let $n = \sum_{i=1}^m n_i$ the total number of decision variables in $\mathcal{P}_2$, $x = [(x^1)^T, \ldots, (x^m)^T]^T \in \mathbb{R}^n$, $A = [A^1 ... A^m] \in \mathbb{R}^{p \times n}$, $c = [(c^1)^T ... (c^m)^T]^T \in \mathbb{R}^n$, $d = [(d^1)^T, \ldots, (d^m)]^T \in \mathbb{R}^n$ and without loss of generality assume that A is full row-rank with $n > p$.

By simply adding a slack variable (s) per row of A:

$$\begin{aligned} \mathcal{P}_2 : \max_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m, s} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i + s = b \\ & 0 \leq x^i \leq d^i \qquad i = 1, .., m, \\ & s \geq 0 \end{aligned} \tag{58}$$

and by bringing the slack variables inside the decision variables vector:

$$\begin{aligned} \mathcal{P}_2 : \max_{\tilde{x}^i} \quad & \textstyle\sum_{i=1}^{m+1} (\tilde{c}^i)^T \tilde{x}^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{m+1} \tilde{A}^i \tilde{x}^i = b \\ & 0 \leq x^i \leq d^i \qquad i = 1, .., m, \\ & s \geq 0 \end{aligned} \tag{59}$$

where

$$\tilde{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ s \end{bmatrix} \qquad \tilde{A} = \begin{bmatrix} A & \mathbb{1}_p^T \end{bmatrix} \qquad \tilde{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_m \\ c_s \end{bmatrix} \tag{60}$$

with $\mathbb{1}_p^T$ indicating a column vector $\in \mathbb{R}^p$ containing all ones as elements, c_s the cost coefficient for the slack variable s, always equal to 0 and $\tilde{x}^*$ its optimal soluton. Proposition 3.1 can now be applied and the same definition for reduced costs and conditions for optimality hold (provided $\tilde{x}^*$ is not **degenerate**).

Since $\mathcal{P}_2$ it is a maximization problem, the reduced costs at an optimal solution must be:

- $c_{Nl}^T - c_B^T A_B^{-1} A_{Nl} \leq 0$
- $c_{Nu}^T - c_B^T A_B^{-1} A_{Nu} \geq 0$

4.2 Dual Problem

Consider now the dual of $\mathcal{P}_2$:

$$\begin{aligned} \mathcal{D}_2 : \min_{\lambda \in \mathbb{R}^p, \{\nu^i \in \mathbb{R}^{n_i}\}_{i=1}^m} \quad & -\lambda^T b - \textstyle\sum_{i=1}^m (\nu^i)^T d^i \\ \text{s.t.} \quad & \lambda^T A^i + \nu^i - c^i \geq 0 \qquad i = 1, .., m \\ & \nu^i \geq 0 \qquad i = 1, .., m \\ & \lambda_j \geq 0 \qquad j = 1, ..., p \end{aligned} \tag{61}$$

whose optimal solution is (λ^*, ν^*) which is going to be characterized in the next proposition:

Proposition 4.1. - Characterization of the dual solution The optimal solution to $\mathcal{D}_2$ is (λ^*, ν^*) where:

- $\lambda^* = (c_B^T A_B^{-1})^T$

- $\nu_i^* = 0 \qquad$ if $i \in Nl$ or $i \in B$

- $\nu_i^* = (c_i - c_B^T A_B^{-1} A_i) := -\overline{c_i} \qquad$ if $i \in Nu$

Proof:

To prove that (λ^*, ν^*) is feasible first assume that c and A^i are ≥ 0 element wise, meaning $\lambda^* = (c_B^T A_B^{-1})^T \geq 0$ which is reasonable since a negative A^i would mean that the agent i is bringing in more resource and a negative c^i would mean that the agent gets discarded immediately from the optimal solution since lowers the profit.

By substituting $\lambda^* = (c_B^T A_B^{-1})^T$ and ν_i^* into $\lambda^T A_i + \nu_i - c_i \geq 0$ one obtains $c_B^T A_B^{-1} A_i + \nu_i^* - c_i \geq 0$ which is satisfied for $i = B$ since it turns into $c_B^T A_B^{-1} A_i + -c_B = 0$, is satisfied for $i \in Nl$ since it becomes $c_B^T A_B^{-1} A_{Nl} - c_{Nl} = -\overline{c}_{Nl} \geq 0$ and is satisfied for $i \in Nu$ since $c_B^T A_B^{-1} A_{Nu} + \overline{c}_{Nu} - c_{Nu} = 0$.

Similarly to the previous case, the optimality of (λ^*, ν^*) is guaranteed if and only if it satisfies the complementary slackness conditions for problem $\mathcal{P}_2$, which in turn can be derived from its Lagrangian function $\mathcal{L}(x, \lambda, \nu)$:

$$\mathcal{L}(x, \lambda, \nu) = c^T x + \lambda^T (b - Ax) + \nu(d - x) \tag{62}$$

The first two are highlighted in red and are found immediately by writing $\mathcal{L}(x, \lambda, \nu)$. To find the last one, let us collect by x, obtaining the dual Lagrangian function:

$$\mathcal{L}_d(x, \lambda, \nu) = x(\lambda^T A + \nu - c) + \nu d + \lambda^T b \tag{63}$$

The conditions (1 for each dual and primal inequality constraint) read:

$$
\begin{aligned}
& x_i(\lambda^T A_i + \nu_i - c_i) = 0 \qquad i = 1, ..., m \\
& \lambda_j^T(b_j - A_j x) = 0 \qquad j = 1, .., p \text{ (condition on rows of A)} \\
& \nu_i(d_i - x_i) = 0 \qquad i = 1, ..., m
\end{aligned}
\tag{64}
$$

Indeed $\lambda^* = (c_B^T A_B^{-1})^T$ and ν^* such that:

- $\nu_i = 0$ if $i \in N_l$ or $i \in B$
- $\nu_i = \overline{c_i}$ for $i \in N_u$ ($\overline{c_i}$ is the reduced cost for variable x_i,which is ≥ 0 for $i \in N_u$)

satisfy these conditions, rendering (λ^*, ν^*) optimal for $\mathcal{D}_2$. In fact, $\nu_i(d_i - x_i) = 0$ is verified for $i \in B$ and $i \in Nl$ since $\nu_i^* = 0$ and for $i \in Nu$ since $x_i = d_i$. $x_i(\lambda^T A_i + \nu_i - c_i) = 0$ holds for $i \in Nl$ since $x_i = 0$ holds for $i \in B$ since $(c_B^T A_B^{-1} A_i - c_B) = 0$ with $\nu_i^* = 0$ for $i \in B$. For $i \in Nu$ $(c_B^T A_B^{-1} A_i + \overline{c}_{Nu} - c_{Nu}) = 0$.
For the second condition $\lambda_j^T(b_j - A_j x) = 0$ if the $j-th$ equality constraint is active it is true that $(b_j - A_j x) = 0$, while if it is not active, the $j-th$ component of the slack variable s, s_j is different from zero, so s_j, not having an upper bound, becomes one of the p variables that are neither at 0 nor at their upper bounds (recall Proposition 3.1), as x_i for $i \in B$. So c_B will include the cost coefficient for s_j, $c_{s_j} = 0$, rendering the $j-th$ component of $\lambda^* = (c_B A_B^{-1})$ equal to zero, verifying the condition.

4.3 New agent arrival

Consider now the problem $\mathcal{P}_{2_+}$:

$$
\begin{aligned}
\mathcal{P}_{2_+}: \quad \max_{\{x^i \in \mathbb{R}^{n_i}\}_{i=1}^m, y \in \mathbb{R}^{n_y}} \quad & \textstyle\sum_{i=1}^m (c^i)^T x^i + (c^y)^T y \\
\text{s.t.} \quad & \textstyle\sum_{i=1}^m A^i x^i + A^y y \leq b \\
& 0 \leq x^i \leq d^i \qquad i = 1, ..., m \\
& 0 \leq y \leq d^y
\end{aligned}
\tag{65}
$$

Its solution $x^\star_+ = (x^*, 0)$, $(y^\star = 0)$ if and only if the reduced cost of the new variable y,

$$\overline{c_y} = (c^y)^T - c_B^T A_B^{-1} A^y \leq 0 \tag{66}$$

But recalling that $\lambda^* = (c_B^T A_B^{-1})^T$, (66) is true if and only if

$$\lambda^{*T} A^y - c^y \geq 0 \tag{67}$$

Since $y^\star = 0$, its associated $\nu = 0$ and its reduced cost is lesser than or equal to 0.
In conclusion, similarly to chapter 3, it can be said that:

$$\begin{aligned} &\mathbb{P}\big\{\delta^y = (n_y, c^y, A^y, d^y) \in \Delta : \; x^\star_+ = (x^*, 0)\big\} = \\ &\mathbb{P}\big\{\delta^y = (n_y, c^y, A^y, d^y) \in \Delta : \; (\lambda^*)^\top A^y - c^y \geq 0\big\}. \end{aligned} \tag{68}$$

Since for problem (57) $s^* \leq p$ is again not always guaranteed, the *Wait and judge* scenario approach will be used to provide a bound for the probability of solution change. By taking $\beta \in (0,1)$ and evaluating the number of support constraints $s^*_m$of the optimal solution x^* of (57) one can retrieve (s^*) from (50) and compute $\epsilon(s^*)$ such that:

$$\begin{aligned} &\mathbb{P}^m\big\{(\delta^1, \ldots, \delta^m) \in \Delta^m : \\ &\mathbb{P}\big\{\delta^y = (n_y, c^y, A^y, d^y) \in \Delta : \; (\lambda^*)^\top A^y + c^y \geq 0\big\} \geq 1 - \epsilon(s^*)\big\} \geq 1 - \beta, \end{aligned} \tag{69}$$

where λ^* is the optimal solution of $\mathcal{D}_2$. Since (68) holds, (69) can be rewritten as:

$$\begin{aligned} &\mathbb{P}^m\big\{(\delta^1, \ldots, \delta^m) \in \Delta^m : \\ &\mathbb{P}\big\{\delta^y = (c^y, A^y, d^y) \in \Delta : \; x^\star_+ = (x^*, 0)\big\} \geq 1 - \epsilon(s^*)\big\} \geq 1 - \beta, \end{aligned} \tag{70}$$

and similar conclusions to the ones expressed in Chapter 3 can be formulated.

5 Application to economic dispatch problem

In this chapter (and in the next one) the results of chapter 3 and 4 will be applied through two practical examples. In particular, the theoretical results will be verified experimentally and given specific interpretations depending on the context.

5.0.1 Problem introduction

Ever-rising costs of fossil fuels and competitive pressure make power system planning a really important phase in energy production.
The economic dispatch problem is a planning procedure carried out by an external dispatch company that has to split a forecasted load among several energy producers in a geographical area. It consists in determining the optimal power output from available energy generation facilities so as to minimize the production cost while meeting the required load and considering also likely limits of the power plants which include:

- Ramp rate (how quickly the generators output can be changed)
- Maximum and minimum generation levels
- Minimum amount of time the generator must run
- Minimum amount of time the generator must stay off once turned off
- Whether there are transmission problems for a plant or not

In general, costs are not limited to the fuel ones, but should also account for losses along the transmission line, environmental compliance and generator startup. Speaking of environmental impact, some producers might employ also renewable sources that generally have (except for hydroelectric generators) lower electricity generation capacity compared to fossil fuels or nuclear plants while having near zero ecological footprint.
Planning is generally a daily task but can be performed multiple times or for shorter periods in order to increase efficiency or in case of increasing demand and is generally preceded by a reliability assessment that ensures the correctness of the load forecast

and the integrity of the transmission line for the units that are about to be employed. In this section a mathematical formulation of the planning problem is presented and a linear programming approximation is proposed [10], which yields the model used in the simulations. Since the point of this case study is not to provide an accurate model for real life power planning but rather to visualize the theoretical results of chapter 3, the only constraint considered for the generators is the maximum and minimum power one.

