Università di Pisa and Scuola Superiore Sant'Anna



Depts. of Computer Science and Information Engineering and
Scuola Superiore Sant'Anna
Master Degree in Computer Science and Networking

**Master Thesis**

# Error Tolerant Descent Methods for Computing Equilibria

**Candidate**
**Roberto Ladu**

**Supervisor**
**Prof. Giancarlo Bigi**

**Examiner**
**Prof. Giuseppe Prencipe**

**Accademic Year 2012/2013**

# Contents

# Introduction

Equilibrium Problems (in short, EP) are a unified approach to express Optimization Problems, Variational Inequalities, Nash Equilibrium, Fixed Point and Saddle Point problems within the very same general mathematical framework. In the last decade they received an increasing interest mainly because many theoretical and algorithmic results developed for one of these models can be extended to the others through the unifying language provided by the common format of EPs. Moreover, they benefit from the vast number of concrete applications that all the above models embrace. As a result, their applicative domain ranges from Engineering (e.g., desing of cognitive radio systems) to Economies (e.g., competition over production and/or distribution) and Computer Science (e.g., cloud computing), and in general EPs arise in the modelling of competitive agents systems.

The format of the Equilibrium Problem reads

$$\text{find } x \in C \text{ s.t. } f(x,y) \geq 0 \quad \forall y \in C,$$

where $C \subseteq \mathbb{R}^n$ is a nonempty closed set and $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is an equilibrium bifunction, i.e. $f(x,x) = 0$ for all $x \in C$.

Several kinds of methods have been proposed to solve (EP): fixed point methods, proximal point methods, regularization methods, Tikhonov-Browder methods and extragradient methods. This thesis focuses on the class of the so-called *descent methods*.

Descent methods come from Optimization Problems aiming at minimizing a function $f : \mathbb{R}^n \to \mathbb{R}$ (the reader can refer to Appendix B for further details on Optimization Problems). Their distinctive feature is that the sequence of points $x_k \in \mathbb{R}^n$ they generate satisfies $f(x_k) > f(x_{k+1})$. As we will see, (EP) can be reformulated as an Optimization Problem and then solved using descent methods.

The main complexity of all the techniques mentioned above lies in the solution of *inner* Optimization Problems, at least one of which has to be solved at every step. In this thesis, we propose two algorithms that ammortize the cost of this sub-problem relying on an *error tolerant* approach, which, roughly speaking, consists in computing only a sub-optimal solution of the inner problem instead of a truly optimal one. In this way the

inner problem solver can end his computation earlier, hopefully enhancing the overall computational cost. As a consequence, our work deals also with practical methods for computing such sub-optimal solutions and control their quality.

It is worth to notice that the two algorithms are tightly coupled to two *non monotone* descent methods: the Enhanced Basic Algorithm and the Nonlinear Constraint Approximation Algorithm. In fact, the two algorithms could be considered an error tolerant extension of theirs.

The thesis is organized in four chapters.

In Chapter 1, we formally introduce (EP) and its reformulation as an Optimization Problem. Moreover, we describe two descent methods: the Basic Algorithm and the Enhanced Basic Algorithm, which differ mainly for the assumptions upon which are built. Finally, we show our first error tolerant algorithm and its proof of correctness.

Chapter 2 deals with the case in which (EP) involves nonlinear inequality constraints. Firstly we present the Nonlinear Constraints Approximation Algorithm, which exploits the particular structure of the constraints, and then we propose our second error tolerant algorithm and its proof of correctness.

This second error tolerant algorithm is more complex than the previous one in Chapter 1. In fact, due to the approximation of the constraints, the Nonlinear Constraints Approximation Algorithm requires to know optimal dual solutions of the inner problem, which in turn cannot be computed exactly in the error tolerant version. In order to overcome this issue we introduce a notion of dual approximated solution which is suitable for our aims. In addition, we develop some methods that can be used to compute jointly the primal and dual approximated solutions. These methods are the subject of Chapter 3.

In Chapter 3, firstly we give a theoretical insight on the nature of sub-optimal solutions, then we exploit these results to prove the correctness of two methods. The first is derived from the Frank-Wolfe algorithm and it reduces to a sequence of linear programs. The second is derived from the Fiacco and McCormick's Barrier Method and it reduces to a single unconstrained nonlinear minimization problem. The main difficulty here originates from the fact that we need to control the *quality* of the sub-optimal solution. Indeed, the error tolerant algorithms non only require an ongoing refinement of the sub-optimal solutions, but also require to know a priori their quality.

Chapter 4 presents an original application that is used as a test case for comparing our algorithms. The choosen applicative domain is Cloud Computing. We consider the point of view of an IaaS provider that sells virtual machines with a certain computational capacity and communication bandwidth to the users. Clearly, a flow allocation in the physical provider's network has to match the communication bandwidth bought by the users. On the other hand, we suppose that the quantity of bandwidth bought depends

also on prices the provider sets. In our scenario the provider has already allocated the virtual machines of his tenant and has a stochastic knowledge of the communication bandwidth intensity that the users will buy as a function of the transmission price. The objective of the provider is to choose proper network routing and transmission prices in order to achieve efficient allocations and high revenues. On the other hand, the users are interested in accessing the best communication bandwidth at the most convenient price. The system is modeled as a non cooperative game and the algorithms developed in the previous chapters are applied to find an equilibrium. Numerical results are shown at the end of the chapter and they show some improvement of performance with respect to the corresponding exact algorithm.

The thesis includes also two appendices. In the first we recall some important theorems about point-to-set maps in mathematical programming, while in the second we summarize the most relevant results about nonlinear optimization.

# Chapter 1

# Descent Methods for Equilibrium Problems

## 1.1 Introduction to the Problem

In what follows we assume $C \subset \mathbb{R}^n$ to be a nonempty compact convex set and $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, to be a bifunction such that

- $f$ is continuously differentiable,
- $f(x, x) = 0$ for any $x \in C$
- $f(x, \cdot)$ is convex for any $x \in C$.

**Definition 1.1** (Equilibrium Problem)
We define the equilibrium problem (EP) as follows:

$$\text{find } x \in C \text{ s.t. } f(x, y) \geq 0 \quad \forall y \in C. \tag{EP}$$

(EP) is a very general model with an expressivity equivalent to the one of Optimization Problems and of Variational Inequalities Problems (see [BCPP13] for more details).

Another class of problems, which is relevant for the application we will describe in the Chapter 4 , is given by the Nash Equilibrium problems. In the next paragraph we introduce the notion of Game and of Nash Equilibrium, and show how the problem of finding a Nash Equilibrium can be expressed through (EP).

### Applications to Game Theory

**Definition 1.2** (Game)
A game $G$ is a triple $G = \langle P, \{S_p\}_{p \in P}, \{u_p\}_{p \in P} \rangle$ where

- $P = \{1, 2, \dots, N\}$ is the set of players,
- $S_p \subseteq \mathbb{R}^{n_p}$ is the set of strategies of player p,
- $u_p : \prod_{j=1}^{N} S_j \to \mathbb{R}$ is the payoff function of player $p$.

Usually given a point $x \in \prod_{p=1}^{N} S_p$ we will denote as $x_p$ the vector of the components of $x$ related to the player $p$ and as $x_{-p}$ the vector of the remaining components of $x$. With a little abuse of notation we will use $u_p(x_i, x_{-i})$ or $u_p(x)$ equivalently and in general, we will act as if for any two vectors $y, x \in \prod_{j=1}^{N} S_j$ the concatenation of $y_i$ and $x_{-i}$ preserve the original order of the components in the vectors $y$ and $x$.

The interpretation of the above definition is straightforward: there are $N$ players who can choose a strategy ($x_p \in S_p$ is the strategy choosen by player $p$) which we can imagine as a trajectory of moves in the game. Once each player has choosen his strategy, the player $p$ can evaluate how good this scenario $x = (s_1, s_2, \dots, s_N)$ is for him through his utility function $u_p$. We assume that each player is interested in choosing his strategy in a way that his cost function is *minimized*.

A game can be given defining for each player the optimization program that determines the *best response*, i.e., the optimal strategy with respect to its payoff function when the strategies of the other player are fixed. Formally, given $s_{-p} \in \prod_{i \neq p} S_i$ the best response of the player $p$ is

$$\min\{u_p(s_p, s_{-p}) \mid s_p \in S_p\}.$$

A common problem in Game Theory is the determination of the *Nash Equilibria* (Nash Equilibrium Problem (NEP)) for a certain Game.

**Definition 1.3** (Nash Equilibrium)
Let $G$ be a game. A point $x^* \in \prod_{p=1}^{N} S_p$ is called Nash Equilibrium for the game $G$ iff for any player $p$,

$$\forall y \in S_p, \quad u_p(y, x^*_{-p}) \geq u_p(x^*).$$

We can interpret a Nash Equilibrium as a scenario in which every player isn't interested in changing his strategy unilateraly because it would result in an increasing of his payoff function.

**Definition 1.4** (Nikaido-Isoda Bifunction ([NI55]))
Let $G$ be a game. Then, we define the Nikaido-Isoda bifunction $NI : \prod_{p=1}^{N} \mathbb{R}^{n_p} \times \prod_{p=1}^{N} \mathbb{R}^{n_p} \to \mathbb{R}$ as follows

$$NI(x, y) = \sum_{p=1}^{N} u_p(y_p, x_{-p}) - u_p(x).$$

The Nikaido-Isoda bifunction allows us to express the problem of finding a Nash Equilibrium as an Equilibrium Problem. Indeed, it is easy to see that the solutions of the following Equilibrium Problem:

$$\text{find } x \in \prod_{p=1}^{N} S_p \ : \ NI(x,y) \geq 0 \ \forall y \in \prod_{p=1}^{N} S_p,$$

are Nash Equilibria and viceversa any Nash Equilibrium solves the above problem.

### 1.1.1 Restating (EP) as an Optimization Problem

In order to design a descent method to solve (EP) we clearly need a function to descend. In this section we show how (EP) can be transformed in an optimization problem together with a family of functions suitable to the implementation of descent methods.

**Definition 1.5** (Gap Function for (EP))
A function $f : \mathbb{R}^n \to \mathbb{R}$ is called *gap function* for (EP) iff satisfies:

a) $f(x) \geq 0$ for any $x \in C$,

b) $f(x^*) = 0$ iff $x^*$ solves (EP).

It is straightforward that a function satisfying the above property can be used to reformulate (EP) as an optimization problem. One well known gap function is the function $\varphi$ defined below.

**Definition 1.6** ($\varphi$)
We define the function $\varphi : \mathbb{R}^n \to \mathbb{R}$ as

$$\varphi(x) = -\min_{y \in C} f(x,y).$$

**Theorem 1.1**
*$\varphi$ is a gap function for (EP).*

*Proof.* Since $0 = f(x,x) \geq \min\{f(x,y) : \ y \in C\}$, then $\varphi(x) \geq 0$ for any $x \in C$, hence a) holds. Furthermore, if $-\varphi(x^*) = \min\{f(x,y) : \in C\} = 0$, then $f(x,y) \geq 0$ for any $y \in C$ and therefore $x^*$ solves (EP). On the other hand is straightforward that if $x^*$ solves (EP) then $0 = f(x,x)$ is the minimum value. Thus, b) holds as well.

□

As a result of the Theorem 1.1, we could solve (EP) through the optimization problem

$$\min_{x \in C} \varphi(x). \tag{1.1}$$

Unfortunately, the assumptions we made on $f$ are not strong enough to ensure two properties which are fundamental in order to develop efficient descent methods. Firstly, we can't guarantee the differentiability of $\varphi$. In fact, exploiting THEOREM A.2 with $\Omega(x) = C$ and $f = -f$ we get that the directional derivative at point $\overline{x} \in C$ and direction $d \in \mathbb{R}^n$ is

$$\varphi'(\overline{x}; d) = \max_{y \in M(\overline{x})} \langle \nabla_x - f(\overline{x}, y), d \rangle$$
$$\text{where} \quad M(\overline{x}) = \{y \in C |\ \varphi(\overline{x}) \leq -f(\overline{x}, y)\},$$

In general, this expression is not linear as a function of $d$.

Secondly, also in the good case when $\varphi$ is differentiable, many optimization methods ensure only to find a stationary point. Thus, a way to ensure that the stationarity of a point implies that it is a minimizer would be valuable. Notice that usually this would be guaranteed by the convexity of the gap function $\varphi$, but unfortunately in general, $\varphi$ isn't convex.

### 1.1.2 Auxiliary Problem Principle

A possible solution to achieve continuous differentiability is to add a regularizing term $h$ to the bifunction $f$ in such a way that the solutions of the equilibrium problem do not change. This approach appears in [Mas03], inspired by the work on Variational Inequalities of Zhu and Marcotte ([ZM94]) and Fukushima ([Fuk92]).

**ASSUMPTION 1.1** (Regularizing bifunction $h$)
*We assume $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ to be a continuously differentiable bifunction over $\mathbb{R}^n \times \mathbb{R}^n$ such that for any $x \in C$*

- $h(x, y) \geq 0 \quad \forall y \in C$
- $h(x, x) = 0$,
- $h(x, \cdot)$ *is strictly convex,*
- $\nabla_y h(x, x) = 0$.

An example is the bifunction

$$h(x, y) = \frac{1}{2}\|x - y\|^2. \tag{1.2}$$

In general, the whole family of Bregman distances provides bifunctions which are suitable to play the role of $h$.

**Definition 1.7** ($f_\alpha$)
Given $\alpha > 0$, we define the bifunction $f_\alpha : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ as

$$f_\alpha(x, y) = f(x, y) + \alpha h(x, y).$$

Notice that $f_\alpha$ inherits all the proprieties of $f$, in addition $f_\alpha(x, \cdot)$ is strictly convex for any $x \in C$ which is a desirable propriety if we whish to solve the problem $\min_{y \in C} f_\alpha(x, y)$.

We can introduce definitions as we did for $f$, defining the Equilibrium Problem and gap function corresponding to $f_\alpha$.

**Definition 1.8** ($\alpha$- Regularized Equilibrium Problem)
Given $\alpha > 0$, we define the $\alpha$ regularized equilibrium problem as follows:

$$\text{find } x \in C \text{ s.t. } f_\alpha(x, y) \geq 0 \quad \forall y \in C. \tag{$\alpha$-EP}$$

**Definition 1.9**
Given $\alpha > 0$, $x \in C$ we define the optimization problem $(P_x^\alpha)$ as

$$\min_{y \in C} f_\alpha(x, y). \tag{$P_x^\alpha$}$$

Notice that since $f_\alpha$ is strictly convex the solution of $(P_x^\alpha)$ is unique.

**Definition 1.10**
Given $\alpha > 0$, we define the functions $\varphi_\alpha : \mathbb{R}^n \to \mathbb{R}$ and $y_\alpha : \mathbb{R}^n \to \mathbb{R}^n$ to be respectively the changed sign optimum value and the minimizer of $(P_x^\alpha)$, i.e.

$$\varphi_\alpha(x) = -\min_{y \in C} f_\alpha(x, y),$$
$$y_\alpha(x) = \arg\min_{y \in C} f_\alpha(x, y).$$

It is straightforward to check that by Definition 1.5 $\varphi_\alpha$ is a gap function for ($\alpha$-EP), moreover we have that $\varphi_\alpha(x) = -f_\alpha(x, y_\alpha(x))$.

The following Theorem ensures that we can work with $f_\alpha$ and $\varphi_\alpha$ in order to solve (EP) without loss of generality. In other words, it shows that $\varphi_\alpha$ is also a gap function for (EP).

**Theorem 1.2** ((EP) and ($\alpha$-EP) Equivalency [Mas00])
*Given any $\alpha > 0$, and $x \in C$, then $x$ solves (EP) iff $x$ solves ($\alpha$-(EP)).*

*Proof.* Suppose that $\alpha > 0$ and that $x^*$ solves ($\alpha$-EP), then Definition 1.5 guarantees $\varphi_\alpha(x) = 0$, which implies $\min\{ f_\alpha(x, y) : y \in C\} = 0$. Since $f_\alpha(x, \cdot)$ is strictly convex by construction (sum of a convex and strictly convex function), the unique $y$ that minimizes

$f_\alpha(x,\cdot)$ has to be $x^*$ (in fact we have $f_\alpha(x^*,x^*)=0$). Indeed, because $x^*$ is optimal, it is necessarily a stationary point for $f_\alpha(x^*,\cdot)$:

$$\langle \nabla_y f_\alpha(x^*,x^*), z-x^*\rangle \geq 0 \quad \forall z \in C.$$

This equation jointly with the fact that $\nabla_y h(x^*,x^*)=0$ implies that

$$\langle \nabla_y f(x^*,x^*), z-x^*\rangle \geq 0 \quad \forall z \in C,$$

and so $x^*$ minimizes also $f(x,\cdot)$. Thus, $\varphi(x)=-f(x^*,x^*)=0$ and $x^*$ solves (EP).

Now, suppose that $x^*$ solves (EP), then by Theorem 1.1 $\varphi(x^*)=0$, but we know that for any $x \in C$ $-f(x,y) \geq -f(x,y)-\alpha h(x,y)$ because Assumption 1.1 guarantees $h(x,y)\geq 0$. Therefore, $\varphi(x)\geq\varphi_\alpha(x)\geq 0$, where the last inequality follows from Definition 1.5, thus $\varphi(x^*)=0$ implies $\varphi_\alpha(x^*)=0$, and this concludes the proof. □

The following theorems justify the exploitation of the regularizing bifunction $h$. The next theorem ensures the continuity of $y_\alpha$ and $\varphi_\alpha$ in the more general case when $\alpha$ is considered to be a variable.

**Theorem 1.3**

*The functions $\phi : \mathbb{R}^n \times int\ \mathbb{R}_+ \to \mathbb{R}$ and $y : \mathbb{R}^n \times int\ \mathbb{R}_+ \to \mathbb{R}^n$ defined as*

$$\begin{aligned} \phi(x,\alpha) &= \min_{y\in C} f(x,y)+\alpha h(x,y) \\ y(x,\alpha) &= \arg\min_{y\in C} f(x,y)+\alpha h(x,y) \end{aligned} \tag{1.3}$$

*are continuous over $\mathbb{R}^n \times int\ \mathbb{R}_+$.*

*Proof.* We can invoke Theorem A.1 with $\Omega(x)=C$ and $f(x,\alpha,y)=-f(x,y)-\alpha h(x,y)$ to obtain that $\phi$ is continuous over $\mathbb{R}^n \times \text{int}\ \mathbb{R}_+$. The assumptions on $f$ are met because of the proprieties $(x,\alpha,y)\mapsto -f(x,y)-\alpha h(x,y)$ inherits from $f$ and $h$ (viz. continuity) and it is easy to see that the constant point-to-set map $\Omega$ is continuous and uniformly compact.

Suppose that $y(x,\alpha)$ is not continuos over $\mathbb{R}^n \times \text{int}\ \mathbb{R}_+$, then there must exists sequences $\{x_k\}$ and $\{\alpha_k\}$ such that

$$\begin{aligned} &x_k \to x^* \in C, \\ &\alpha_k \to \alpha^* \in \text{int}\ \mathbb{R}_+, \\ &y(x_k,\alpha_k) \to \overline{y} \neq y(x^*,\alpha^*) \text{ or } \{y(x_k,\alpha_k)\} \text{ diverges.} \end{aligned} \tag{1.4}$$

Suppose that the $y(x_k, \alpha_k) \to \overline{y}$. Since $C$ is compact $\overline{y} \in C$. Furthermore by the continuity of $\phi$ and $(x, \alpha, y) \mapsto -f(x, y) - \alpha h(x, y)$ we have that

$$\phi(x^*, \alpha^*) = \lim_k -f(x_k, y(x_k, \alpha_k)) - \alpha_k h(x_k, y(x_k, \alpha_k)) = -f(x^*, \overline{y}) - \alpha^* h(x^*, \overline{y}).$$

This implies that $\overline{y}$ is a minimizer of $-f_{\alpha^*}(x^*, \cdot)$ over $C$ but since $-f_{\alpha^*}(x^*, \cdot)$ is strictly convex the minimizer has to be unique and thus $\overline{y} = y(x^*, \alpha^*)$, in contradiction with (1.4).

Now suppose that $\{y(x_k, \alpha_k)\}$ diverges. At least one between $\limsup_k y(x_k, \alpha_k)$ and $\liminf_k y(x_k, \alpha_k)$ must be different from $y(x^*, \alpha^*)$, otherwise we would have that $y(x_k, \alpha_k) \to y(x^*, \alpha^*)$. Suppose without loss of generality that $\limsup_k y(x_k, \alpha_k) = \overline{y}_2 \neq y(x^*, \alpha^*)$ then we have found a sequence converging to a value different from $y(x^*, \alpha^*)$ and we can repeat the same reasoning done for the previous case. □

As a corollary of Theorem 1.3, keeping $\alpha$ fixed we obtain the following result:

**Corollary 1.1**

*$y_\alpha$ is continuous over $\mathbb{R}^n$.*

**Theorem 1.4**

*Given $\alpha > 0$, $x \in C$ solves (EP) iff $x = y_\alpha(x)$*

*Proof.* Suppose that $x^* \in C$ solves (EP) then by Theorem 1.2 and Definition 1.5 we have that $f_\alpha(x^*, y_\alpha(x^*)) = f(x^*, x^*) = 0$. Since $f_\alpha$ is strictly convex the minimizer must be unique and therefore $y_\alpha(x^*) = x^*$. Now suppose that $x^* = y_\alpha(x^*)$, thus $\varphi_\alpha(x^*) = 0$ and by Definition 1.5 $x^*$ solves ($\alpha$-EP) and thus (EP). □

**Theorem 1.5**

*$\varphi_\alpha$ is continuously differentiable over $C$ with the following gradient at point $x$:*

$$\nabla \varphi_\alpha(x) = -\nabla_x f_\alpha(x, y_\alpha(x)). \tag{1.5}$$

*Proof.* We can invoke Theorem A.2 with $\Omega(x) = C$ and $f = -f_\alpha$ to obtain that for any $\overline{x}, d \in X$, the directional derivative $\varphi'_\alpha(\overline{x}; d)$ exists and

$$\varphi'_\alpha(\overline{x}; d) = \max_{y \in M(\overline{x})} \langle \nabla_x - f_\alpha(\overline{x}, y), d \rangle$$
$$\text{where} \quad M(\overline{x}) = \{y \in C |\ \varphi_\alpha(\overline{x}) \leq -f_\alpha(\overline{x}, y)\}.$$

But because $f_\alpha(\overline{x}, \cdot)$ is strictly convex $M(\overline{x}) = \{y_\alpha(\overline{x})\}$ is a singleton. Thus,

$$\varphi'_\alpha(\overline{x}; d) = -\langle \nabla_x f_\alpha(\overline{x}, y_\alpha(\overline{x})), d \rangle.$$

The continuity follows from the continuity of $\nabla_x f$, $\nabla_x h$, which we supposed and from the continuity of $y_\alpha$, which we proved in Corollary 1.1. Therefore, $\varphi_\alpha$ is differentiable and (1.5) holds. □

To Summarize, in this sub-section we have shown how we can conveniently switch to our main problem (EP) to a family of Equilibrium Problems ($\alpha$-EP) gaining the continuous differentiability of the gap function. It is still not clear how we can overcome the second issue (stationarity): at the best of our knowledge, two solutions are possible. Both of them consists in strenghten the assumptions on $f$. The difference between the two solutions lies in the choosen assumption: the first is called *strictly $\nabla$-monotonicity*, the second is called *c-monotonicity*. We will introduce the assumptions and the corresponding algorithms in the following sections.