In this chapter the scenario approach theory will be employed to evaluate a situation where the planning is initially done on a small group of already known power plants although many others exist and are not initially taken into consideration. The energy dispatch company is willing to consider new producers in the hope that a more advantageous generator is discovered and the energy dispatch solution is improved. On the other hand obtaining information on previously not considered power plants is a costly and time consuming operation so that starting a campaign to augment the pool of producers among which to dispatch the energy load requires an assessment of its effectiveness. To this purpose, the scenario approach theory will be used to give an estimate of the probability that the solution might be improved by adding a previously unseen power plant to the initial pool of producers. In other words, the effectiveness of a new poll to collect information about power production facilities previously not taken into account is assessed in the probabilistic framework developed in this thesis.

5.1 Quadratic form problem

An energy dispatch company has information about various power plants and to keep the fuel costs low it will prioritize the most efficient or less expensive plants for the production of electrical energy. The energy generation cost curve of a generic generator i is ideally similar to a half parabola and can be modeled as it follows:

$$F^i = a^i(P^i)^2 + b^iP^i + c^i, \tag{71}$$

where F^i is the cost needed for plant i to produce power P^i and a^i, b^i and c^i are cost coefficients, all positive. Each generator has a minimum and maximum power it can output, say $P^{i,min}$ and $P^{i,max}$, so the cost curve will be bound between these values. Figure 4 shows an example of a generator cost curve. The superlinear increasing trend has an intuitive appeal, since usually producing more energy results in higher and additional costs. Moreover a case where the dispatch company only pays for energy production costs is considered, not accounting for startup costs or costs arising from the plant production reserved to other dispatch companies. As a result all c^i coefficients (representing generator startup and already existing costs) are equal to zero and a situation is considered where the dispatch company may not avail of a given producer. This amounts to setting the minimum power for all generators $P^{i,min} = 0, \forall i$.
Lastly, the sum of all the powers produced by the generators must be equal to the required load. This is a strict requirement since producing more is highly disadvantageous since storing the excess energy is extremely expensive and is possible for a very limited amount of energy. On the other hand, producing less than required would result in a severe disruption for the final users. Thus, summarizing, the energy dispatch problem consists in selecting P^i, the production of each energy generator, so as to minimize the production costs $\sum F^i$, while satisfying the load equality constraint and the production limits for all generators.
In formulas, the resulting minimization problem reads :

$$\begin{aligned} \min_{P^i} \quad & \textstyle\sum_{i=1}^N a^i (P^i)^2 + b^i P^i + c^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^N P^i = Load \\ & P^{imin} \leq P^i \leq P^{imax} \qquad i = 1, ..., N \end{aligned} \tag{72}$$

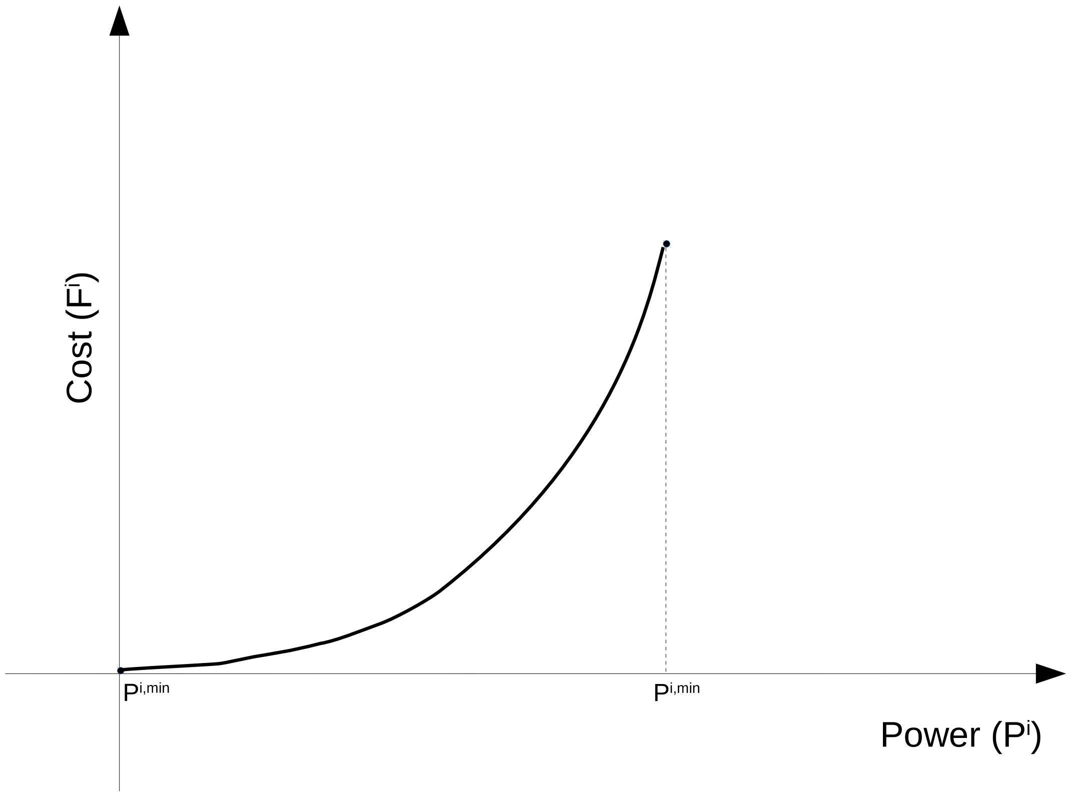

Figure 4: Cost curve

5.2 Linear programming reformulation

In order to make use of linear programming theory, a linear approximation of the cost function of the optimization model (72) introduced in the previous section is hereby proposed. Specifically, the parabolic curves are approximated with piecewise linear functions whose fit depends on the numbers of segments it is made of. A visual example of such an approximation is shown in Figure 5 where, for instance, the cost curve for generator i is approximated with four connected segments. Each of the m_i segments for each generator i is defined over an interval $[\overline{P}^i_j, \overline{P}^i_{j+1}]$ where $\overline{P}^i_1, \ldots .\overline{P}^i_{m_i+1}$ is a partition of $[P^{i,min}, P^{i,max}]$ with $\overline{P}^i_1 = P^i_{min}$ and $\overline{P}^i_{m_i+1} = P^{i,max}$.

To obtain a linear optimization problem, then, the original variable P^i in (72) is replaced by m_i decision variables $x^i_j,\ j = 1, \ldots, m_i$, where m_i is the number of segments used for the approximation, while $x^i = [x^i_1, \ldots, x^i_N]^T$ will denote the vector of the new decision variables for generator i. Each x^i_j represents the part of power output P^i within the interval $[\overline{P}^i_j, \overline{P}^i_{j+1}]$ and as such its value is bounded between 0 and $\overline{P}^i_{j+1} - \overline{P}^i_j = \overline{x^i_j}$. In particular the x^i_j have to be thought of as a decomposition of P^i and it must be that $P^i = \sum_{j=1}^{m_i} x^i_j$ and

$$x^i_j = 0 \quad \text{if} \quad x^i_j < \overline{P}^i_{j+1} - \overline{P}^i_j \tag{73}$$

This constraint means that the total energy produced cannot be obtained from the various segments in the partition independently. A segment is activated only if the allowance of the previous ones is exhausted.

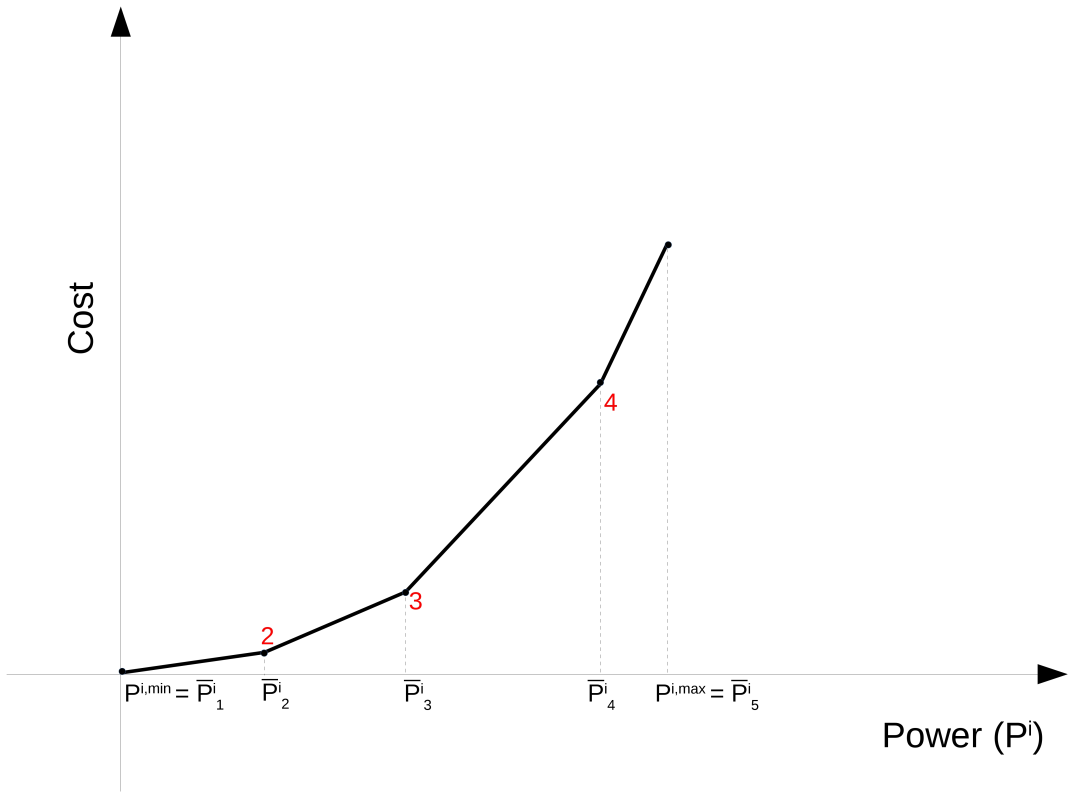

Figure 5: Approximated cost curve

In the reformulation of (72), with respect to the new variables, the cost coefficient $c^i_j,\ j = 1, \ldots, m_i$ for each x^i_j is the slope of the corresponding segment in the cost curve approximation resulting in the total cost $\sum_{j=1}^{m_i} c^i_j x^i_j$ for generator i. Given that the approximated function is a convex one, the slope of each segment is higher than the slope

of the preceding ones, that is $c^i_{j+1} \geq c^i_j \; \forall j, i$. This means that, when the cost function is minimized, x^i_{j+1} cannot be different from zero if x^i_j is not at its upper bound, since the solver prioritizes the variables with lower cost coefficients. Hence, minimization automatically enforces the satisfaction of (73), which need not be explicitly accounted for.
Now that the modeling assumptions have been clarified, by letting $c^i = [c^i_1, \ldots, c^i_{m_i}]^T$ and $\mathbb{1}_{m_i} = [1, \ldots, 1] \in \mathbb{R}^{1 \times m_i}$ the reformulated minimization problem can be stated:

$$\begin{aligned} \min_{x^i} \quad & \textstyle\sum_{i=1}^N (c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^N \mathbb{1}^T_{m_i} x^i = Load \\ & 0 \leq x^i_j \leq \overline{x}^i_j \qquad i = 1, ..., N \quad j = 1, ..., m_i \end{aligned} \tag{74}$$

The problem in (74) is exactly of the type of $\mathcal{P}_1$ introduced in Chapter 3. In the current setup, $\mathbb{1}_{m_i}$ corresponds to A^i in $\mathcal{P}_1$ while the $\overline{x}^i_j$ match the upper bounds for the decision variables d^i in $\mathcal{P}_1$.
Lastly, it is worth noting that if a generator has no upper bound $P^{i,max}$, the problem (74) is always feasible since its last segment x^i_j has no upper bound $\overline{x}^i_j$ either.