## 1.2 Descent Method Under strictly $\nabla$-monotonicity Assumption

**Definition 1.11**
A differentiable bifunction $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called

- $\nabla$-monotone over $C$ if

$$\langle \nabla_x g(x,y) + \nabla_y g(x,y), y - x \rangle \geq 0, \quad \forall x, y \in C;$$

- strictly $\nabla$-monotone over $C$ if

$$\langle \nabla_x g(x,y) + \nabla_y g(x,y), y - x \rangle > 0, \quad \forall x, y \in C;$$

- strongly $\nabla$-monotone over $C$ if there exists $\tau > 0$ such that

$$\langle \nabla_x g(x,y) + \nabla_y g(x,y), y - x \rangle \geq \tau \|y - x\|^2, \quad \forall x, y \in C.$$

The next theorem suggests a possible descent direction under the assumption that $f_\alpha$ is strictly $\nabla$-monotone. Notice that choosing a regularizing bifunction as defined in Eq. (1.2) the strict $\nabla$-monotonicity of $f$ is a sufficient and necessary condition to $f_\alpha$ to be strictly $\nabla$-monotone. Indeed, in this case

$$\langle \nabla_x h(x,y) + \nabla_y h(x,y), y - x \rangle = \langle x - y, y - x \rangle + \langle y - x, y - x \rangle = 0.$$

The next theorem motivates the above assumption and will be heavily exploited in the algorithm.

**Theorem 1.6** ([Mas03] [BCP09])
*Suppose that $x \in C$ is not a solution of (EP). If $f_\alpha$ is strictly $\nabla$-monotone on $C$, then*

$$\langle \nabla\varphi_\alpha(x), y_\alpha(x) - x\rangle < 0.$$

*Proof.* We have that for any $x \in C$:

$$\begin{aligned} &\langle \nabla\varphi_\alpha(x), y_\alpha(x) - x\rangle \\ (\text{Theorem 1.5}) = &- \langle \nabla_x f_\alpha(x, y_\alpha(x)), y_\alpha(x) - x\rangle \\ (\text{strict } \nabla\text{-monotonicity}) < &\langle \nabla_y f_\alpha(x, y_\alpha(x)), y_\alpha(x) - x\rangle \\ \leq &0, \end{aligned}$$

where the last inequality follows from the fact that $y_\alpha(x)$ is the minimizer over $C$ of $f_\alpha$. □

**Theorem 1.7** ([BCP09])
*Suppose that $f_\alpha$ is strictly $\nabla$-monotone. If $\overline{x} \in C$ is a stationary point of $\varphi_\alpha$ over $C$, i.e.*

$$\langle \nabla\varphi_\alpha(\overline{x}), y - \overline{x}\rangle \geq 0 \quad \forall y \in C,$$

*then $\overline{x}$ solves (EP).*

*Proof.* By contraddiction, suppose that $\overline{x}$ doesn't solve (EP), then by Theorem 1.6 we have that

$$\langle \nabla\varphi_\alpha(\overline{x}), y_\alpha(\overline{x}) - \overline{x}\rangle < 0$$

but this would contradict the assumption of stationarity of $\overline{x}$. □

### Basic Algorithm

The algorithm below (we name Basic Algorithm) is due to Mastroeni ([Mas03]) and have been designed for strictly $\nabla$-monotone $f$.

At every iteration $k$ the algorithm maintains an approximated solution for (EP) $x_k$ and compute the next one as $x_{k+1} = x_k + t_k d_k$. $d_k$ is setted to $y_\alpha(x_k) - x_k$. Thanks to Theorem 1.6, we notice that it is a descent direction for $\varphi_\alpha$, in the sense that there must exists a steplength $\bar{t} > 0$ such that

$$\varphi_\alpha(x_k + \bar{t} d_k) < \varphi(x_k).$$

$t_k$ is the stepsize choosen by the algorithm, aiming to obtain the maximum decrement of the function $\varphi_\alpha$ in the direction $d_k$ without exiting from $C$. In other words, $t_k = \arg\min_{t\in[0,1]} \varphi_\alpha(x_k + t d_k)$.

Algorithm 1.1: Basic Algorithm

1. Set $k = 0$ and choose $x_0 \in C$
2. Compute $y_k = \arg\min_{y \in C} f_\alpha(x_k, y)$
3. Set $d_k = y_k - x_k$
4. Compute $t_k = \arg\min_{t \in [0,1]} \varphi_\alpha(x_k + t d_k)$
5. $x_{k+1} = x_k + t_k d_k$
6. If $\|x_{k+1} - x_k\| = 0$
   then STOP
   else Set $k = k + 1$ and GOTO Step 2.

Formally, the correctness is stated by the following theorem:

**Theorem 1.8** (Theorem 3.2 in [Mas03])
*Let $f_\alpha$ be a strictly $\nabla$-monotone bifunction, then for any $x_0 \in C$, the sequence $\{x_k\}$ generated by Algorithm 1.1 belongs to the set $C$ and any accumulation point of $\{x_k\}$ is a solution of (EP).*

## 1.3 Descent Method Under c-monotonicity Assumption

### Considerations about $\nabla$-monotonicity

Previous assumptions are not completely satisfactory since many bifunction of interests are not strictly $\nabla$-monotone nor they lead to a gap function $\varphi_\alpha$ such that all its stationary points over $C$ are minimizers. As an example[1] consider the bifunction $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ defined as $f(x, y) = x - y$ together with the set $C = [-1, 1]$. It is a $\nabla$-monotone function which is not strictly $\nabla$-monotone. The solution of (EP) is the point $x^* = 1$. Choosing $h(x, y) = \frac{(x-y)^2}{2}$ as regularizing bifunction, we find out that

$$y_\alpha(x) = \begin{cases} -1 \text{ if } x \in (-\infty, -1 - \alpha^{-1}), \\ x + \alpha^{-1} \text{ if } x \in [-1 - \alpha^{-1}, 1 - \alpha^{-1}], \\ 1 \text{ if } x \in (1 - \alpha^{-1}, +\infty), \end{cases}$$

and

$$\varphi_\alpha(x) = \begin{cases} -\frac{(1+x)(\alpha x + \alpha - 2)}{2} \text{ if } x \in (-\infty, -1 - \alpha^{-1}), \\ (2\alpha)^{-1} \text{ if } x \in [-1 - \alpha^{-1}, 1 - \alpha^{-1}], \\ \frac{(1-x)(\alpha x + 2 - \alpha)}{2} \text{ if } x \in (1 - \alpha^{-1}, +\infty). \end{cases}$$

The graph of $\varphi_\alpha$ is given in Figure 1.1. Notice that that every point in $[-1-\alpha^{-1}, 1-\alpha^{-1}]$ is a local minimum. Hence if we try to solve (EP) minimizing $\varphi_\alpha$ and employing an algorithm that stops when a stationary point is found, choosing a starting point $x_0 \in$

[1] This example is taken from [BCP09].

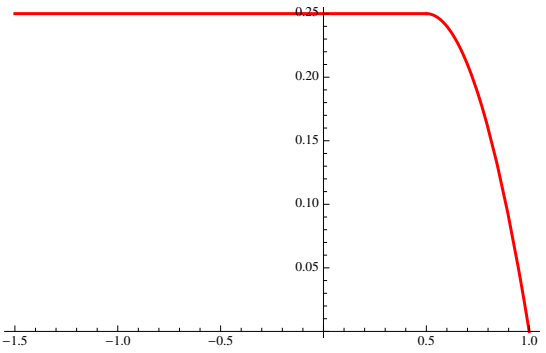



FIGURE 1.1
Plot of $\varphi_\alpha$ over $C$ for $\alpha = 2$.

$[-1-\alpha^{-1}, 1-\alpha^{-1}]$, the algorithm would stop at the first iteration, failing in finding the solution of (EP).

### c-monotonicity

A possible patch to this problem consists in changing our assumptions and consequently also the algorithm: instead of strict $\nabla$-monotonicity we suppose c-monotonicity.

**DEFINITION 1.12** ([BCP09], [BP14])
A differentiable bifunction $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called *c-monotone* if satisfies the following inequality:

$$g(x,y) + \langle \nabla_x g(x,y), y - x \rangle \geq 0 \quad \forall x, y \in C. \tag{1.6}$$

Notice that c-monotonicity is nor a stronger nor a weaker assumption than strictly $\nabla$-monotonicity. In fact, it can be checked (see Example 3.4 in [BP14]) that c-monotonicity doesn't imply strict $\nabla$ monotonicity and viceversa strict $\nabla$ monotonicity doesn't imply c-monotonicity.

Hence, although we have presented c-monotonicity as a patch to solve some particular issues it is way more: it virtually enlarges the set of problems we can handle.

Conversely, it can be proven that c-monotonicity is stronger than simple $\nabla$-monotonicity.

**THEOREM 1.9** (Theorem 3.1 in [BCP09])
*Let $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a differentiable bifunction, if $g$ is c-monotone, it is also $\nabla$-monotone.*

In the remaining part of this section supposing $f$ to be c-monotone allows us to achieve properties which are fundamental to design a suitable descent method.

**THEOREM 1.10** (Theorem 3.3 in [BCP09])
*Suppose that $f$ is c-monotone. If $x \in C$ is not a solution of (EP) then there exists $\overline{\alpha} > 0$ such that $x$ is not a stationary point for $\varphi_\alpha$ and $y_\alpha(x) - x$ is a descent direction for any $\alpha \in (0, \overline{\alpha}]$.*

THEOREM 1.10 allows to finding a descent direction changing the gap function simply descreasing parameter $\alpha$.

The next theorem provides an upper bound on the directional derivative of $\varphi_\alpha$ in direction $y_\alpha - x$.

**THEOREM 1.11** (Theorem 3.4 in [BCP09])
*If $f$ is c-monotone and $h$ is $\nabla$-monotone, it holds that*

$$\langle \nabla \varphi_\alpha(x), y_\alpha(x) - x \rangle \leq f(x, y_\alpha(x)) - \alpha(\langle \nabla_x h(x, y_\alpha(x)), y_\alpha(x) - x \rangle) \leq 0$$

*for any $\alpha > 0$ and $x \in C$.*

### Enhanced Basic Algorithm

The algorithm below is due to Bigi, Castellani and Pappalardo ([BCP09]). It exploits the considerations we developed so far in order to obtain a solution of (EP) under c-monotonicity assumption.

Algorithm 1.2: Enhanced Basic Algorithm

0. Fix $\eta, \gamma \in (0,1)$, $\beta \in (0, \eta)$ and a positive sequence $\delta_i \to 0$.
1. Choose $x_0 \in C$, set $\alpha_0 = \delta_0$, $k = 0$ and $i = 0$.
2. If $\varphi_{\alpha_k}(x_k) = 0$ STOP.
3. Compute $y_k = \arg\min_{y \in C} f_{\alpha_k}(x_k, y)$ and set $d_k = y_k - x_k$.
4. If
$$\varphi_{\alpha_k}(x_k) - \alpha_k(h(x_k, y_k) + \langle \nabla_x h(x_k, y_k), y_k - x_k \rangle) < -\eta \varphi_{\alpha_k}(x_k),$$
then set $\alpha_{k+1} = \alpha_k$, and $t_k = \gamma^m$
where $m$ is the smallest nonnegative integer such that
$$\varphi_{\alpha_k}(x_k + \gamma^m d_k - \varphi_{\alpha_k}(x_k) < -\beta \gamma^m \varphi_{\alpha_k}(x_k)$$
else set $i = i + 1$, $\alpha_{k+1} = \delta_i$ and $t_k = 0$.
5. Set $x_{k+1} = x_k + t_k d_k$.
6. Increase $k$ and GOTO 2.