5.3 Solution stability evaluation

An energy dispatch company has successfully planned the optimal usage of a pool of known power plants in order to meet the required load. However the company is certainly interested in reducing the costs by searching for additional generators able to supply the desired area that may have smaller costs or better capacity. Finding new facilities though is a costly and time consuming operation so it would be ideal for the company knowing in advance how beneficial this poll for new generators could be. The *Wait and judge* scenario approach theory introduced in Chapter 3 will be used in order to retrieve the probability that this polling operation could bring a change in the original solution: a high probability of improving the solution will act as an incentive to start a new search for better facilities.

Consider problem (74) where initially only N agents are taken into account:

$$\begin{array}{ll} \min\limits_{x^i} & \sum_{i=1}^{N}(c^i)^T x^i \\ \text{s.t.} & \sum_{i=1}^{N} \mathbb{1}_{m_i}^T x^i = Load \\ & \quad 0 \leq x_j^i \leq \overline{x}_j^i \qquad i=1,...,N \quad j=1,...,m_i, \end{array} \tag{75}$$

and denote by x^* the solution to (75) as usual.
The underlying assumption is that these N producers can be seen as independently extracted (chosen) from a probability distribution that represents the entire population of producers. Polling can be seen as an independent extraction of new producers ferom the same population.
When a new agent y is added, the problem becomes:

$$\begin{array}{ll} \min\limits_{x^i, y_j} & \sum_{i=1}^{N}(c^i)^T x^i + \sum_{j=1}^{n_y} c_j^y y_j \\ \text{s.t.} & \sum_{i=1}^{N} \mathbb{1}_{m_i}^T x^i + \sum_{j=1}^{n_y} \mathbb{1}_{m_y}^T y_j = Load \\ & \quad 0 \leq x_j^i \leq \overline{x}_j^i \qquad i=1,...,N \quad j=1,...,m_i \\ & \quad 0 \leq y_j \leq \overline{y}_j \qquad j=1,...,m_y \end{array} \tag{76}$$

where m_y is the number of segments that represent the new agent y, and $\overline{y}_j$ the local upper bounds for such segments. By calling $x_+^\star$ the solution of (76), the stability index of x^*, $V(x^*)$ is defined as:

$$V(x^*) = \mathbb{P}\big\{\delta^y = (m_y, c^y, A^y, \overline{y}) \in \Delta : \; x_+^\star \neq (x^*, \mathbb{0}_{m_y})\big\}, \tag{77}$$

where $\overline{y}$ denotes the set of the upper bounds $\overline{y}_j$ $\forall j = 1, \ldots, m_y$ for the segments of y and $\mathbb{0}_{m_y} \in \mathbb{R}^{1 \times m_y}$ indicates that the new decision vector $y \in \mathbb{R}^{1 \times m_y}$ is composed only by zeros. (77) defines the probability that the newly found producer does not improve the initial solution found on N agents.
Recalling the main result of Chapter 3, the stability index of x^* should also be bounded

as follows:

$$\mathbb{P}^N\big\{(\delta^1,\dots,\delta^N)\in\Delta^N : V(x^*){<}\epsilon(s^*)\big\}\geq 1-\beta, \tag{78}$$

where s^* denotes the number of support agents of x^* and $\beta\in(0,1)$, $\epsilon\in(0,1)$ and the collection $(\delta^1,\dots,\delta^N)$ refers to the initial N agents.
In particular, according to Definition 3.1, an agent x^i in problem (75) is a support agent if at least one of its components x^i_j is different from zero, meaning that the generator i is utilized to produce some energy in the solution of (75). Intuitively, and as it will be seen in the numerical simulations, the higher s^* is, the more generators are used to retrieve the initial solution x^* in the original pool, meaning that some not so optimal ones may be chosen in order to fulfill the load, increasing the probability that the new agent y is able to improve the solution.

5.4 Simulation setup and numerical results

The aim of the following simulations is to show that the estimated probability of x^* changing upon the arrival of y is less than or equal than the theoretical bound provided by the scenario approach theory which has been proven to be suitable for this kind of problem in Chapter 3, thus validating the theoretical results.
Specifically, after solving the original problem (75), the number of *support agents* of the solution was evaluated. $M = 50N$ new agents were then extracted and tested one by one and the empirical probability of improving the solution was retrieved simply dividing the number of times it happened by M.
This procedure was repeated for 100 times obtaining a vector of 100 probabilities of improving the solution, each one associated with one of the initial 100 solutions and therefore with its number of *support agents*. These probabilities were then plot on a graph against the number of *support agents* to verify they were below the theoretical bound $\varepsilon(k)$ with the expected confidence.
The main parameters used for the simulations are:

- N number of initial agents or generators (extractions);

- M number of subsequent extractions for empirical probability of violation evaluation, fixed to $50N$;

- P_{min} (always set to 0) and P_{max} which are the values for the minimum lower bound and maximum upper bound for the decision variables P^i. The actual bounds $P^{i,max}$ are extracted from a uniform distribution ranging between those two values;

- L the right hand value for the coupling constraint ($Load$);

- β confidence coefficient for computing the theoretical bound $\epsilon(k)$, always fixed to 10^{-7};

- *maxslope* which denotes the maximum slope (or maximum cost) that can be used for the linear approximation. The slopes of the various segments are extracted from a uniform distribution ranging from 0 to *maxslope*.

To construct the approximated parabolas, each generator is assigned a $P^{i,max}$ as described before and then a random number, between 3 and 10, of intermediate points and associated cost coefficients is extracted from a uniform distribution. Then, the intermediate points are assigned a random value from a uniform distribution between 0 and $P^{i,max}$ and the cost coefficients one from 0 to *maxslope*. This new points will be used to build the upper bounds for the new decision variables used for the linear approximation (see Figure 2). The following results, grouped by N, will show the sensitivity of the outcomes to parameters N, L, and P_{max}, while *maxslope* is fixed for all simulations to value 5.
In all cases, as it was expected, all estimated probabilities of violation remain under the theoretical bound. Since $\beta = 10^{-7}$ only one point (representing 1 iteration of the algorithm) out of 10 million should be above the red curve.

5.4.1 Results for $N = 100$

Following the procedure described in the previous section, the initial problem with $N = 100$ was solved retrieving x^* and evaluating s^*. $50N$ new agents were then extracted and added one by one to see if they could improve the solution. The number of times it changed was then divided by $50N$, obtaining an empirical estimate of the probability of solution change which was plot against s^*, obtaining a point on the graph (Figure 6). Repeating the procedure 100 times yielded a cloud of points. By reiterating the whole routine for different values of P_{max}, 3 different clouds of points were plotted together to better visualize the impact the maximum power limit has on the problem. If the upper bounds P_{max} are set to be larger, the probability of changing the solution upon a new arrival will be lower, since the solver is able to use larger values for decision variables with an advantageous cost coefficient from the start. This translates to a real life situation in which the power plants have high capacities and the planner can just select a small number of them (the most efficient ones of course) to meet demand. As expected, all the points in the three different simulations remain below the theoretical upper bound $\epsilon(k)$, function of the number of support agents.

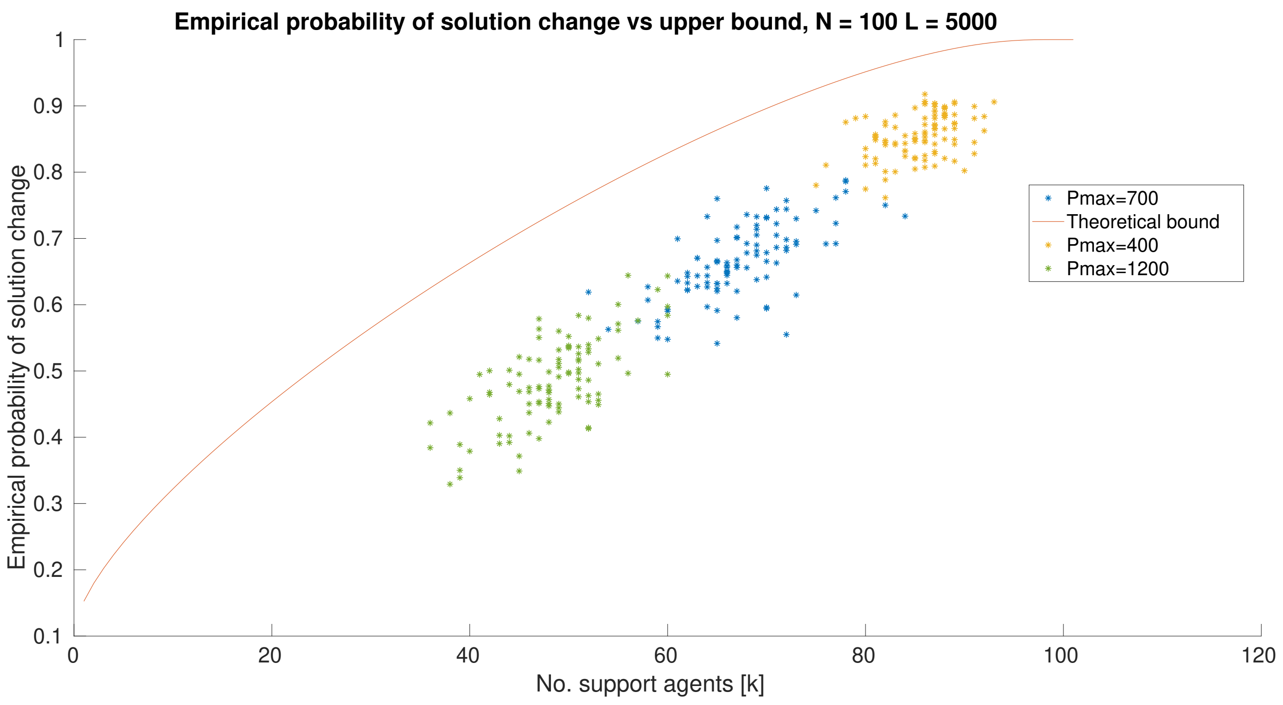

Figure 6: Increasing P_{max} results in higher probability of solution change alongside more support agents

5.4.2 Results for $N = 200$

The same routine, repeated again for different values of P_{max}, was run for $N = 200$. The results in Figure 7 show, as expected, all probabilities below the theoretical bound and also, with respect to the previous case, that the probabilities of solution change are roughly halved per same P_{max}. In fact, increasing N increases the pool of agents with which the initial solution is found, lowering the probability of solution change per same P_{max} when compared to a case with a lower N. This means that the planner is able to choose from more plants right away, so having observed a larger sample the probability of discovering a more cost effective facility lowers.

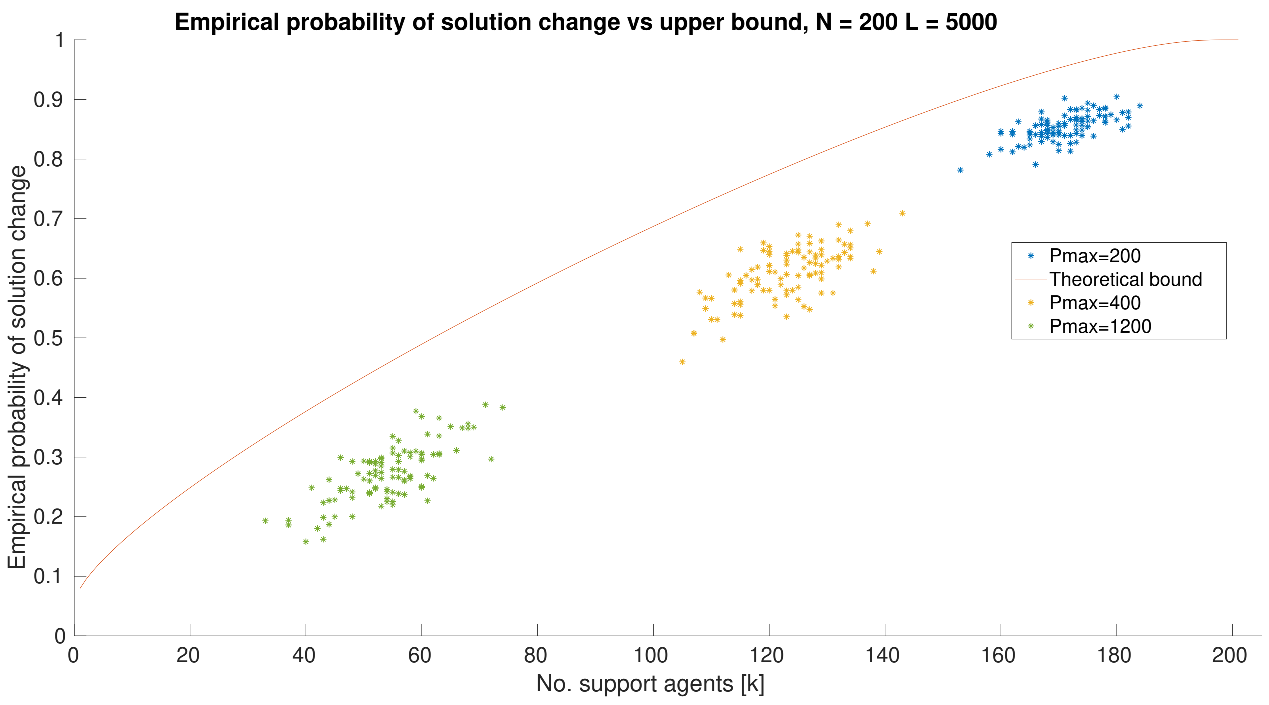

Figure 7: Doubling the number of agents similar results are obtained by roughly halving P_{max}

5.4.3 Results for $N = 100$ and L=10000

In Figure 8, results are shown for various simulations in which $N = 100$ but L was doubled. The same procedure was adopted, obtaining three different clouds of points on the graph, all staying under the theoretical bound. A larger L means more agents are required to fulfill the goal (since this is a equality constrained problem), thus resulting in higher probabilities of violation per same P_{max} when compared to cases with lower L. This case depicts a situation in which the planner is able to fulfill the requested load using a big portion of the initial pool of plants, including not so efficient ones so discovering new facilities has an higher probability of improving the solution.

It can be observed that if P_{max} is set to larger values, the points tend to spread more along the x axis. Since P_{min} is fixed to 0, a bigger P_{max} increases the variance of the uniform distribution the upper bounds for the power plants are taken from, thus yielding more diverse results in which the number of agents that contribute to the solution is more varied.

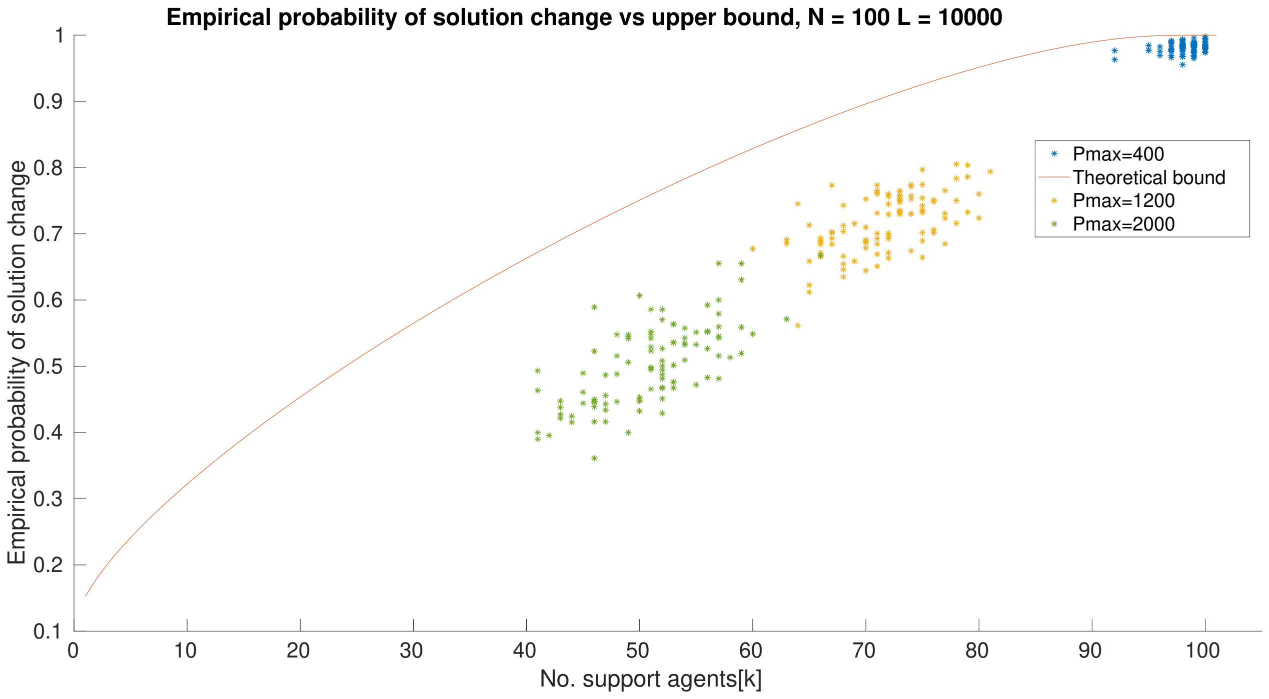

Figure 8: This time only L was doubled, obtaining similar results as with increasing N alone

5.4.4 Results for $N = 100$, Normal distribution

A fundamental result of the scenario theory developed in Chapter 3 is that it holds irrespectively of the probability distribution the agents are extracted from. This property is fundamental as the application of the result does not require any knowledge of this distribution. In the present context, this means that the energy dispatch company does not need to possess any information about the producers population (which is indeed unlikely) and can simply apply the result based on the *a posteriori* information given by the number of support agents, which is readily available.

To test this important property, simulations where the $P^{i,max}$ are sampled from a truncated normal distribution ranging from 0 to P_{max} and the intermediate points from the same kind of distribution between 0 and $P^{i,max}$ for each generator i are also included and shown in Figure 9.

Figure 9 show the results for the same procedure used for the previous simulations but with the differences introduced above. As it was to be expected, all the obtained probabilities of solution change stay below the theoretical bound proving that indeed the *Wait and judge* scenario approach theory holds regardless of the probability distribution the uncertain constraints/agents are sampled from.

Moreover, similarly to what was seen in Figures 4,5,6 increasing the variance of the normal distribution leads to a larger spread of the clouds of points on the plot.

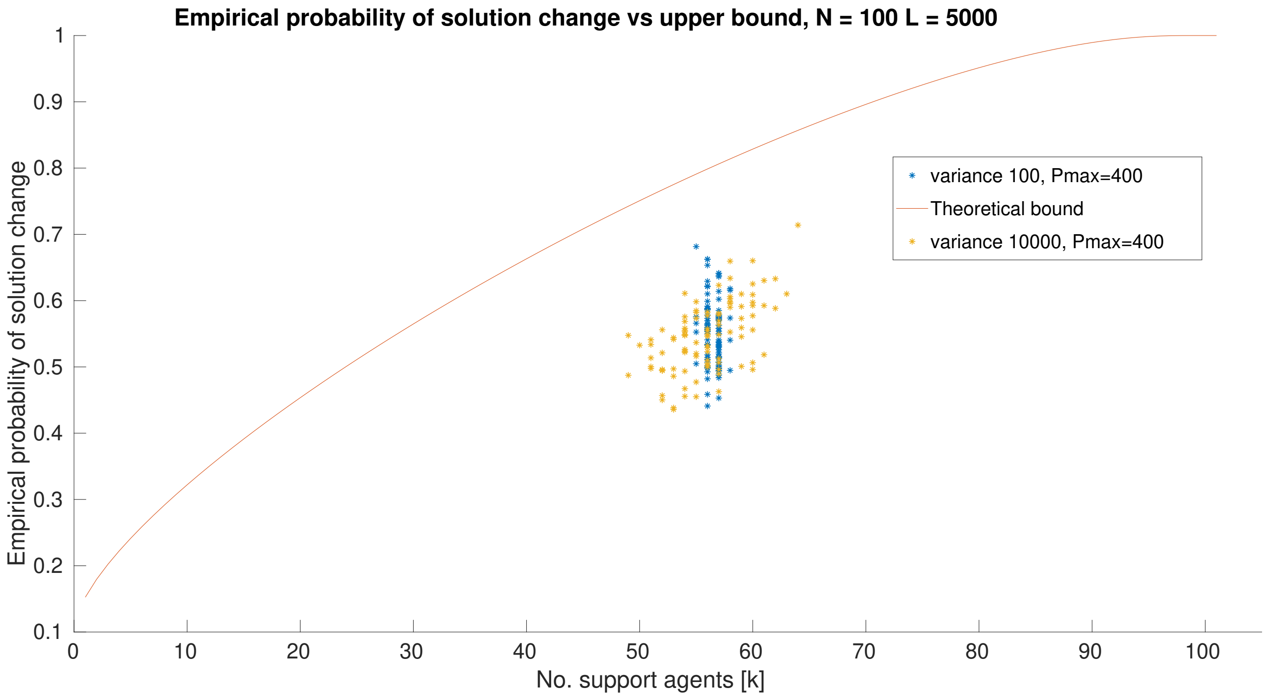

Figure 9: Results using a truncated normal distribution for the agents. P_{max} was fixed to 400 and only the variance of the distribution was varied, making the points spread wider.

6 Application to Cargo aircraft loading

Air freight companies offer their airplanes to traders who need to transport their goods. Airborne transportation is typically much more expensive than other methods but also faster and so it is the priority choice when delivering urgent or perishable items. Shipment dimension varies by a great factor and practically anything can be transported by plane. For this reason, the cargo is usually arranged inside special metal containers called ULDs, unit load devices, which have different shape in order to use efficiently the available space in the fuselage. Each container has a limited capacity that should be exploited in the best way possible, while also avoiding packing items that pose a risk when too close to each other and providing an appropriate padding to prevent the merch from being damaged, effectively creating a sub-problem within the aircraft load planning one.

The air cargo loading problem is solved generally 24 hours before departure and it begins usually when the cargo space is fully booked or almost. First ULDs of various form factors are chosen, then the booked shipment is sorted and arranged inside them, finally the ULDs are loaded into the aircraft.

It may happen that one or more ULDs are damaged or arrive late, in that case it is replaced by another one, loaded with different shipment, in order to still be able to fill the airplane. Sometimes the planned shipment is packed more efficiently than expected in the containers, leaving room for other ULDs that would have shipped later onto another flight, increasing the total value of the cargo.

The placement of the ULDs in the aircraft is not an easy task: beside the obvious weight and volume capacities of the airplane, they must be arranged in such a way to guarantee optimal balance of the aircraft throughout the flight and also to lower as much as possible its center of mass to reduce fuel consumption. These last two aspects will not be considered in this case study, whose aim is not to accurately model a real world situation as in [11], far too complex and including matters not interesting to this research which will consider a simpler model [12] with further relaxations. Moreover

only one aircraft performing one stopless flight will be considered, neglecting other problematics such as refueling and partial unloading and reloading of the vehicle.
The problem will consider a situation where the maximum profit from carrying different goods is to be attained while filling the cargo airplane as much as possible. The planning procedure is initially solved but during the actual loading phase as it was discussed before, some room can be left for an additional ULD, some container can be repacked to carry more items or even swapped for a more rewarding one thus improving the initially thought solution. These operations obviously take time and it may not be possible to replan or rearrange the ULDs in the aircraft in order to obtain a better solution. The scenario approach theory will be exploited to evaluate the probability of improving the original solution if the operators wait to see if more urgent or rewarding (from an economical point of view) shipment comes by.

6.1 Linear programming modeling

The decision variables x^i for this problem are the quantities in kg of every different good to be carried, which has a value coefficient c^i that represents how much the freight company is paid for carrying a certain quantity of the specified ware. Specifically, more urgent shipments may be paid more in order to arrive on time.
Each x^i has a lower bound set to 0 (good not shipped) and an upper bound set by its estimated demand (by the customers of the transportation company) in order to avoid shipping excessive quantities of a merch that would remain unsold.
Finally, the employed cargo aircraft has maximum weight (W) and volume (V) capacities that set limits not only on the sum of all the x^i but also on the average density of the shipment. Most of the times the two constraints are not both active, so an equivalent problem with only equality coupling constraint would be usually not feasible.