Formally, the correctness is stated by the following theorem:

**THEOREM 1.12** (Theorem 3.6 in [BCP09])
*If $f$ is c-monotone and $h$ is $\nabla$-monotone, then ALGORITHM 1.2 stops at a solution of (EP) after a finite number of steps or it produces a sequence $\{x_k\}$ such that any of its cluster points solves (EP).*

## 1.4 Error Tolerant Descent Method Under c-monotonicity Assumption

In a recent article ([LPS13]) Di Lorenzo, Passacantando and Sciandrone proposed a descent method for (EP) that converges to a solution of (EP) without computing exact solutions of the problem (**??**). In fact their approach relies only on the computation of the so called $\varepsilon$-approximated solution of $(P_x^\alpha)$ defined as follows:

**Definition 1.13** ($\varepsilon$ approximated solution of $(P_x^\alpha)$)
Given $\varepsilon, \alpha > 0$ and $x \in C$ a point $\overline{y} \in \mathbb{R}^n$ it is called $\varepsilon$ approximated solution of $(P_x^\alpha)$ if satisfies

i) $\overline{y} \in C$

ii) $f_\alpha(x, \overline{y}) - \varepsilon \leq f_\alpha(x, y_\alpha(x)) \equiv -\varphi_\alpha(x)$.

Notice that condition i) implies that $f_\alpha(x, \overline{y}) \geq f_\alpha(x, y_\alpha(x))$.

The authors suppose that the sequence $\{\varepsilon_k\}$ of approximation parameters goes to 0 and develop a convergent method under the assumption of strict $\nabla$-monotonicity of $f$. Inspired by their work, we have designed an error tolerant algorithm that converges under the assumption of c-monotonicity of $f$.

Our tractation is structured as follows: firstly a couple of Lemmas concerning properties of the gap function for some particular sequence of parameters $\alpha$ and $\varepsilon$ are stated. It follows the algorithm pseudocode and its proof of correctness.

### 1.4.1 Auxiliary Lemmas

In what follows we will use the symbol $y_{\alpha\varepsilon x}$ to represent an $\varepsilon$-approximation of $(P_x^\alpha)$.

Notice that given $\alpha$ and $x$ there could be more then one $\varepsilon$ approximated solution of $(P_x^\alpha)$, instead, since $f_\alpha(x)$ is strictly convex the minimizer $y_\alpha(x)$ is unique.

In addition in this section we will strengthen our assumption on the bifunction $h$ as follows:

**Assumption 1.2**
*We assume $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ to be a continuously differentiable bifunction over $\mathbb{R}^n \times \mathbb{R}^n$ such that for any $x \in C$*

- $h(x, y) \geq 0 \quad \forall y \in C$,
- $h(x, x) = 0$,

– $h(x,\cdot)$ *is* strongly *convex,*

– $\nabla_y h(x,x) = 0$.

Now we are ready to state the first couple of Lemmas on the limit proprieties of $f_\alpha$.

**Lemma 1.1**
*Given the sequences*

– $\{\alpha_k\}$ *such that* $\alpha_k > 0$ *for any* $k$,

– $\{\varepsilon_k\}$*such that* $\varepsilon_k > 0$ *for any* $k$, $\varepsilon_k \to 0$,

– $\{x_k\}$ *such that* $x_k \in C$ *for any* $k$, $x_k \to x^* \in C$.

– $\{y^k_{\alpha\varepsilon x}\}$ *sequence of* $\varepsilon_k$ *approximations of* $(P^{\alpha_k}_{x_k})$, *i.e.,* $y^k_{\alpha\varepsilon x} \equiv y_{\alpha_k\varepsilon_k x_k}$ *for any* $k$,

*it holds*

*a)* $\lim_{k\to\infty} \|f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) - \varphi_{\alpha_k}(x_k)\| = 0$,

*b)* $\lim_{k\to\infty} \|y^k_{\alpha\varepsilon x} - y_{\alpha_k}(x_k)\| = 0$

*Proof.* Consider the difference $f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) - \varphi_{\alpha_k}(x_k)$, by Definition 1.13 it holds

$$0 \le f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) - \varphi_{\alpha_k}(x_k) \le \varepsilon_k.$$

Since $\varepsilon_k \to 0$, we have

$$\lim_{k\to\infty} f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) - \varphi_{\alpha_k}(x_k) = 0. \tag{1.7}$$

Thus part *a)* of the statement is proved. Furthermore, because of the strongly convexity of $f_\alpha$ (see Assumption 1.2), the following chain of inequalities holds:

$$\begin{aligned}
&f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) - \varphi_{\alpha_k}(x_k)\\
&= f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) - f_{\alpha_k}(x_k, y_{\alpha_k}(x_k))\\
&\ge \langle \nabla_x f_{\alpha_k}(x_k, y_{\alpha_k}(x_k)), y^k_{\alpha\varepsilon x} - y_{\alpha_k}(x_k)\rangle + M\|y^k_{\alpha\varepsilon x} - y_{\alpha_k}(x_k)\|^2\\
&\ge M\|y^k_{\alpha\varepsilon x} - y_{\alpha_k}(x_k)\|^2 \ge 0.
\end{aligned}$$

Where we have exploited the optimality of $y_{\alpha_k}(x_k)$. Since by (1.7) the LHS tends to 0, we have that $M\|y^k_{\alpha\varepsilon x} - y_{\alpha_k}(x_k)\|^2 \to 0$. □

**Lemma 1.2**
*Consider sequences* $\{\alpha_k\}$, $\{\varepsilon_k\}$, $\{x_k\}$ , *and* $\{y^k_{\alpha\varepsilon x}\}$ *defined in* Lemma *1.1. If* $\alpha_k \to 0$ *and*

$$f(x_k, y^k_{\alpha\varepsilon x}) - \alpha_k\langle \nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k\rangle > \eta(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) \quad \forall k > 0, \tag{1.8}$$

*for any fixed $\eta \in (0,1)$, then $x^*$ solves (EP).*

*Proof.* The condition (1.8) could be restated as:

$$\frac{-\alpha_k\langle\nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k\rangle + \eta\varepsilon_k}{\eta - 1} < f(x_k, y^k_{\alpha\varepsilon x}) \leq \varepsilon_k \quad \forall k > 0$$

where the last inequality follows from Definition 1.13 taking into account that $\varphi_\alpha(x) \geq 0$ for any $x \in C$. Since $y^k_{\alpha\varepsilon x} \in C$ and $C$ is bounded we can take a subsequence $k \in L$ for some infinite $L \subseteq \mathbb{N}$ such that $y^k_{\alpha\varepsilon x} \to y^*_{\alpha\varepsilon x}$. Hence exploiting Lemma 1.1 we can assert that

$$\lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} -f(x_k, y^k_{\alpha\varepsilon x}) = \lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} \varphi_{\alpha_k}(x_k).$$

Thus taking the subsequential limit we obtain

$$\lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} \frac{-\alpha_k\langle\nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k\rangle + \eta\varepsilon_k}{\eta - 1} \leq \lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} f(x_k, y^k_{\alpha\varepsilon x}) \leq \lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} \varepsilon_k$$

$$\Leftrightarrow$$

$$\lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} \frac{-\alpha_k\langle\nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k\rangle + \eta\varepsilon_k}{\eta - 1} \leq \lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} -\varphi_{\alpha_k}(x_k) \leq \lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} \varepsilon_k.$$

Besides, Lemma 1.1 *b)* guarantees $\lim\limits_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} y^k_{\alpha\varepsilon x} = y^*_{\alpha\varepsilon x} \in C$. Thus, thanks to the continuity of $\nabla_x h(x,y)$ it holds that

$$\langle\nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k\rangle \to \langle\nabla_x h(x^*, y^*_{\alpha\varepsilon x}), y^*_{\alpha\varepsilon x} - x^*\rangle \in \mathbb{R}$$

and therefore $\lim\limits_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} \dfrac{-\alpha_k\langle\nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k\rangle + \eta\varepsilon_k}{\eta - 1} \to 0$. Putting this together with the latter inequality above we get:

$$0 \leq \lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} -\varphi_{\alpha_k}(x_k) \leq 0.$$

To complete the proof, we will show that $\varphi_{\alpha_k}(x_k) \to 0$ as $k \to \infty$ with $k \in L$ implies that $x^*$ solves (EP). Let's start exploiting the propriety of $-\varphi_{\alpha_k}(x_k)$ of being a minimum:

$$f(x_k, y) + \alpha_k h(x_k, y) \geq -\varphi_{\alpha_k}(x_k) \quad \forall y \in C.\ \forall k \in L,$$

taking the limit as $k \to \infty$ with $k \in L$, we obtain

$$f(x^*, y) \geq 0 \quad \forall y \in C,$$

i.e., $x^*$ solves (EP).

□

**Corollary 1.2**

*Consider $x^* \in C$ that doesn't solve (EP), and two positive sequences, $\{\alpha_k\}$, $\{\varepsilon_k\}$, such that $\alpha_k \to 0$, $\varepsilon_k \to 0$, then $\exists k^*$ such that $\forall k > k^*$*

$$f(x_k, y^k_{\alpha\varepsilon x}) - \alpha_k \langle \nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k \rangle \leq \eta(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k),$$

*holds for all $k \geq k^*$ with any fixed $\eta \in (0, 1)$.*

*Proof.* Ab absurdo, suppose

$$f(x_k, y^k_{\alpha\varepsilon x}) - \alpha_k \langle \nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k \rangle > \eta(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) \quad \forall k.$$

Applying Lemma 1.2 with $x_k := x^*$ for all $k$, we get that $x$ solves (EP). □

### 1.4.2 The Algorithm and its Correctness

The pseudo code of Algorithm 1.3 is given below.

**Algorithm 1.3:** Error Tolerant Algorithm

0. Choose $x_0 \in C$ and positive (decreasing) sequences $\{\sigma_k\} : \sigma_k \to 0$, $\{\delta_k\} : \delta_k \to 0$, $\sum_{k=0}^{\infty} \delta_k < \infty$ and $\{\varepsilon_k\} : \varepsilon_k < \delta_k$.
1. Set $\alpha_0 := \sigma_0$ , set $k := 0, i := 0$ choose constants $\gamma, \eta, \beta \in (0, 1), \eta > \beta$.
2. Compute $y^k_{\alpha\varepsilon x}$, $\varepsilon_k$ approximated solution of $(P^{\alpha_k}_{x_k})$.
3. Set $d_k := y^k_{\alpha\varepsilon x} - x_k$.
4. If

$$f(x_k, y^k_{\alpha\varepsilon x}) - \alpha_k \langle \nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k \rangle \leq \eta(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k),$$

5. Compute the smallest non negative integer $s$ such that

$$-f_{\alpha_k}(x_k + \gamma^s d_k, y_{\alpha_k \varepsilon_k (x_k + \gamma^s d_k)}) \leq -f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) + \beta\gamma^{2s}(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) + \delta_k,$$

set $t_k := \gamma^{2s}$ and $\alpha_{k+1} := \alpha_k$,
else
6. set $t_k := 0$, $i := i + 1$, $\alpha_{k+1} := \sigma_i$.
7. Set $x_{k+1} := x_k + t_k d_k$ and $k := k + 1$.
8. If $f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) \leq \varepsilon_k$ STOP else GOTO 2.

**Lemma 1.3**

*The line search procedure at Step 5 of Algorithm 1.3 is well defined, i.e., it terminates in a finite number of steps.*

*Proof.* Assume by contradiction that for all $s$

$$-f_{\alpha_k}(x_k+\gamma^s d_k, y_{\alpha_k\varepsilon_k(x_k+\gamma^s d_k)}) + f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) > \beta\gamma^{2s}(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) + \delta_k$$

holds. Since by DEFINITION 1.13 the LHS is less or equal to $\varphi_{\alpha_k}(x_k+\gamma^s d_k) - \varphi_{\alpha_k}(x_k) + \varepsilon_k$, it also holds

$$\varphi_{\alpha_k}(x_k+\gamma^s d_k) - \varphi_{\alpha_k}(x_k) + \varepsilon_k > \beta\gamma^{2s}(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) + \delta_k.$$

Taking $\varepsilon_k$ to the RHS, and dividing by $\gamma^s$ we obtain

$$\frac{\varphi_{\alpha_k}(x_k+\gamma^s d_k) - \varphi_{\alpha_k}(x_k)}{\gamma^s} > \beta\gamma^s(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) + \frac{\delta_k - \varepsilon_k}{\gamma^s}.$$

For sufficiently large $s$, we have

$$\frac{\delta_k - \varepsilon_k}{\gamma^s} \geq \langle \nabla_x \varphi_{\alpha_k}(x_k), d_k\rangle + 1,$$

Thus for those $s$ we have that

$$\frac{\varphi_{\alpha_k}(x_k+\gamma^s d_k) - \varphi_{\alpha_k}(x_k)}{\gamma^s} \geq \beta\gamma^s(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) + \langle \nabla_x \varphi_{\alpha_k}(x_k), d_k\rangle + 1.$$

Taking the $\lim_{s\to\infty}$ we get to the contradiction

$$\langle \nabla_x \varphi_{\alpha_k}(x_k), d_k\rangle \geq \langle \nabla_x \varphi_{\alpha_k}(x_k), d_k\rangle + 1.$$

□

**LEMMA 1.4**

*Let $f$ be c-monotone. Consider a subsequence indexed by $k_r$ (and his induced natural numbers subset $R$), such that $x_{k_r} \to x^*$ and $\alpha_{k_r} \to \sigma \neq 0$ and suppose that*

$$f(x_k, y^k_{\alpha\varepsilon x}) - \alpha_k\langle \nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k\rangle \leq \eta(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k)$$

*holds. Then*

$$\langle \nabla_x \varphi_\sigma(x^*), y_\sigma(x^*) - x^*\rangle \leq \eta f(x^*, y_\sigma(x^*))$$

*Proof.* For all $k \in R$ the following chain of inequalities holds:

$$\begin{aligned}
&\langle \nabla_x \varphi_{\alpha_k}(x_k), y_{\alpha_k}(x_k) - x_k \rangle \\
&\leq f(x_k, y_{\alpha_k}(x_k)) - \alpha_k \langle \nabla_x h(x_k, y_{\alpha_k}(x_k)), y_{\alpha_k}(x_k) - x_k \rangle \\
&\leq f(x_k, y^k_{\alpha\varepsilon x}) + \alpha_k (h(x_k, y^k_{\alpha\varepsilon x}) - h(x_k, y_{\alpha_k}(x_k))) - \alpha_k \langle \nabla_x h(x_k, y_{\alpha_k}(x_k)), y_{\alpha_k}(x_k) - x_k \rangle \\
&\leq \eta(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) + \alpha_k (h(x_k, y^k_{\alpha\varepsilon x}) - h(x_k, y_{\alpha_k}(x_k))) \\
&\quad + \alpha_k (\langle \nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k \rangle - \langle \nabla_x h(x_k, y_{\alpha_k}(x_k)), y_{\alpha_k}(x_k) - x_k \rangle)
\end{aligned} \tag{1.9}$$

Where the first inequality is Theorem 1.11, the second is implied by Definition 1.13 and the third is provided by the assumption. We conclude the proof taking the limit as $k \to \infty$ with $k \in R$ of (1.9). In fact, for Theorem 1.3 we have

$$y_{\alpha_k}(x_k) \to y_\sigma(x^*). \tag{1.10}$$

In addition, Lemma 1.1 guarantees $\lim_{\substack{k \to \infty \\ k \in R \subset \mathbb{N}}} \|y^k_{\alpha\varepsilon x} - y_{\alpha_k}(x_k)\| = 0$. Hence, $y^k_{\alpha\varepsilon x} \to y_\sigma(x^*)$ and we obtain

$$\begin{aligned}
&\alpha_k (h(x_k, y^k_{\alpha\varepsilon x}) - h(x_k, y_{\alpha_k}(x_k))) \\
&+ \alpha_k (\langle \nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k \rangle - \langle \nabla_x h(x_k, y_{\alpha_k}(x_k)), y_{\alpha_k}(x_k) - x_k \rangle) \to 0 \\
&\eta(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) \to \eta f(x^*, y_\sigma(x^*)).
\end{aligned}$$

□

**Lemma 1.5**

*Consider the sequence of $\{\alpha_k\}$ generated by the algorithm. If $\alpha_k \to \sigma \neq 0$, then*

$$\lim_{k \to \infty} |t_k^2 (f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k)| = 0$$

*Proof.* The condition at Step 4 can be not met at most a finite number of times (otherwise $\alpha_k \to 0$). Therefore, there must exist $\overline{k}$ such that for all $k \geq \overline{k}$ $\alpha_k = \sigma$. The definition of $t_k$ (Step 5) guarantees

$$f_{\alpha_k}(x_k + t_k d_k, y_{\alpha_k \varepsilon_k (x_k + t_k d_k)}) \leq -f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) + \beta t_k^2 (f(x_k, y^k_{\alpha\varepsilon x}) - \epsilon_k) + \delta_k \quad \forall k \geq \overline{k}.$$

Thus, the definition of $\varepsilon$-approximation leads to the following equation:

$$\begin{aligned}
\varphi_{\alpha_k}(x_{k+1}) - \varphi_{\alpha_k}(x_k) - \varepsilon_k \leq & - f_{\alpha_k}(x_k + t_k d_k, y_{\alpha_k \varepsilon_k (x_k + t_k d_k)}) + f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) \\
\leq & \beta t_k^2 (f(x_k, y^k_{\alpha\varepsilon x}) - \epsilon_k) + \delta_k \quad \forall k \geq \overline{k}.
\end{aligned}$$

Summing up these inequalities from $\overline{k}$ to $k$ we get

$$\sum_{i=\overline{k}}^{k} \varphi_\sigma(x_{i+1}) - \varphi_\sigma(x_i) - \varepsilon_i \leq \sum_{i=\overline{k}}^{k} \beta t_i^2 (f(x_i, y^i_{\alpha\varepsilon x}) - \epsilon_i) + \delta_i.$$

Since

$$\sum_{i=\overline{k}}^{k} \varphi_\sigma(x_{i+1}) - \varphi_\sigma(x_i) = \varphi_\sigma(x_k) - \varphi_\sigma(x_{\overline{k}}), \tag{1.11}$$

taking $\delta_i$ to the LHS and multiplying by $-1$ we obtain

$$-\varphi_\sigma(x_k) + \varphi_\sigma(x_{\overline{k}}) + \sum_{i=\overline{k}}^{k} (\varepsilon_i + \delta_i) \geq \sum_{i=\overline{k}}^{k} -\beta t_i^2 (f(x_i, y^i_{\alpha\varepsilon x}) - \epsilon_i).$$

Since $\varepsilon_i + \delta_i \leq 2\delta_i$, $\varphi_\sigma(x_k) \leq 0$ and (by definition of the approximation) $f(x_i, y^i_{\alpha\varepsilon x}) - \epsilon_i \leq 0$, the above inequality implies that

$$\varphi_\sigma(x_{\overline{k}}) + 2\sum_{i=\overline{k}}^{k} \delta_i \geq \sum_{i=\overline{k}}^{k} \beta |t_i^2 (f(x_i, y^i_{\alpha\varepsilon x}) - \epsilon_i)|.$$

By assumption, the series $\sum_{i=\overline{k}}^{\infty} \delta_i$ is convergent, hence also the series $\sum_{i=\overline{k}}^{\infty} \beta |t_i^2 (f(x_i, y^i_{\alpha\varepsilon x}) - \epsilon_i)$ is convergent, which implies $\lim_{k\to\infty} \beta |t_k^2 (f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k)| = 0$. □

**Lemma 1.6**

*Let $f$ be c-monotone. If $\alpha_k \to \sigma \neq 0$ and $x^*$ is a cluster point of $\{x_k\}$, then $x^*$ solves (EP).*

*Proof.* Without loss of generality, suppose that the subsequence $x_{k_r}$ satisfy $x_{k_r} \to x^*$. Lemma 1.5 guarantees

$$\lim_{r\to\infty} \beta |t_{k_r}^2 (f(x_{k_r}, y^{k_r}_{\alpha\varepsilon x}) - \varepsilon_{k_r})| = 0.$$

Two cases may occour, $f(x_{k_r}, y^{k_r}_{\alpha\varepsilon x}) - \varepsilon_{k_r} \to 0$ or $t_{k_r} \to 0$.

1. If $f(x_{k_r}, y^{k_r}_{\alpha\varepsilon x}) - \varepsilon_{k_r} \to 0$ then from Definition 1.13 we have

$$\lim_{r\to\infty} f_{\alpha_{k_r}}(x_{k_r}, y^{k_r}_{\alpha\varepsilon x}) \geq \lim_{r\to\infty} -\varphi_{\alpha_{k_r}}(x_{k_r}) \geq \lim_{r\to\infty} f_{\alpha_{k_r}}(x_{k_r}, y^{k_r}_{\alpha\varepsilon x}) - \varepsilon_{k_r}$$

Therefore $\varepsilon_{k_r} \to 0$ and $f(x_{k_r}, y^{k_r}_{\alpha\varepsilon x}) \to 0$ guarantee

$$\lim_{r\to\infty} -\varphi_{\alpha_{k_r}}(x_{k_r}) = \lim_{r\to\infty} f_{\alpha_{k_r}}(x_{k_r}, y^{k_r}_{\alpha\varepsilon x}) \geq \lim_{r\to\infty} f(x_{k_r}, y^{k_r}_{\alpha\varepsilon x}) = 0,$$

and thus, $\lim_{r\to\infty} -\varphi_\sigma(x_{k_r}) = 0$. Thanks to the non negativity of $\varphi_\sigma$. Since we have found a pair $(x^*, y_\sigma(x^*))$ that minimizes the gap function, THEOREM 1.1 guarantees that $x^*$ solves (EP).

2. Now suppose that $t_{k_r} \to 0$. Let $R$ be the infinite subset of $\mathbb{N}$ induced by $\{k_r\}$. The line search at Step 5,ensures that

$$-f_{\alpha_k}(x_k + \frac{t_k}{\gamma}d_k, y_{\alpha_k\varepsilon_k(x_k+\frac{t_k}{\gamma}d_k)}) > -f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) + \beta(\frac{t_k}{\gamma})^2(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) + \delta_k,$$

is valid for all $k \in R$. Furthermore, (DEFINITION 1.13) guarantees

$$\begin{cases} \varphi_{\alpha_k}(x_k + \frac{t_k}{\gamma}d_k) \geq -f_{\alpha_k}(x_k + \frac{t_k}{\gamma}d_k, y_{\alpha_k\varepsilon_k(x_k+\frac{t_k}{\gamma}d_k)}) \\ -f_{\alpha_k}(x_k, y^k_{\alpha\varepsilon x}) \geq \varphi_{\alpha_k}(x_k) - \varepsilon_k \end{cases}$$

for all $k \in R$. Hence, for all $k \in R$, it holds

$$\varphi_{\alpha_k}(x_k+\frac{t_k}{\gamma}d_k)-\varphi_{\alpha_k}(x_k) > \beta(\frac{t_k}{\gamma})^2(f(x_k, y^k_{\alpha\varepsilon x})-\varepsilon_k)+\delta_k-\varepsilon_k \geq +\beta(\frac{t_k}{\gamma})^2(f(x_k, y^k_{\alpha\varepsilon x})-\varepsilon_k).$$

By the mean value theorem there exists $\theta_k \in (0,1)$ such that

$$\langle \nabla\varphi_{\alpha_k}(x_k + \theta_k\frac{t_k}{\gamma}d_k), d_k\rangle \geq \beta\frac{t_k}{\gamma}(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) \quad \forall k \in R.$$

Taking the limit as $k \to \infty$ we obtain

$$\langle \nabla\varphi_\sigma(x^*), d^*\rangle \geq 0.$$

Where $d^* := y_\sigma(x^*) - x^*$. Besides, LEMMA 1.4 guarantees

$$\langle \nabla\varphi_\sigma(x^*), d^*\rangle \leq \eta f(x^*, y_\sigma(x^*)).$$

Putting the two inequalities together, we get

$$\eta f(x^*, y_\sigma(x^*)) \geq 0,$$

that implies that $-\varphi_\sigma(x^*) \geq 0$. Therefore, as in the previous case, $x^*$ solves (EP).