In conclusion, the linear problem reads:

$$\begin{aligned}
\max_{x^i} \quad & \textstyle\sum_{i=1}^N (c^i)^T x^i \\
\text{s.t.} \quad & \textstyle\sum_{i=1}^N x^i \leq W \\
& \textstyle\sum_{i=1}^N a^i x^i \leq V \\
& 0 \leq x^i \leq d^i \qquad i = 1, ..., N,
\end{aligned} \tag{79}$$

where a^i is the density of x^i and d^i its estimated demand. The problem in (79) is exactly of the type of $\mathcal{P}_2$ introduced in Chapter 4. In the current setup, $[1, a^i]^T$ corresponds to A^i in $\mathcal{P}_2$ while b is matched by $[W, V]^T$.

6.2 Solution stability evaluation

An air freight company has received an initial batch of requests from customers and has planned the optimal arrangement of these initial items on an aircraft. However after the airplane has been loaded some late items from other customers arrive. Often times there is still room left in the aircraft after the loading procedure or it can be created anyway by rearranging items inside the ULDs or even discarding some, and shipping them on another plane departing later, if the late ones are more profitable. In all of these cases, waiting for the new items to arrive and reloading the aircraft takes additional time that can surely cause a delay and requires extra work from the workers that ultimately result in more costs for the company but it may be worth waiting for more items if they guarantee an increased profit.

The *Wait and judge* scenario approach theory provides a means of knowing beforehand if waiting for new goods can make up for the costs of rearranging the items already in the aircraft by giving an upper bound for the probability of changing the initial solution. In particular, if such upper bound is high, the company will be eager to wait for new items to come since they have an high probability of improving the original arrangement.

Consider problem (79) where initially only N goods are taken into account:

$$\begin{aligned} \max_{x^i} \quad & \textstyle\sum_{i=1}^{N}(c^i)^T x^i \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{N} x^i \leq W \\ & \textstyle\sum_{i=1}^{N} a^i x^i \leq V \\ & 0 \leq x^i \leq d^i \qquad i = 1, ..., N \end{aligned} \tag{80}$$

Let us call ts solution x^*. Once again these N are assumed to be independently observed from a probability distribution that represents the entire variety of goods. Waiting for new items to arrive can be seen as extracting new goods independently from such distribution.

When a new agent y is added, the problem becomes:

$$\begin{aligned} \max_{x^i, y} \quad & \textstyle\sum_{i=1}^{N}(c^i)^T x^i + y \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{N} x^i + y \leq W \\ & \textstyle\sum_{i=1}^{N} a^i x^i + a^y y \leq V \\ & 0 \leq x^i \leq d^i \qquad i = 1, ..., N \\ & 0 \leq y \leq d^y, \end{aligned} \tag{81}$$

By calling $x^\star_+$ the optimal solution of (81), the stability index of x^*, $V(x^*)$ is defined as:

$$V(x^*) = \mathbb{P}\big\{\delta^y = (c^y, A^y, d^y) \in \Delta : \ x^\star_+ \neq (x^*, 0)\big\}, \tag{82}$$

(82) defines the probability that the newly arrived item does not improve the initial solution found on N agents.

Recalling the main result of Chapter 4, the stability index of x^* should also be bounded as follows:

$$\mathbb{P}^N\big\{(\delta^1, \dots, \delta^N) \in \Delta^N : V(x^*) {<} \epsilon(s^*)\big\} \geq 1 - \beta, \tag{83}$$

where s^* denotes the number of support agents of x^* and $\beta \in (0,1)$, $\epsilon \in (0,1)$ and the collection $(\delta^1, \dots, \delta^N)$ refers to the initial N agents.

An high $V(x^*)$ will act as an incentive to wait for more items to arrive since it means it is more likely to obtain an improvement. As said before in Chapter5, an high s^* means high probability to improve the original solution x^* corresponding to a situation where the airplane is filled with a good portion of the initial N goods, even suboptimal ones in order to attain full capacity.

6.3 Simulation setup and numerical results

Following the same reasoning as in section 5.3, the *Wait-and-judge* scenario approach theory will be used in order to assess the probability of x^* changing upon the arrival of y, validating the results of chapter 4. In a similar fashion, after solving (79), $M = 50N$ new goods were considered and tested one by one, retrieving an empirical estimate of the probability of solution change by dividing the number of times the solution changed by M.

This procedure was repeated for 100 times obtaining a vector of 100 probabilities, each one associated with one of the 100 initial solutions and their number of *support agents.* The probabilities were then plot on a graph to verify they were under the theoretical bound $\varepsilon(k)$, function of the number of *support agents* of the original solution which is equivalent to the number of agents with a value different from zero out of the N initial ones.

The simulation parameters are similar to the ones in 5.3 for the previous example with some more to be taken into account:

- N number of initial agents or items (extractions);
- M number of subsequent extractions for empirical probability of violation evaluation, fixed to $50N$;
- D_{min} and D_{max} which are the minimum demand and maximum demand for the goods. The actual bounds d^i are extracted from a uniform distribution ranging between those two values;

- W the aircraft maximum weight capacity;
- V the aircraft maximum volume capacity;
- β confidence coefficient for computing the theoretical bound $\epsilon(k)$, always fixed to 10^{-7};
- C_{min} and C_{max} values for the maximum and minimum cost coefficient. The actual c^i are extracted from a uniform distribution ranging between those two values.
- a_{min} and a_{max} values for the maximum and minimum density. The actual a^i are extracted from a uniform distribution ranging between those two values.

For all the simulations, W and V are fixed and equal to the weight and volume capacity of a Boeing 737 MAX 8 aircraft and the densities of the goods range between 900 (approximately the one of polyurethane plastic) and 7000 (close to that of iron).
The results will be grouped by N, varying D_{min} and D_{max} to show how the probability of violation and the number of agents contributing to the initial solution vary as well. Even though the coupling constraints are inequality ones, the results bear resemblance to the previous case presented in chapter 5 and it can be noticed again how no points are to be found above the theoretical threshold as it is to be expected when choosing confidence $\beta = 10^{-7}$.

6.4 Results for $N = 100$

In Figure 10, the procedure presented before is repeated four times, in a similar way to Chapter 5, for different values of D_{min} and D_{max}, obtaining four different cloud of points to visualize the impact the two parameters have on the problem. For high D_{max} the number of agents contributing to the original solution tends to decrease, this represent a situation where customers want to ship very large quantities of their merchandise and the cargo company is able to pick the best paying few of them to fill up the airplane. Vice-versa for low D_{max}, the air cargo company will have to rely on a broader variety of goods to exploit the full capacity of the aircraft. One interesting feature is the spread of the points across the graphs: the higher the difference between D_{min} and D_{max} the broader the distribution of the points.

Increasing the gap between minimum and maximum demand corresponds to increasing the variance of the uniform distribution from which the upper bounds are sampled, leading to more diverse outcomes and more different numbers of *support agents*. This means having more variety of goods with different prices to choose from, leading to much more diverse outcomes. As it was expected, all points lie below the theoretical bound $\epsilon(k)$.

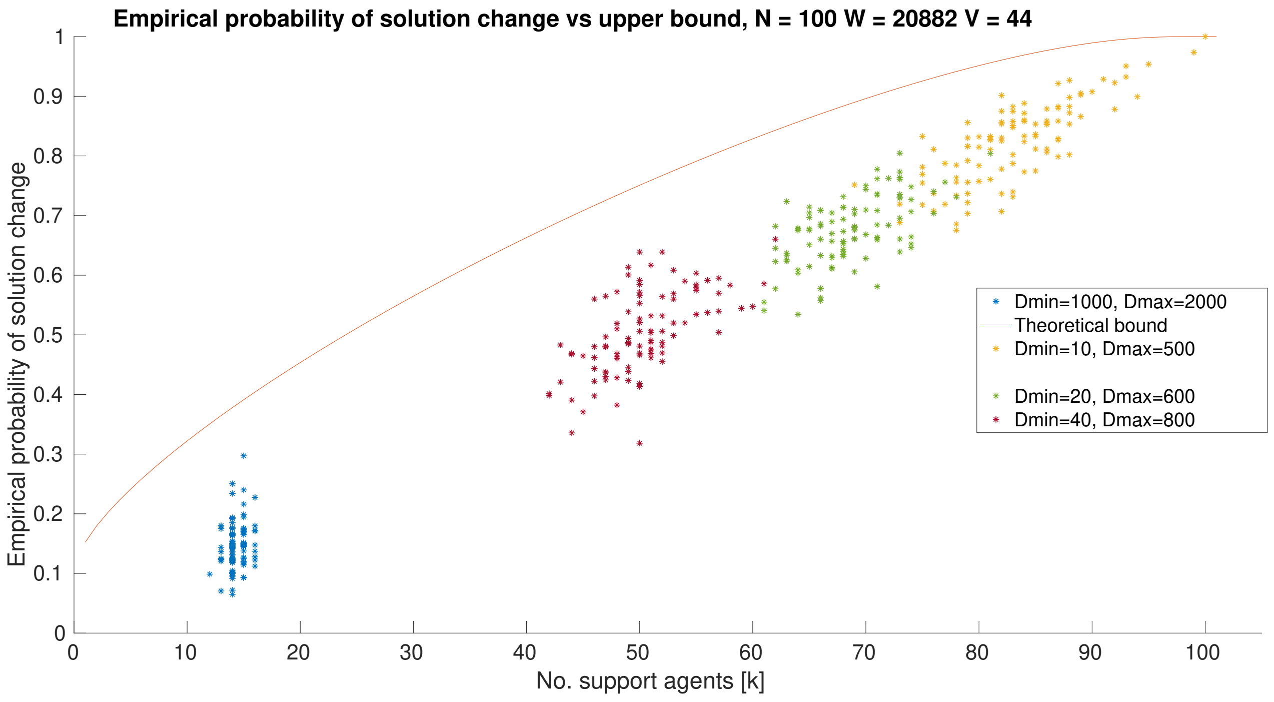

Figure 10: Increasing D_{min} and D_{max} results in higher probability of violations alongside more support constraints

6.5 Results for $N = 200$

In Figure 11 N was increased to 200, resulting in lower probability of solution change per same number of contributing agents, since observing a larger sample from the start leads to lower the probability of seeing a better good later on. Again the procedure was repeated different times, in this case more clouds of points were plotted showing that the number of agents that contribute to the original solution roughly stays the same as in Figure 8 (in the range 0-100).

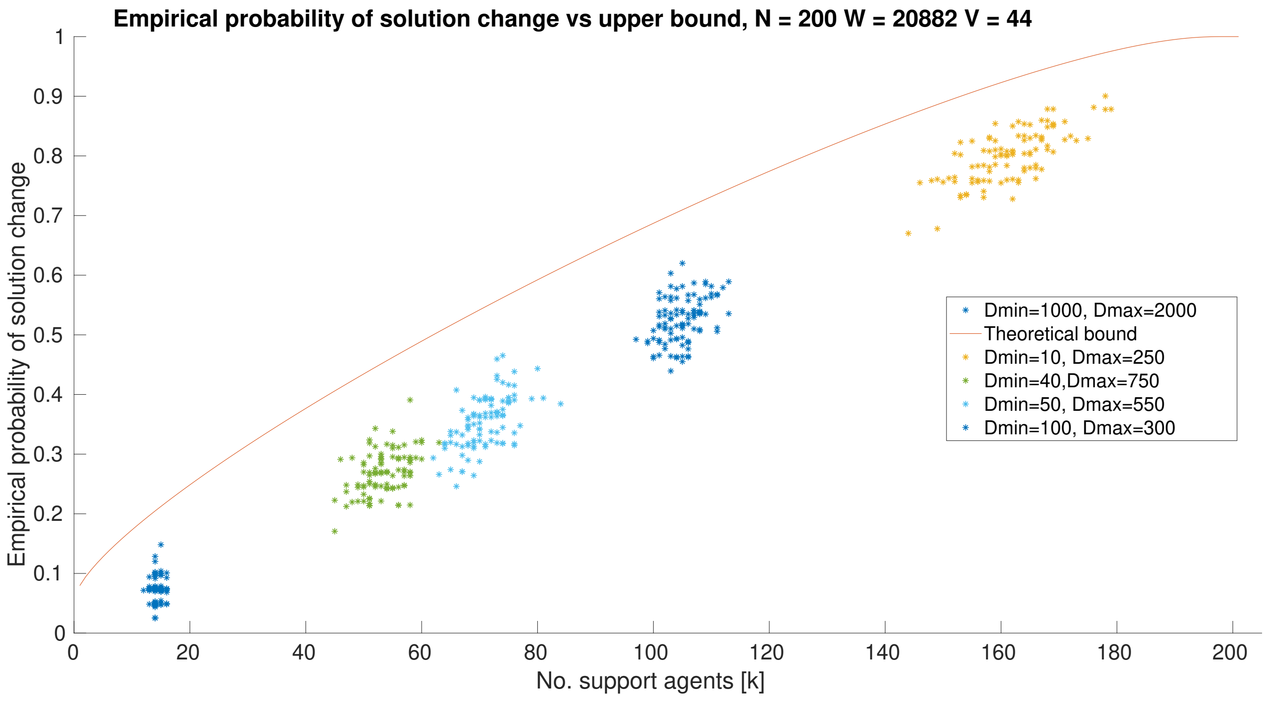

Figure 11

6.6 Results for $N = 100$, very small D_{max}

Figure 12 shows what happens when the upper bounds for the decision variables are too strict and no coupling constraint is active at the original solution.

This results in a probability of solution change of 1, since a newly arrived agent obtains a value different from zero but also the previous values for the original agents do not change. In other words the new solution called $x^{\star}_{+} = (x^{*}, y)$ where x^{*} represent the original solution with N agents and y a nonzero value for the newly arrived agent. This case may represent an error in the original planning phase in which the volume or weight capacities of the aircraft were misinterpreted or a situation where very few requests are received and the company may decide to delay the flight in order to use up all the space in the cargo aircraft.

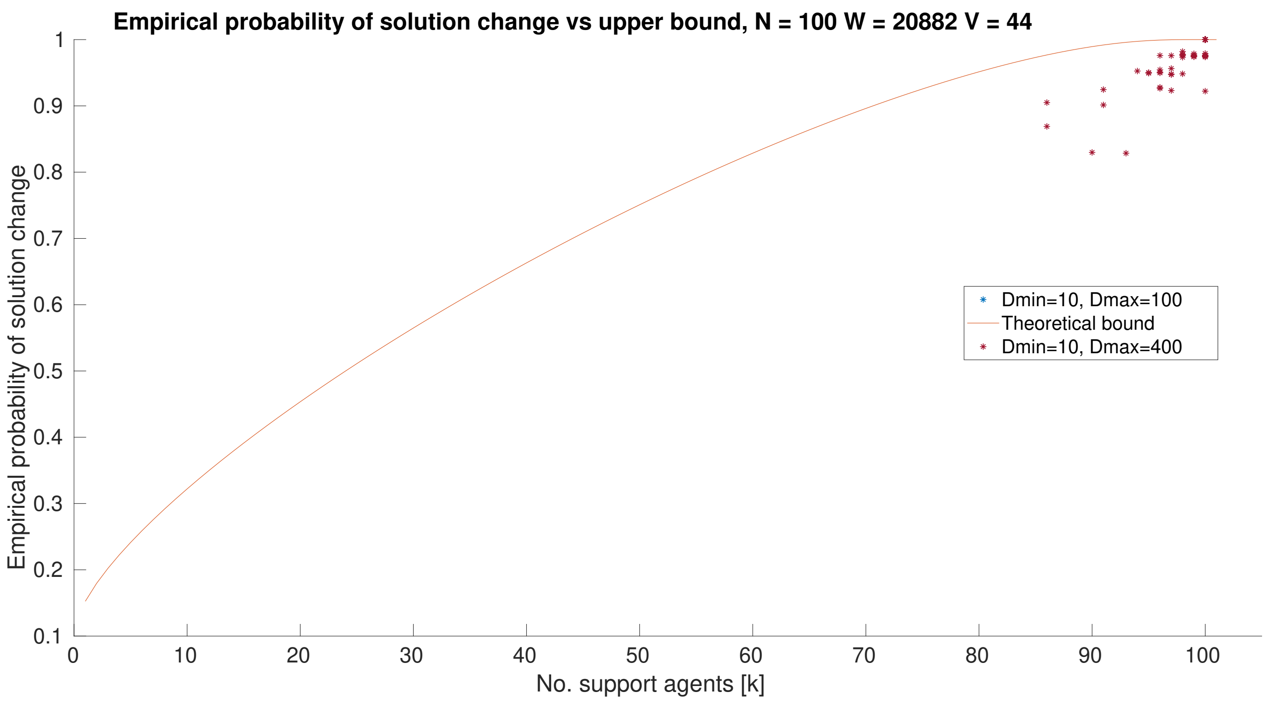

Figure 12: In this cases, D_{max} was too small to activate one of the two coupling constraints, so adding a new agent brings a change to the original solution almost surely with probability one. Results with $D_{max} = 100$ are covered by the ones with $D_{max} = 400$ because the points are all concentrated in (100,1).

6.7 Results for $N = 100$, Normal distribution

As in Chapter 5, a simulation was dedicated to demonstrate that the scenario theory introduced in Chapter 3 and extended for this type of problems in Chapter 4 holds irrespectively of the distribution the agents are observed from. The results in Figure 13 show that indeed the *Wait and judge* scenario approach theory is valid independently from the probability distribution the agents are sampled from. In a real life situation, this means that the air cargo company can simply apply the result derived from the number of support agents without having information on the population of the items. Similar considerations to section 6.3 can be made with respect to increasing or decreasing D_{max}, D_{min} and their difference.

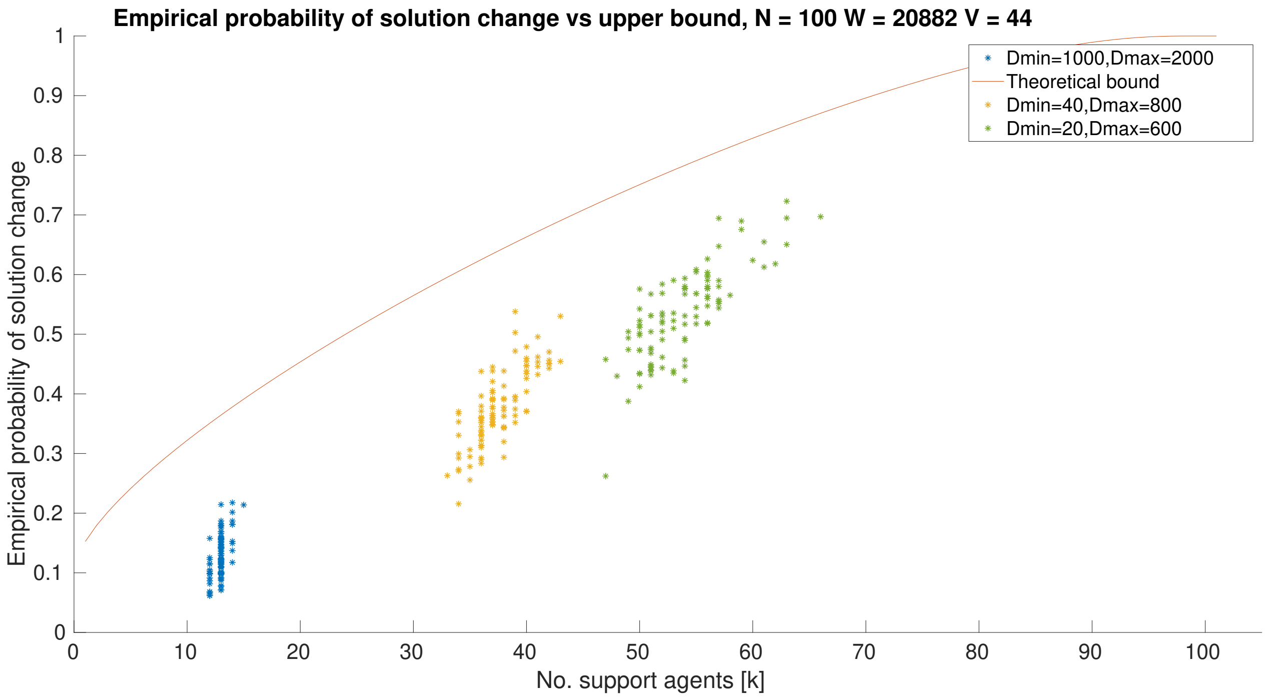

Figure 13: Here the d_i are extracted from a chopped normal distribution. The results are almost identical to the previous case in which a uniform distribution was used.

7 Conclusions

In this research work, methods for assessing the probability of improving the solution of resource allocation problems were presented. In particular, the type of problems treated in this thesis were linear programs where some kind of resource was to be shared among different users, or agents in order to minimize or maximize a cost that depended on coefficients associated to the agents themselves. These problems involve a finite number of agents observed from a population from which however more agents can be extracted. Observing a new agent different from the initial ones and adding it to the problem may bring an improvement in the value of the cost function, or it may not. Moreover, the act of searching for a new agent is usually a demanding and time consuming operation and for this reason assessing the probability of improving the initial solution, *called stability index*, upon finding a new agent is important. To this purpose, results from both the classic and the wait-and-judge scenario theory were employed in order to give suitable upper bounds for the *stability index* of a solution.
At first, in Chapter 2, linear problems in standard form where the whole resource must be allocated and there are no limits to the quantity of that resource assigned to each agent are considered. For this type of problems, a characterization of the *stability index* was already given [5] through the classic scenario approach theory providing an *a priori* upper bound to such probability of solution change based only on the number of initial agents and a user chosen parameter.
Then, in Chapter 3, the main contribution of the thesis is presented. After acknowledging that the previous results provided by the classic scenario approach theory are not valid for resource sharing problems with upper limits on the quantity of resource allocated to each agent, a characterization of the *stability index* in the new context is provided using the wait-and-judge scenario approach theory. This more recent development is employed in order to obtain an *a posteriori* upper bound to the *stability index*, meaning that the initial problem has to be solved in order to observe a quantity that is used to compute the bound to the probability of solution change.

In Chapter 4, problems with upper limits on each agent and inequality constraints on the shared resource, instead of equality constraints, were treated. Also in this case, the results of the classic scenario theory were proved to be not suitable and a characterization of the stability index was provided by resorting to the wait-and judge variant. Moreover it was proved that this kind of problems are fundamentally equivalent to the one proposed in Chapter 3, obtaining similar results.
The theoretical contributions of Chapters 3 and 4 were then validated in Chapters 5 and 6 with two numerical examples. The example in Chapter 5 was about the optimal allocation of a requested load among different power producers and focused on problems with upper limits on the agent decision vectors as in the framework of Chapter 3. The compliance of the *stability index* with the bound provided in Chapter 3 was empirically verified by solving a problem with randomly generated agents. In Chapter 6 an aircraft loading problem was introduced, with inequality constraints on the available resources as in Chapter 4 and a similar procedure for evaluating the correctness of the theoretical bound was adopted as in Chapter 5. The results of both simulations were satisfying and in line with what was predicted in the theory chapters.

7.1 Future work

An interesting direction future works on the subject could take would be providing a characterization of the stability index for the optimal solutions of nonlinear programming problems (NLPs). The differences between the linear and nonlinear duality theories mean that many of the properties used in this research may not be valid for nonlinear problems. In particular it is difficult to define appropriate observable quantities (as s^*) for the characterization of the stability index, since how much the solution of a nonlinear problem can improve depends on the specific problem at hand.

A Scenario approach

Let us consider a generic problem [2]:

$$\begin{aligned} \min_{x \in \chi \subseteq \mathbb{R}^d} \quad & c^T x \\ \text{s.t.} \quad & x \in \chi_\delta, \quad \delta \in \Delta \end{aligned} \tag{84}$$

Where d denotes the number of decision variables, χ, $\chi_{\delta(i)}$ represent a convex and closed sets and δ is an uncertain parameter. Problem (84) is affected by uncertainty on constraints and usually Δ has infinite cardinality.