□

**THEOREM 1.13** (Correctness of ALGORITHM 1.3)
*Let $f$ be c-monotone. Let $x^*$ be a cluster point of the sequence $\{x_k\}$ generated by ALGORITHM 1.3, then $x^*$ solves (EP).*

*Proof.* Firstly notice that, the existence of a cluster point is guaranteed by the compactness of $C$. Let $\{k_r\}$ be a subsequence such that $x_{k_r} \to x^*$. We distinguish two cases: $\alpha_k \to 0$, and $\alpha_k \to \sigma \neq 0$ (indeed, there are no other possibility for $\{\alpha_k\}$).

1. Suppose $\alpha_k \to 0$. Then, we can choose an appropriate subsequence $S$ of $\{k_r\}$ such that

$$f(x_k, y^k_{\alpha\varepsilon x}) - \alpha_k \langle \nabla_x h(x_k, y^k_{\alpha\varepsilon x}), y^k_{\alpha\varepsilon x} - x_k \rangle > \eta(f(x_k, y^k_{\alpha\varepsilon x}) - \varepsilon_k) \quad \forall k_r \in S,$$

   This could be for example the subsequence of $\alpha_{k_r}$ obtained restricting to different values of $\alpha_{k_r}$. Now we are in condition to apply Lemma 1.2 and prove that $x^*$ solves (EP).

2. Suppose that $\alpha_k \to \sigma \neq 0$, then the thesis follows from Lemma 1.6.

□

# Chapter 2

# Handling Nonlinear Constraints

This Chapter is devoted to the particular case when the set $C$ is described not only by linear constraints but also by nonlinear inequalities. Since solving the auxiliary problem ($P_x^\alpha$) could become a cumbersome task in presence of non linear inequalities, approximation techniques have been adopted in literature to contain the algorithmic cost.

Firstly, we show a descent method that approximates the nonlinear constraints with their first order Taylor approximation in order to solve more efficiently the auxiliary problem. Then, we develop an error tolerant version of the method and, in the next chapter, discuss possible ways to compute solution of the auxiliary problem.

## 2.1 Nonlinear Constraints Approximation Algorithm

Throughout this Chapter we will make further assumptions on $C$.

**Assumption 2.1**
*We suppose $C$ to be the intersection of a bounded polyhedron $D$ and a convex set given through convex inequalities, namely $C = D \cap \tilde{C}$ with*

$$D = \{y \in \mathbb{R}^n : \ \langle a_j, v\rangle \leq b_j, \quad j = 1, \dots, r_1, \ \langle a_j, v\rangle = b_j, \ j = r+1, \dots, r\}$$
*for some $a_j \in \mathbb{R}^n$ and $b_j \in \mathbb{R}$*
$$\tilde{C} = \{y \in \mathbb{R}^n : c_i(y) \leq 0 \ i = 1, \dots, m\}$$
*where $c_i : \mathbb{R}^n \to \mathbb{R}$ are twice continuously differentiable (nonlinear) convex functions.*

*Furthermore we assume that the vectors $a_j$ with $j = 1, \dots, r_1$ are linearly independent and that there exists $\hat{y} \in D$ such that $\langle a_j, \hat{y}\rangle < b_j$ for $j = 1, \dots, r_1$ and $c_i(\hat{y}) < 0$ for $i = 1, \dots, m$.*

The method we describe has been proposed by Bigi and Passacantando ([BP12]) and is built upon a new gap function $\psi : \mathbb{R}^n \to \mathbb{R}$. While computing the gap function $\varphi$ at a point $x$ implicitly involves an Optimization Problem over the set $C$ defined in Assumption 2.1, computation of $\psi(x)$ involves an Optimization Problem over the polyhedron $P(x)$, defined as follows:

**Definition 2.1** (Polyhedron $P(x)$)
Given (EP) and a point $x \in \mathbb{R}^n$ we define the polyhedron $P(x)$ as:

$$P(x) = \{y \in D : c_i(x) + \langle \nabla_x c_i(x), y - x \rangle \leq 0, \quad i = 1, 2, \ldots, m\},$$

where $c_i$ and $D$ are the ones defined in Assumption 2.1.

As a consequence of the simpler structure of the feasible set, the computation of $\psi(x)$ should be less expensive than computing $\varphi(x)$.

Before step deeply into the method, we want to point out some properties of $P(x)$ that will be used extensively throughout this Chapter.

**Proposition 2.1** ([BP12])
*Given $x \in \mathbb{R}^n$, it holds*

*a)* $C \subseteq P(x) \subseteq D$;

*b)* $x \in C \Leftrightarrow x \in P(x)$.

*Proof.* a) By Definition 2.1 we have that $P(x) \subseteq D$. On the other hand the convexity of $c_i$ guarantees

$$c_i(x) \geq c_i(x_0) + \langle \nabla c_i(x_0), x_0 - x \rangle, \tag{2.1}$$

for any $x_0, x \in \mathbb{R}^n$. thus, if $x \in C$, we have that $0 \geq c_i(x)$ and hence $x \in P(x)$.

b) Consider a point $x \in D$ by Definition 2.1 we have that $x \in P(x)$ iff $c_i(x) \leq 0$ for any $i$.

□

We still not have given a clear definition of the function $\psi$. Formally, given $x \in D$ we define the problem $\mathcal{P}_x$ as

$$\min_{y \in P(x)} f(x, y), \tag{$\mathcal{P}_x$}$$

and define the function $\psi : D \to \mathbb{R}$ as the changed sign optimal value of $\mathcal{P}_x$ for given $x$.

In order to handle a differentiable function we can exploit the auxiliary problem principle defining the problem $(\mathcal{P}_x^\alpha)$ and the function $\psi_\alpha$. Notice that in this way we gain also the strictly convexity of the objective function.

**Definition 2.2**
Given $\alpha > 0$, $x \in C$ we define the optimization problem $(\mathcal{P}_x^\alpha)$ as

$$\text{find } \nu_\alpha(x) = \arg\min_{y \in P(x)} f_\alpha(x, y) \tag{$\mathcal{P}_x^\alpha$}$$

Notice that since $f_\alpha$ is strictly convex the solution of $(\mathcal{P}_x^\alpha)$ is unique.

**Definition 2.3**
Given $\alpha > 0$, we define the functions $\psi_\alpha : \mathbb{R}^n \to \mathbb{R}$ and $\nu_\alpha : \mathbb{R}^n \to \mathbb{R}^n$ to be respectively the changed sign optimum value and the minimizer of $(\mathcal{P}_x^\alpha)$, i.e.

$$\psi_\alpha(x) = -\min_{y \in P(x)} f_\alpha(x, y),$$
$$\nu_\alpha(x) = \arg\min_{y \in P(x)} f_\alpha(x, y).$$

Functions $\psi_\alpha$ and $\nu_\alpha$ are clearly the counterparts of $\varphi_\alpha$ and $y_\alpha$.

We will see in few pages that we can deal with $\psi_\alpha$ or $\nu_\alpha$ as we did previously with the gap function $\varphi_\alpha$ or $y_\alpha$, i.e. they allow us to reformulate (EP) as an Optimization Problem.

Under Assumption 2.1, the map $x \mapsto \nu_\alpha(x)$ is single valued (remember that $f_\alpha(x, \cdot)$ is strictly convex and thus the solution of $\min\{f_\alpha(x, y) : y \in P(x)\}$ is unique), furthermore it allows a fixed point reformulation of (EP) as stated by the following theorem:

**Lemma 2.1** ([BP12])
*Given any $\alpha > 0$, $x^*$ solves (EP) iff $\nu_\alpha(x^*) = x^*$.*

*Proof.* Suppose that $x^*$ solves (EP), then for Theorem 1.2 it also solves ($\alpha$-EP).

and therefore thanks to Theorem 1.1 it minimizes $f_\alpha(x^*, \cdot)$ over $C$.

Hence, there exist Lagrange multiplier vectors $\lambda^* \in \mathbb{R}_+^m$, and $\mu^* \in \mathbb{R}^r$ such that $\mu_1^*, \mu_2^*, \ldots, \mu_{r_1}^* \geq 0$ and

$$\begin{cases} \nabla_y f_\alpha(x^*, x^*) + \sum_{i=0}^m \lambda_i^* \nabla c_i(x^*) + \sum_{j=0}^r \mu_j^* a_j = 0 \\ \lambda_i^* c_i(x^*) = 0 \quad i = 1, 2, \ldots, m \\ \mu_j^*(\langle a_j, x^* \rangle - b_j) = 0 \quad j = 1, 2, \ldots, r \\ c_i(x^*) \leq 0 \quad i = 1, 2, \ldots, m \\ \langle a_j, x^* \rangle - b_j \leq 0 \quad j = 1, 2, \ldots, r_1 \\ \langle a_j, x^* \rangle - b_j = 0 \quad j = r_1 + 1, 2, \ldots, r. \end{cases}$$

Defining $g_i(y) = c_i(x^*) + \langle \nabla c_i(x^*), y - x \rangle$, then we have $g_i(x^*) = c_i(x^*)$ and we can rewrite the above conditions in the following way:

$$\begin{cases} \nabla_y f_\alpha(x^*, x^*) + \sum_{i=0}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=0}^{r} \mu_j^* a_j = 0 \\ \lambda_i^* g_i(x^*) = 0 \quad i = 1, 2, \dots, m \\ \mu_j^* (\langle a_j, x^* \rangle - b_j) = 0 \quad j = 1, 2, \dots, r \\ g_i(x^*) \leq 0 \quad i = 1, 2, \dots, m \\ \langle a_j, x^* \rangle - b_j \leq 0 \quad j = 1, 2, \dots, r_1 \\ \langle a_j, x^* \rangle - b_j = 0 \quad j = r_1 + 1, 2, \dots, r, \end{cases}$$

which are the Karush-Kuhn-Tucker conditions for the problem of minimizing $f_\alpha(x^*, \cdot)$ over $P(x^*)$. Since this is a strictly convex problem and both $x^*$ and $\nu_\alpha(x^*)$ solve it we have that $\nu_\alpha(x^*) = x^*$.

Now suppose that $\nu_\alpha(x^*) = x^*$, therefore $x^* \in C$ because $x^* = \nu_\alpha(x^*) \in P(x^*)$. Since $\nu_\alpha(x^*)$ minimizes $f_\alpha(x^*, \cdot)$ over $C$, the necessary and sufficient conditions read

$$\langle \nabla_y f(x^*, x^*), z - x^* \rangle \geq 0 \quad \forall z \in P(x^*),$$

taking into account that $\nabla_y h(x, x) = 0$ for any $x \in C$. Since $C \subseteq P(x^*)$, we also have

$$\langle \nabla_y f(x^*, x^*), z - x^* \rangle \geq 0 \quad \forall z \in C.$$

Hence, $x^*$ is a minimizer of $f(x^*, \cdot)$ over $C$ and $f(x^*, x^*) = 0$. Thus, $x^*$ solves (EP). □

**Lemma 2.2** ([BP12])
*Given $\alpha > 0$, the function $\psi_\alpha$ is a gap function for ($\alpha$ EP), i.e.,*

*(a) $\psi_\alpha(x) \geq 0$ for any $x \in C$,*

*(b) $x^*$ solves (EP) iff $\psi_\alpha(x^*) = 0$ and $x^* \in C$.*

*Proof.* (a) If $x \in C$, then optimality of $\nu_\alpha(x)$ guarantees

$$-\psi_\alpha(x) = f_\alpha(x, \nu_\alpha(x)) \leq f_\alpha(x, x) = 0.$$

(b) If $x^*$ solves (EP), then $x^* \in C$ and by Lemma 2.1 $\nu_\alpha(x^*) = x^*$. Thus $\psi_\alpha(x^*) = 0$. On the other hand, suppose that $\psi_\alpha(x^*) = f_\alpha(x^*, x^*) = 0$ and $x^* \in C$. Since $f_\alpha(x, \cdot)$ is strictly convex $\nu_\alpha(x^*) = x^*$ is the only minimizer of that function. Therefore we can apply again Lemma 2.1 to conclude the proof. □

**Lemma 2.3** ([BP12])
*For any $\alpha > 0$, the map $\nu_\alpha$ is continuous on $\mathbb{R}^n$.*

**Lemma 2.4** ([BP12])
*For any $\alpha > 0$, $\psi_\alpha$ is locally Lipschitz continuous on $\mathbb{R}^n$.*

**Definition 2.4** (Clarke Generalized Directional Derivative [Cla87])
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function, then we define the Clarke generalized directional derivative in direction $d \in \mathbb{R}^n$ at point $x \in \mathbb{R}^n$as:

$$f^\circ(x; d) = \limsup_{(z,t)\to(x,0)} \frac{f(z + td) - f(z)}{t}$$

In a similar way we can define the generalized gradient.

**Definition 2.5** (Generalized Gradient [Cla87])
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function, then given $x \in \mathbb{R}^n$, we define the *generalized gradient set* $\partial^\circ f(x)$ of $f$ at $x$ as

$$\partial^\circ f(x) = \{\xi \in \mathbb{R}^n \mid f^\circ(x; v) \geq \langle \xi, v\rangle \quad v \in \mathbb{R}^n\}.$$

an element of $\partial^\circ f(x)$ is called *generalized gradient* of $f$ at $x$.

In our case the generalized directional derivative plays the role of the directional derivative in order to validate a descent direction: if it is negative, we have found a descent direction. The main reason to exploit the generalized directional derivative is the exploitation of the mean value theorem:

$$f(x + d) - f(x) = \langle \xi, d\rangle$$

where $\xi$ is a generalized gradient of $f$ at a point in the line between $x$ and $d$. Notice that this property is not guaranteed by the directional derivative in the case of non differentiable functions.

The following theorem (which is the counterpart of Theorem 1.11) is fundamental: it provides an upper bound on the generalized directional derivative $\psi^\circ_\alpha(x; \nu_\alpha(x) - x)$, thus allowing to avoiding its direct computation.

**Theorem 2.1** ([BP12])
*Given $\alpha > 0$, the inequality*

$$\psi^\circ_\alpha(x, \nu_\alpha(x) - x) \leq -\langle \nabla_x f_\alpha(x, \nu_\alpha(x)), \nu_\alpha(x) - x\rangle,$$

*holds for any $x \in D$.*

When we substitute $C$ with $P(x)$, implicitely we loose the guarantee to remain inside $C$ while moving along the direction $\nu_\alpha(x) - x$. In fact, $\nu_\alpha(x)$ could belong to $D \setminus C$ while

exploiting the direction $y_\alpha(x) - x$ with proper stepsize, we were sure to never get out of $C$ (assuming $x \in C$).

This issue have been tackled in [BP12] exploiting penalization techniques: instead of minimizing the gap function $\psi_\alpha$, they minimize the function defined as follows:

**Definition 2.6**
Given $\alpha, \varrho > 0$, we define the function $\Psi_\alpha^\varrho : \mathbb{R}^n \to \mathbb{R}$

$$\Psi_\alpha^\varrho(x) = \psi_\alpha(x) + \frac{1}{\varrho}\|c^+(x)\|, \tag{2.2}$$

and $c^+(x) = (c_1^+(x), c_2^+(x), \ldots, c_m^+(x))$ with

$$c_i^+(x) = \begin{cases} 0 & \text{if } c_i(x) < 0 \\ c_i(x) & \text{otherwise.} \end{cases} \tag{2.3}$$

It's immediate that for all the points outside $C$, the additive term $\frac{1}{\varrho}\|c^+(x)\|$ act as a penalization increasing the value of the objective function, instead for all points belonging to $C$, $\Psi_\alpha^\varrho$ is equal to $\psi_\alpha$. $\varrho$ is employed as the penalization's tuner: reducing $\varrho$ the penalization increases and viceversa increasing it the penalization decreses.

Exploiting penalization, we obtain a gap function over the whole set $D$ as stated by the following theorem:

**Theorem 2.2** ([BP12])
*Given $\overline{\alpha} > 0$, there exist $\overline{\varrho} > 0$ such that*

*a) $\Psi_\alpha^\varrho(x) \geq 0$ for any $x \in D$,*

*b) $x^*$ solves (EP) iff $\Psi_\alpha^\varrho(x^*) = 0$,*

*where $\alpha \in [0, \overline{\alpha}]$ and $\varrho \in (0, \overline{\varrho})$.*

The next theorem involves some inequality that will be very useful in the designing of a descent method.

**Lemma 2.5** (Lemma 4 and Theorem 4 in [BP12])
*Suppose that $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is c-monotone, and let $\Lambda_\alpha(x)$ be the set of Lagrangian multipliers associated to $\nu_\alpha(x)$, then it holds that:*

*(i)* $\Psi_\varrho^{\alpha\circ}(x; \nu_\alpha(x) - x) \leq -\Psi_\varrho^\alpha(x) - \alpha[h(x, \nu_\alpha(x)) + \langle \nabla_x h(x, \nu_\alpha(x)), \nu_\alpha(x) - x\rangle]$,

*(ii)* *If* $x \in D \setminus C$ *and* $(\lambda, \mu) \in \Lambda_\alpha(x)$, *then* $\Psi_\varrho^{\alpha\circ}(x; \nu_\alpha(x) - x) < 0$ *for any* $\varrho$ *such that* $\frac{1}{\varrho} > \|(\lambda^+)\|$, *where*

$$\lambda_i^+ = \begin{cases} \lambda_i & if\ c_i(x) > 0 \\ 0 & else. \end{cases}$$

*(iii)* *If* $x \in C$ *does not solve (EP) and* $\eta \in (0,1)$, *then*

$$\Psi_\alpha^\varrho(x) - \alpha_k(h(x, \nu_\alpha(x)) + \langle \nabla_x h(x, \nu_\alpha), \nu_\alpha(x) - x\rangle) \leq -\eta\Psi_\alpha(x)$$

*holds for any* $\varepsilon > 0$, *and any sufficiently small* $\alpha > 0$.

The first statement is the counterpart of THEOREM 2.1 and provides a way to check whether or not the direction $\nu_\alpha(x) - x$ is a descent one. The second and third statement suggests a condition on the penalization parameter $\varrho$ and the Lagrangian multipliers that will be used to ensure the correctness of the algorithm.

Now we are ready to give the algorithm pseudocode (see the listing ALGORITHM 2.1) and state its correcteness.

Algorithm 2.1: Nonlinear Constraints Approximation Algorithm

0. Fix $\eta, \gamma, \delta \in (0,1)$, $\beta \in (0, \eta)$ and positive sequence $\alpha_k, \varrho_k \downarrow 0$, choose $x_0 \in D$, and set $k = 1$.
1. Set $z^0 = x_{k-1}$ and $j = 0$.
2. Compute $\nu_j = \arg\min_{y \in P(z_j)} f_{\alpha_k}(z_j, y)$ and $\lambda_j \in \mathbb{R}^m$ any Lagrange multiplier vector corresponding to the linearized constraints.
3. Set $d_j = \nu_j - z_j$, if $d_j = 0$ STOP.
4. If the following relations hold:
   a) $\Psi_{\alpha_k}^{\varrho_k}(z_j) > 0$
   b) $\frac{1}{\varrho_k} \geq \|\lambda_k^+\| + \delta$
   c) $\Psi_{\alpha_k}^{\varrho_k}(z_j) - \alpha_k(h(z_j, \nu_j) + \langle \nabla_x h(z_j, \nu_j), \nu_j - z_j\rangle) < -\eta\Psi_{\alpha_k}^{\varrho_k}(z_j)$

   then compute the smallest non negative integer $s$ such that

   $$\Psi_{\alpha_k}^{\varrho_k}(z_j + \gamma^s d_j) - \Psi_{\alpha_k}^{\varrho_k}(z_j) \leq -\beta\gamma^{2s}\|d_j\|,$$

   set $t_j = \gamma^s$, $z_{j+1} = z_j + t_j d_j$, $j = j + 1$ and GOTO Step 2, else set $x_k = z_j$, $k = k + 1$ and GOTO Step 1.

**THEOREM 2.3** (Correctness of ALGORITHM 2.1 [BP12])
*If* $f$ *is c-monotone, then either the algorithm stops at a solution of (EP) after a finite number of iterations or it produces either an infinite sequence* $\{x_k\}$ *or an infinite sequence* $\{z_j\}$ *such that any of its cluster points solves (EP).*

## 2.2 Error Tolerant Extension

As we did before with Algorithm 1.2, now we are going to develop an Error Tolerant Version of Algorithm 2.1. Our work is based on the following definitions:

**Definition 2.7** ($\varepsilon$ approximated solution of $(\mathcal{P}_x^\alpha)$)
Given $\varepsilon, \alpha > 0$ and $x \in D$ a point $y^* \in \mathbb{R}^n$ it is called $\varepsilon$ approximated solution of $(\mathcal{P}_x^\alpha)$ if satisfies

i) $y^* \in P(x)$ and

ii) $f_\alpha(x, y^*) - \varepsilon \leq f_\alpha(x, \nu_\alpha(x)) \equiv \psi_\alpha(x)$.

**Definition 2.8** ($\Lambda_\alpha^\varepsilon(x)$)
Given $\alpha > 0$, we define the point to set map $\Lambda_\alpha : \text{int } \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}_+^m$ as a point to set map satisfying the following properties:

a) given $\varepsilon > 0$ and $x \in P(x)$, $\Lambda_\alpha^\varepsilon(x)$ is bounded,

b) for any positive sequence $\{\varepsilon_k\}$ and $x_k \in D$ satisfying $\varepsilon_k \to 0$ and $x_k \to x^*$ and for any sequence $\{\lambda_k\}$ such that $\lambda_k \in \Lambda_{\alpha_k}^{\varepsilon_k}(x_k)$ there exists a subsequence induced by $S \subset \mathbb{N}$ such that

$$\lim_{k \in S} \lambda_k \in \Lambda_\alpha(x^*),$$

where $\Lambda_\alpha(x) \subset \mathbb{R}_+^m$ is the set of Lagrange multipliers vectors corresponding to the nonlinear constraints for the problem $(\mathcal{P}_x^\alpha)$.

Notice that differentely from Algorithm 1.3, now we need both approximated primal (Definition 2.7) and dual solutions (2.8).

In addition, consider the following definition.

**Definition 2.9** ($\Gamma_\alpha^\varrho$)
Given $\alpha, \varrho > 0$, we define the function $\Gamma_\alpha^\varrho : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ as

$$\Gamma_\alpha^\varrho(x, \nu) = -f_\alpha(x, \nu) + \frac{1}{\varrho}\|c(x)^+\|$$

Notice that from this definition it follows that $\Psi_\alpha^\varrho(x) = \Gamma_\alpha^\varrho(x, \nu_\alpha(x))$ for any $x$.

In the remaining part of this chapter we give the algorithm pseudocode and we prove its correctness, afterwards in the next Chapter we propose some methods to compute suitable approximated primal and dual solutions of $(\mathcal{P}_x^\alpha)$ for a given error parameter $\varepsilon$.