The scenario approach theory allows to deal with the uncertainty in (84) guaranteeing that the solution of the problem has an high probability of being feasible fro all the constraints in Δ while keeping the computational cost low unlike other more conservative techniques.

Consider now problem (84) in which only N randomly and independently extracted constraints are enforced:

$$\begin{aligned} \min_{x \in \chi \subseteq \mathbb{R}^d} \quad & c^T x \\ \text{s.t.} \quad & x \in \bigcap_{i=1,\dots,N} \chi_{\delta(i)} \end{aligned} \tag{85}$$

Assumption A.1. Problem (85) is always feasible regardless of the specific constraints extracted $\delta^{(1)}, \dots, \delta^{(N)}$ and its minimizer is unique.

Problem (85), also called scenario program, has a finite number of constraints (N) and because of this it can be solved with a lower computational effort than (84).

The solution to (85) however cannot satisfy all of the constraints in Δ but hopefully a large portion of them. The size of this portion is assessed in a probabilistic way, introducing the definition of probability of *violation*:

Definition A.1. - Violation probability [2] The violation probability of a given $x \in \chi$ is defined as $V(x) = \mathbb{P}\{\delta \in \Delta : x \notin \chi_\delta\}$.

By denoting with x_N^* the solution of (85), to make sure that x_N^* satisfies a large portion of Δ, $V(x_N^*)$ should be under a desired threshold ε but since $V(x_N^*)$ is a random variable itself defined over a set of N independent extractions $\delta^{(1)}, \dots, \delta^{(N)}$ the overall product probability $\mathbb{P}^N\{V(x_N^*) > \varepsilon\}$ should be bounded. In particular [2] states that:

$$\mathbb{P}^N\{V(x_N^*) > \varepsilon\} \leq \sum_{i=0}^{d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \tag{86}$$

(86) holds for all classes of convex problems but it is tight (holds with =) for a special class of problems called *fully supported problems*, that will be defined below.

Definition A.2. -Support constraints [2] A constraint of a scenario program is a support constraint if its removal changes the solution. Moreover, the number of support constraints is always less than or equal than d, the size of x.

Definition A.3. -Fully supported problem [2] A scenario problem is a *fully supported problem* if the number of support constraints is equal exactly to d.

Result (86) is presented by theorem 2.4 of [2], along with another statement valid only for *fully supported problems*:

$$\mathbb{P}^N\{V(x_N^*) > \varepsilon\} = \sum_{i=0}^{d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \tag{87}$$

Typically, coefficients $\varepsilon \in (0,1)$ and $\beta \in (0,1)$ (confidence factor), are chosen in order to get the required number of extractions N needed to ensure $\mathbb{P}^N\{V(x_N^*) > \varepsilon\} \leq \beta$. These coefficients must satisfy :

$$\sum_{i=0}^{d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} = \beta. \tag{88}$$

A.0.1 Scenario approach with constraint removal

After extracting N constraints, it may be desirable ot remove some of them in order to improve the value of the cost function. There exist different ways to remove constraints that can be either optimal or heuristic and while the value of the cost function may differ depending on which strategy is adopted, the bound on the probability of violation that will be provided holds regardlessly.
By denoting with θ_k^* any solution to (85) that violates k constraints, the following result from theorem 2.1 of [13] guarantees that:

$$\mathbb{P}^N\{V(\theta_k^*) > \varepsilon\} \leq \binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i}. \tag{89}$$

It should be noted that removing k constraints is not the same thing as violating k constraints, since at a later stage in the removal procedure the violated constraints may be satisfied once again. In order to apply the result (89), the constraints that turn out to be satisfied later on will be reintroduced into the problem and other ones will be removed to have exactly k violated constraints in the end.
Result (89) can be used to compute the largest k, after fixing β, ε and N, such that the right hand side of (89) is smaller than β. This allows to select the largest number of constraints that can be removed while respecting the bounds ε and β on the violation probability.

A.1 Wait-and-judge scenario approach

The wait-and-judge scenario theory enables for an *a posteriori* evaluation of the probability of the solution of a random program $\mathcal{P}_m$ changing upon the arrival of a new agent.
The notion of support constraint is used in the wait-and-judge scenario approach to provide a stricter bound for the probability of violation than the standard scenario approach theory, which is useful in the case of problems where the number of support

constraints is less than the number of decision variables (*non fully supported problems*). The result provided is considered a posteriori because in order to compute the violation bound the solution to (85) has to be computed first in order to evaluate the number of support constraints, while the standard scenario approach gives a result valid regardless of such information, which is more conservative.
Under the assumptions [4] that the problem admits a unique minimizer (if necessary, recurring to a tie-break rule) and that this minimizer remains unchanged upon removal of constraints that are not support constraints, the new bound $\varepsilon(k)$ can be introduced as resulting from [4].
Fixing $\beta \in (0,1)$ (confidence), for any $k = 0, 1, \ldots, d$, there exists one and only solution $t(k) \in (0,1)$ to the equation:

$$\frac{\beta}{N+1} \sum_{m=k}^{N} \binom{m}{k} t^{m-k} - \binom{N}{k} t^{N-k} = 0 \tag{90}$$

where N is the number of initial extractions of constraints. By letting $\varepsilon(k) = 1 - t(k)$, under the previous two assumptions it holds that:

$$\mathbb{P}^N\{V(x_N^*) \, > \, \varepsilon(s_N^*)\} \leq \beta \tag{91}$$

where x_N^* denotes the solution of the initial problem with N extractions, s_N^* the number of support constraints that characterize x_N^*, $V(x_N^*)$ the probability of violation of solution x_N^* and $\mathbb{P}^N$ is a probability product assuming the N sampled constraints are i.i.d..
$\varepsilon(k)$ can be extrapolated from (90) through a bisection algorithm.
Similarly to section (A.0.1) a situation where one is allowed to violate constraints in order to improve the cost value can be evaluated as in [8]. Consider a variant of problem

(84):

$$\begin{aligned} \min_{\substack{x\in\mathcal{X},\\ \xi_i\geq 0, i=1,\dots,N}} \quad & c^T x + \rho \textstyle\sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & f(x,\delta_i) \leq \xi_i, \quad i=1,\dots,N, \end{aligned} \tag{92}$$

where $\delta_i, i = 1, \dots, N$ is an independent random sample from $(\Delta, \mathcal{F}, \mathbb{P})$, $f(x,\delta_i)$ is a convex function for any given δ (referring to the previous notation for (84)) and $\mathcal{X}_\delta = \{x : f(x,\delta) \leq 0\}$. The function f is used to express the regret for violating a constraint: for a given δ, the regret at x is $f(x,\delta)$. If $\xi_i > 0$, the constraint $f(x.\delta_i) \leq 0$ is relaxed to $f(x,\delta_i) \leq \xi_i$. The probability of violation for (92) can then be defined as

$$V(x_N^*) = \mathbb{P}\{\delta \in \Delta : f(x,\delta) > 0\}, \tag{93}$$

where x_N^* is the optimal solution of (92). This time the number of support constraints s_N^* associated with x_N^* will be the number of δ_i for which $f(x_N^*, \delta_i) \geq 0$ and similar conclusions as for (85) can be drawn [8].

With respect to the problems treated in this research, $\mathcal{P}_{0,1,2}$ and $\mathcal{D}_{0,1,2}$, the above definition (A.2) applies to $\mathcal{D}_{0,1,2}$ since the theory is defined on problems of that form. The number of constraints of $\mathcal{D}_{0,1,2}$ directly corresponds to the number of decision variables in $\mathcal{P}_{0,1,2}$, so the *support constraints* are linked to the decision variables of $\mathcal{P}_{0,1,2}$ that are different from zero in the solution. Removing an agent that contributed to the solution would indeed result in a change. Moreover in the two problems treated in chapters 5 and 6, $d = N$, since the agents (decision variables) for the initial problem are extracted N times.

A.2 MATLAB code

Here the MATLAB code for running the simulations shown in Chapter 5 and 6 is provided. Note that this code uses IBM ILOG CPLEX to solve the linear programming problems but it can be replaced by MATLAB *linprog* if needed without modifying the code syntax. Moreover the MATLAB Parallel Computing Toolbox is needed when calling *parfor*. By changing all instances of *parfor* to *for* the toolbox is no longer needed, at the cost of computational speed.

A.2.1 Economic dispatch problem

The function *scenario_main.m* is used to run the complete simulation, calling all the other auxiliary functions. Its inputs are:

- N, number of initial extractions;
- beta the confidence parameter;
- L the requested load;
- $Pmass$ the maximum power limit for the generators which is equivalent to $Pmax$ in Chapter 5;
- C_min and C_max the minimum and maximum values for the cost coefficients;
- $Npoints$ the number of times the empirical stability index has to be evaluated;
- $flag$ a parameter that when set to 1 switches the problem to the chopped normal distribution case.

```
function Pemp = scenario_main(N,beta,L,Pmass,Cmin,Cmax,Npoints
    ,flag)
tic

%% Generators problem

    d=N;
    Pmin=zeros(1,N);
    maxslope=5;
    sigma=zeros(1,Npoints);
    Pemp=zeros(1,Npoints);
    fprintf('Progress:\n');
    fprintf(['\n' repmat('.',1,Npoints) '\n\n']);

    parfor g=1:Npoints
        fprintf('\b|\n');
        Pmax=abs(round(normrnd(Pmass/2,100,[1,N])))+1;
        if flag==1
        end
        if flag ==0
        Pmax=randi(Pmass,1,N);
        end
        C=randi([Cmin,Cmax],1,N);
        Q=randi([3,10],1,N);

        [x,sigmaori,ub,f]= Dispatch_solve_improved(N,Q,L,Pmin,
            Pmax,C,maxslope,flag,Pmass);
        sigma(1,g)=sigmaori;

        Phat = Monte_Carlo(N,ub,x,f,L,maxslope,Pmass,Cmin,Cmax
            ,flag,Pmass);
        Pemp(1,g)=Phat;
    end

figure
hold on
plot(sigma,Pemp,'*')
hold on
```

```
plot(epsilon(N,d,beta))
title(['Empirical probability of solution change vs upper
    bound, N = ',num2str(N),' L = ',num2str(L)])
xlabel('No. support agents [k]')
ylabel('Empirical probability of solution change')
toc
end
```

Solve the problem - Dispatch_solve_improved.m

```
function  [x,sigmaori,ub,f]= Dispatch_solve_improved(N,Q,L,
    Pmin,Pmax,C,maxslope,flag,Pmass)

[S,Ub,Lmod] =setup2(N,Q,L,Pmin,Pmax,C,maxslope,flag,Pmass);

Ub=Ub';
ub=Ub(:);
ub=nonzeros(ub);
lb=zeros(size(ub));
Aeq=ones(size(ub))';
Beq=[Lmod];
Aineq=[];
Bineq=[];

S2=S';
S2=S2(:);
f=S2;
f=nonzeros(f);

[x,fval] = cplexlp(f,Aineq,Bineq,Aeq,Beq,lb,ub);
```

```
for a=1:N
      Ptot(1,a)=0;
      for b=1:Q(1,a)
      Ptot(1,a)=Ptot(1,a)+x(b,1);
      end
end

for t=1:length(Ptot)
            Ptot(1,t)=Ptot(1,t)+Pmin(1,t);
end

sigmaori=0;

o=zeros(1,100);
o(1,1)=1;

for t=2:length(Q)
      o(1,t)=o(1,t-1)+Q(1,t-1);
end

for t=1:length(o)
      if x(o(1,t),1) ~= 0
      sigmaori=sigmaori+1;
      end
end

end
```

Variables setup - setup2.m

```
function [S,Ub,Lmod] = setup2(N,Q,L,Pmin,Pmax,C,maxslope,flag,
    Pmass)

P=zeros(N,max(Q(1,:)));
S=zeros(N,max(Q(1,:)));

if flag==0
for a=1:N
      for b=1:Q(1,a)
            P(a,b)=(Pmax(1,a)-Pmin(1,a))*rand(1,1)+Pmin(1,a);
            S(a,b)=maxslope*rand(1,1);
      end
      P(a,:)=sort(P(a,:));
      S(a,:)=sort(S(a,:));
end
end

if flag == 1

for a=1:N
      for b=1:Q(1,a)
      px=Pmin(1,a):Pmax(1,a);
      mi=(Pmax(1,a)-Pmin(1,a))/2;
      dev=10;
      p=1./(dev*sqrt(2*pi))*exp((-(px-mi).^2)./(2*dev^2));
      P(a,b)=randpdf(p,px,[1,1]);
                 px=0:1;
            S(a,b)=maxslope*normrnd(1/2,10,[1,1]);
```

```
        end
        P(a,:)=sort(P(a,:));
        S(a,:)=sort(S(a,:));
        end

end

c       = size(P, 1);
[~, F] = sort(P == 0, 2);
P       = P((1:c).' + (F - 1) * c);

c       = size(S, 1);
[~, F] = sort(S == 0, 2);
S       = S((1:c).' + (F - 1) * c);

Ub=zeros(N,max(Q(1,:)));
for a=1:N
        Ub(a,1)=P(a,1)-Pmin(1,a);
        for b=2:(Q(1,a)-1)
                Ub(a,b)=P(a,b)-P(a,b-1);
        end
        Ub(a,Q(1,a))=Pmax(1,a)-P(a,Q(1,a)-1);
end

for g=1:length(Pmin)
L=L-Pmin(1,g);
end
Lmod=L;
```

```
end
```

Stability index empirical evaluation - Monte_Carlo.m

```
function [Phat]=Monte_Carlo(N,ub,x,f,L,maxslope,Pmax,Cmin,Cmax
    ,flag,Pmass)
M=50*N;

      Aineq=[];
      Bineq=[];
      Beq=[L];
      Pmaxnew=randi(Pmax,1,M);
      Cnew=randi([Cmin,Cmax],1,M);
      Qnew=randi([3,10],1,M);
      Pmin=zeros(1,M);

      [Snew,Ubnew,Lmod] = setup2(M,Qnew,L,Pmin,Pmaxnew,Cnew,
          maxslope,flag,Pmass);
      parfor j=1:M
           ubmod=[ub;nonzeros(Ubnew(j,:)')];
           fnew=[f;nonzeros(Snew(j,:))];
           lbnew=zeros(size(ubmod));
           Aeqnew=ones(size(ubmod))';
           xnew = cplexlp(fnew,Aineq,Bineq,Aeqnew,Beq,lbnew,ubmod
               );

           if not(isequal([zeros(Qnew(1,j),1)],xnew(end-(Qnew(1,j
               )-1):end,1)))
                 viol(1,j)=1;
      end
```

```
end
      nviol=sum(viol);
      Phat=nviol/M;

end
```

Theoretical bound - epsilon.m

```
function out = epsilon(d,N,bet)
out = zeros(d+1,1);
for k = 0:d
m = [k:1:N];
aux1 = sum(triu(log(ones(N-k+1,1)*m),1),2);
aux2 = sum(triu(log(ones(N-k+1,1)*(m-k)),1),2);
coeffs = aux2-aux1;
t1 = 0;
t2 = 1;
while t2-t1 > 1e-10
t = (t1+t2)/2;
val = 1 - bet/(N+1)*sum( exp(coeffs-(N-m')*log(t)) );
if val >= 0
t2 = t;

else
t1 = t;
end
end
out(k+1) = 1-t1;
end
```

Optional graphical prompt - gui3.m Running this function gives a graphical prompt to make running the simulation easier. It comes with preset values that can be modified.

```
function gui3
prompt = {'Number of agents (N):','Confidence (beta):','Goal (
    L):','Max power (Pmass):','Max cost (Cmax):','Min cost (Cmin
    ):','Number of points on graph (Npoints)','Gaussian flag'};
dlgtitle = 'Input';
dims = [1 35];
definput = {'100','1e-6','5000','400','60','20','1000','0'};
answer = inputdlg(prompt,dlgtitle,dims,definput) %cell array

Pemp=scenario_main(str2num(answer{1}),str2num(answer{2}),
    str2num(answer{3}),str2num(answer{4}),str2num(answer{6}),
    str2num(answer{5}),str2num(answer{7}),str2num(answer{8}))
end
```

A.2.2 Air cargo loading problem

The function *scenario_main2.m* is used to run the complete simulation, calling all the other auxiliary functions. Its inputs are:

- N, number of initial extractions;
- beta the confidence parameter;
- W the weight capacity of the plane;
- V the volume capacity of the plane;
- U_{min} and U_{max} the minimum and maximum values for the cost coefficients;

- D_{min} and D_{max} the minimum and maximum values for the distribution from which the upper bounds are extracted;
- $Npoints$ the number of times the empirical stability index has to be evaluated;
- $flag$ a parameter that when set to 1 switches the problem to the chopped normal distribution case.

```
function x= scenario_main2(N,beta,W,V,Umax,Umin,Dmin,Dmax,
    Npoints,flag)
%% Flight weight capacity problem
    d=N;
    ngoods=N;
    sigma=zeros(1,Npoints);
    Pemp=zeros(1,Npoints);
    fprintf('Progress:\n');
    fprintf(['\n' repmat('.',1,Npoints) '\n\n']);
    parfor g=1:Npoints
        fprintf('\b|\n');

        [x,sigmaori,f,Values,Aeq,Beq,Aineq,Bineq,Wei,lb,ub]=
            flightsolve(N,beta,W,V,Umax,Umin,Dmin,Dmax,ngoods,
            flag);
        sigma(1,g)=sigmaori;

        Phat = Monte_Carlo2(N,Values,x,f,Aeq,Beq,Aineq,Bineq,
            Wei,lb,ub,Dmin,Dmax,Umin,Umax,ngoods,flag);
        Pemp(1,g)=Phat;
    end
figure
```

```
hold on
plot(sigma,Pemp,'*')
hold on
plot(epsilon(d,N,beta))
title(['Empirical probability of solution change vs upper
    bound, N = ',num2str(N),' W = ',num2str(W),' V = ',num2str(V
    ), ' Dmin = ',num2str(Dmin), ' Dmax = ',num2str(Dmax)])
xlabel('No. support agents [k]')
ylabel('Empirical probability of solution change')

end
```

Solve the problem - flightsolve.m

```
function [x,sigmaori,f,Values,Aeq,Beq,Aineq,Bineq,W,lb,ub,
    exitflag]=flightsolve(N,beta,W,V,Umax,Umin,Dmin,Dmax,ngoods,
    flag)

Values=randi([Umin,Umax],1,ngoods);
minDemand=randi([Dmin,Dmax],1,ngoods);
if flag == 1
     px=Dmin:Dmax;
     mi=(Dmax-Dmin)/2;
     dev=10000;
     p=1./(dev*sqrt(2*pi))*exp((-(px-mi).^2)./(2*dev^2));
     minDemand=round(randpdf(p,px,[ngoods,1])');

     alpha=rand(1,ngoods);
     Values=alpha.*minDemand;
end
```

```

ub=minDemand';
lb=zeros(ngoods,1);
Densities=randi([950,7000],1,ngoods);
A=ones(1,ngoods);
B=(1./Densities);
Aeq=[];
Beq=[];
Aineq=[A;B];
Bineq=[W;V];
f=-Values';
[x,fval,exitflag] = cplexlp(f,Aineq,Bineq,Aeq,Beq,lb,ub);

if exitflag ~= 1
    x=[];
    return
end

sigmaori=0;
for t=1:length(x)
    if x(t,1) ~= 0
    sigmaori=sigmaori+1;
    end
end

end
```

Stability index empirical evaluation - Monte_Carlo2.m

```
function [Phat]=Monte_Carlo2(N,Values,x,f,Aeq,Beq,Aineq,Bineq,
   W,lb ,ub ,Dmin ,Dmax,Umin ,Umax,ngoods , flag )

M=50*N;

      viol=zeros(1,M);
      Aeq=[];
      Beq=[];
      parfor  j=1:M
            Demandnew=randi([Dmin,Dmax]);
            Value=randi([Umin,Umax]);

 if  flag == 1
      px=Dmin:Dmax;
      mi=(Dmax-Dmin)/2;
      dev=1000;
      p=1./(dev*sqrt(2*pi))*exp((-(px-mi).^2)./(2*dev^2));
      Demandnew=round(randpdf(p,px,[1,1])');
      alpha=rand(1);
      Value=alpha.*Demandnew;
 end

            Density=1/randi([950,7000],1);
            Anew=[Aineq  [1;Density]];
            fnew=[f  ;-Value];
            ubnew=[ub  ;Demandnew];
            lbnew=[lb  ;0];
            xnew = cplexlp(fnew,Anew,Bineq,Aeq,Beq,lbnew,ubnew);

```

```
        if not(isequal(0,xnew(end)))
            viol(1,j)=1;

        end

    end
    nviol=sum(viol);
    Phat=nviol/M;
end
```

Theoretical bound - epsilon.m

```
function out = epsilon(d,N,bet)
out = zeros(d+1,1);
for k = 0:d
m = [k:1:N];
aux1 = sum(triu(log(ones(N-k+1,1)*m),1),2);
aux2 = sum(triu(log(ones(N-k+1,1)*(m-k)),1),2);
coeffs = aux2-aux1;
t1 = 0;
t2 = 1;
while t2-t1 > 1e-10
t = (t1+t2)/2;
val = 1 - bet/(N+1)*sum( exp(coeffs-(N-m')*log(t)) );
if val >= 0
t2 = t;


else
t1 = t;
end
```

```
end
out(k+1) = 1-t1;
end
```

Optional graphical prompt - gui2.m Running this function gives a graphical prompt to make running the simulation easier. It comes with preset values that can be modified.

```
function gui2

prompt = {'Number of extractions (N):','Confidence (beta):','
    Weight capacity(W):','Volume capacity(V):','Max value (Umax)
    :','Min value (Umin):','Min demand (Dmin):','Max demand (
    Dmax):','Number of points on graph (Npoints)','Switch to
    chopped gaussian (1 for yes)'};
dlgtitle = 'Input';
dims = [1 35];
definput = {'100','1e-6','20882','44','60','20','1000','2000',
    '1000','0'}; %possibly lower Dmax
answer = inputdlg(prompt,dlgtitle,dims,definput) %cell array

scenario_main2(str2num(answer{1}),str2num(answer{2}),str2num(
    answer{3}),str2num(answer{4}),str2num(answer{5}),str2num(
    answer{6}),str2num(answer{7}),str2num(answer{8}),str2num(
    answer{9}),str2num(answer{10}))

end
```

References

[1] G. Calafiore and M. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[2] M. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.

[3] M. Campi, S. Garatti, and M. Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149 – 157, 2009.

[4] M. C. Campi and S. Garatti. Wait-and-judge scenario optimization. *Mathematical Programming*, 167(1):155–189, January 2018.

[5] Alessandro Falsone, Kostas Margellos, Simone Garatti, and Maria Prandini. Linear programs for resource sharing among heterogeneous agents: The effect of random agent arrivals. *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3853–3858, 2017.

[6] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1st edition, 1997.

[7] David Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*, volume 67. 01 1984.

[8] Simone Garatti and Marco Campi. Risk and complexity in scenario optimization. *Mathematical Programming*, 11 2019.

[9] Kurt Eisemann. The primal-dual method for bounded variables. *Operations Research*, 12(1):110–121, 1964.

[10] Ahsan Ashfaq and Akif Khan. Optimization of economic load dispatch problem by linear programming modified methodology. 05 2014.

[11] Felix Brandt. *The Air Cargo Load Planning Problem*. PhD thesis, 09 2017.

[12] Kuancheng Huang and Heng Lu. A linear programming-based method for the network revenue management problem of air cargo. *Transportation Research Part C: Emerging Technologies*, 7, 05 2015.

[13] Marco Campi and Simone Garatti. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. *Journal of Optimization Theory and Applications*, 148:257–280, 02 2011.