**Algorithm 2.2:** Error Tolerant Algorithm Nonlinear Constraints

0. Choose $x_0 \in D$, and positive sequences $\{a_j\},\{\epsilon_j\},\{\rho_j\},\{\delta_j\}$ converging to zero such that $\sum_{j=1}^{\infty}\delta_j < +\infty$ and $\epsilon_j < \delta_j$. Set all indexes $I_\varrho, I_\alpha, I_\varepsilon$ to 0 and $k = 0$.
1. Increase $I_\varepsilon$.
2. Compute $\nu$, $\epsilon_{I_\varepsilon}$ approximated solution of $(\mathcal{P}_{x_k}^{a_{I_\alpha}})$ and $\lambda$, a vector in $\Lambda_{a_{I_\alpha}}^{\epsilon_{I_\varepsilon}}(x_k)$.
3. If $\Gamma_{a_{I_\alpha}}^{\rho_{I_\varrho}}(x_k,\nu) < 0$ $(\neg C1)$ or

$$-\eta(\Gamma_{a_{I_\alpha}}^{\rho_{I_\varrho}}(x_k,\nu)+\epsilon_{I_\varepsilon}) < -\Gamma_{a_{I_\alpha}}^{\rho_{I_\varrho}}(x_k,\nu) - a_{I_\alpha}[h(x_k,\nu)+\langle\nabla_x h(x_k,\nu),\nu-x_k\rangle] \quad (\neg C2)$$

   increase $I_a$ and $I_\rho$ and GOTO Step 1.
4. If $\frac{1}{\rho_{I_\varrho}} \leq \|\lambda^+\|$ $(\neg C3)$ increase $I_{\rho_{I_\varrho}}$ and GOTO Step 1.
5. Set $d_k = \nu - x_k$.
6. Set $k = k+1$ and $\alpha_k = a_{I_\alpha}$, $\varrho_k = \rho_{I_\varrho}$, $\varepsilon_k = \epsilon_{I_\varepsilon}$.
7. If $d_{k-1} = 0$ set $x_k = x_{k-1}$, $t_{k-1} = 0$ and GOTO Step 1.
8. Find smallest $s \in \mathbb{N}$ such that

$$\Gamma_{\alpha_k}^{\varrho_k}(x_{k-1}+\gamma^s d_{k-1},\nu_s) - \Gamma_{\alpha_k}^{\varrho_k}(x_{k-1},\nu) \leq -\beta\gamma^{2s}\|d_{k-1}\| + \delta_{k-1}$$

where $\nu_s$ is an $\varepsilon_k$ approximated solution of $(\mathcal{P}_{x_{k-1}+\gamma^s d_{k-1}}^{\alpha_k})$.
9. Set $t_{k-1} = \gamma^s$, $x_k = x_{k-1} + t_{k-1}d_{k-1}$ and GOTO Step 1.

### 2.2.1 Algorithm and Proof of Correctness

The pseudocode of Algorithm 2.2 is given in the listing below.

The first Lemma is devoted to prove that the line search at Step 8 eventually terminates.

**Lemma 2.6**
*Given $x, d \in D$, and $\alpha, \varrho, \varepsilon, \delta > 0$, $\gamma \in (0,1)$, let $\nu_s \in P(x)$ be an $\varepsilon$ approximated solution of $(\mathcal{P}_{x+\gamma^s d}^{\alpha})$, and $\nu_0$ an $\varepsilon$ approximated solution of $(\mathcal{P}_x^{\alpha})$. Suppose that $\delta > \varepsilon$, then there must exist $s \in \mathbb{N}$ such that*

$$\Gamma_\alpha^\varrho(x+\gamma^s d,\nu_s) - \Gamma_\alpha^\varrho(x,\nu_0) \leq -\beta\gamma^{2s}\|d\| + \delta \tag{2.4}$$

*Proof.* Definition 2.7 guarantees

$$\Gamma_\alpha^\varrho(x,\nu_\alpha(x)) \leq \Gamma_\alpha^\varrho(x,\nu_0) + \varepsilon$$
$$\Gamma_\alpha^\varrho(x+\gamma^s d,\nu_\alpha(x+\gamma^s d)) \geq \Gamma_\alpha^\varrho(x,\nu_s).$$

Therefore, supposing that (2.4) hold, for any $s > 0$, we get

$$\Gamma_\alpha^\varrho(x+\gamma^s d,\nu_\alpha(x+\gamma^s d)) - \Gamma_\alpha^\varrho(x,\nu_\alpha(x)) + \varepsilon > -\beta\gamma^{2s}\|d\| + \delta \tag{2.5}$$

Since the generalized Clarke derivative in the direction $d$: $\Gamma_\alpha^{\varrho\circ}(x,\nu_\alpha(x);d) = \Psi_\alpha^{\varrho\circ}(x;d)$ is finite and $\delta-\varepsilon>0$, there must exist $\overline{s}$ such that $\frac{\delta-\varepsilon}{\gamma^s} \geq \Gamma_\alpha^{\varrho\circ}(x,\nu_\alpha(x);d)+1$ holds for any $s>\overline{s}$. Thus (2.5) produces the following contradiction:

$$\limsup_{s\to\infty} \frac{\Gamma_\alpha^\varrho(x+\gamma^s d,\nu_\alpha(x+\gamma^s d)) - \Gamma_\alpha^\varrho(x,\nu_\alpha(x))}{\gamma^s} \geq \limsup_{s\to\infty} -\beta\gamma^s\|d\| + \Gamma_\alpha^{\varrho\circ}(x,\nu_\alpha(x);d)+1,$$

i.e. $0\geq 1$. □

Looking at the proof, we understand why we need to control the entity of the approximation $\varepsilon$. Without any guarantee on the approximation, we cannot set a proper $\delta$ to ensure that this kind of line search terminates. A good reason to use a line search like this, is that without assuming that the direction is a descent one, it allows us to prove the next Lemma, which is very important for the correctness of the method.

**Lemma 2.7**

*Let $\{x_k\}$ be a sequences in $D$, let $\{\delta_k\},\{\varepsilon_k\},\{t_k\}$ be positive numbers sequences, such that $\sum_{i=1}^\infty \delta_k<\infty$ and $\delta_k>\varepsilon_k$, and let $\alpha,\varrho>0$, and let $\nu_k^+,\nu_k^-\in P(x)$ be respectively an $\varepsilon_k$ approximated and an $\varepsilon_{k+1}$ approximated solution of $(\mathcal{P}_{x_k}^\alpha)$. In addition suppose that the following inequalities:*

$$\begin{aligned}&\Gamma_\alpha^\varrho(x_k,\nu_k^+) - \Gamma_\alpha^\varrho(x_{k-1},\nu_{k-1}^-) \leq -\beta t_k^2\|d_k\| + \delta_k\\ &\Gamma_\alpha^\varrho(x_k,\nu_k^-) \geq 0\end{aligned}$$

*hold for all $k$. Then $\lim_{k\to\infty}\|t_k^2 d_k\| = 0$.*

*Proof.* Exploiting the definition of $\varepsilon_k$ approximation, the line search procedure implies that :

$$\Gamma_\alpha^\varrho(x_k,\nu_\alpha(x_k)) - \varepsilon_k - \Gamma_\alpha^\varrho(x_{k-1},\nu_\alpha(x_{k-1})) \leq -\beta t_k^2\|d_k\| + \delta_k,$$

holds for any $k$. The above inequality leads to

$$\sum_{k=2}^N [\Gamma_\alpha^\varrho(x_k,\nu_\alpha(x_k)) - \Gamma_\alpha^\varrho(x_{k-1},\nu_\alpha(x_{k-1}))] \leq \sum_{k=2}^N -\beta t_k^2\|d_k\| + \sum_{k=2}^N(\delta_k+\varepsilon_k),$$

where $N>2$. Thus, multiplying by $-1$, we obtain

$$-\Gamma_\alpha^\varrho(x_N,\nu_\alpha(x_N)) + \Gamma_\alpha^\varrho(x_1,\nu_\alpha(x_1)) + \sum_{k=2}^N(\delta_k+\varepsilon_k) \geq \sum_{k=2}^N \beta t_k^2\|d_k\|.$$

Since $\Gamma_\alpha^\varrho(x_N,\nu_N^-)\geq 0$, then also $\Gamma_\alpha^\varrho(x_N,\nu_\alpha(x_N))\geq 0$. Thus, we obtain that

$$\Gamma_\alpha^\varrho(x_1,\nu_\alpha(x_1)) + \sum_{k=2}^N(\delta_k+\varepsilon_k) \geq \sum_{k=2}^N \beta t_k^2\|d_k\|.$$

holds for any $N > 2$. Thus, since $\sum_{k=1}^{\infty}(\delta_k + \varepsilon_k)$ is convergent, then also $\sum_{k=1}^{\infty} \beta t_k^2 \|d_k\|$ is. As a consequence the generic series term have to go to 0:

$$\lim_{k\to\infty} \|t_k^2 d_k\| = 0.$$

□

**Lemma 2.8**

*Consider positive sequences* $\{\alpha_k\}, \{\varepsilon_k\}, \{\varrho_k\}$, *and sequences* $\{x_k\}, \{\nu_k\}$ *in* $D$. *Suppose that the set* $S \subset \mathbb{N}$ *induces a subsequence such that*

$$\lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} (x_k, \alpha_k, \varrho_k, \varepsilon_k) = (x^*, 0, 0, 0) \text{ and}$$

$$\Gamma_{\alpha_k}^{\varrho_k}(x_k, \nu_k) < 0 \qquad (\neg C1)$$

*or*

$$-\eta(\Gamma_{\alpha_k}^{\varrho_k}(x_k, \nu_k) + \varepsilon_k) < -\Gamma_{\alpha_k}^{\varrho_k}(x_k, \nu_k) - \alpha_k[h(x_k, \nu_k) + \langle \nabla_x h(x_k, \nu_k), \nu_k - x_k\rangle] \quad (\neg C2)$$

*holds for any* $k \in S$, *for some given sequence* $\{\nu_k\}$ *of* $\varepsilon_k$ *approximated solutions of* $(\mathcal{P}_{x_k}^{\alpha_k})$. *Then* $x^*$ *solves (EP).*

*Proof.* Either condition $\neg C1$ or condition $\neg C2$ has to be true an infinite number of times. Suppose there exists an infinite set $L \subset S$ such that

$$\Gamma_{\alpha_k}^{\varrho_k}(x_k, \nu_k) < 0$$

holds for any $k \in L$. where $\{\nu_k\}$ is a sequence of $\varepsilon_k$ approximated solutions of $(\mathcal{P}_{x_k}^{\alpha_k})$.

Then, by Definition 2.7 we also have that

$$\Gamma_{\alpha_k}^{\varrho_k}(x_k, \nu_{\alpha_k}(x_k)) < 0 \quad \forall k \in L. \tag{2.7}$$

Since $\nu_{\alpha_k}(x_k)$ belongs to a compact set, we can assume without loss of generality that $\lim_{\substack{k\to\infty \\ k\in L\subset\mathbb{N}}} \nu_{\alpha_k}(x_k) = \nu^*$. In addition, we notice that taking the limit as $k \to \infty$ with $k \in L$ in the following inequalities

$$f(x_k, \nu_{\alpha_k}(x_k)) + \alpha_k h(x_k, \nu_{\alpha_k}(x_k)) \le f(x_k, \nu) + \alpha_k h(x_k, \nu) \quad \forall \nu \in P(x_k),$$

leads to

$$f(x^*, \nu^*) \le f(x^*, \nu) \quad \forall \nu \in P(x^*). \tag{2.8}$$

Notice that we exploited the fact that the point to set map $P : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and therefore open, that is for any $\nu \in P(x^*)$ we can devise a sequence of $\nu_k \in P(x_k)$ converging to $\nu$.

As a consequence of (2.8), we have $\nu^* \in \arg\min\{f(x^*, \nu) : \nu \in P(x^*)\}$ . Notice that $x^*$ must belong to $C$, otherwise taking the limit as $k \to \infty$ with $k \in L$ of (2.7), we obtain

$$\Gamma^{\varrho_k}_{\alpha_k}(x_k, \nu_{\alpha_k}(x_k)) = -f_{\alpha_k}(x_k, \nu_{\alpha_k}(x_k)) + \frac{\|c^+(x_k)\|}{\varrho_k} \to +\infty \leq 0$$

since $\frac{\|c^+(x_k)\|}{\varrho_k} \to +\infty$ and $f_{\alpha_k}(x_k, \nu_{\alpha_k}(x_k)) \to f(x^*, y^*)$.

Finally, taking the limit as $k \to \infty$ with $k \in L$ of (2.7), we obtain

$$-f(x^*, \nu^*) \leq 0$$

and therefore $x^*$ solves (EP). This concludes the case in which $\neg(C1)$ is true an infinite number of times. Now we move to the other case.

Suppose there exists $L \subset S$ such that

$$-\eta(\Gamma^{\varrho_k}_{\alpha_k}(x_k, \nu_k) + \varepsilon_k) < -\Gamma^{\varrho_k}_{\alpha_k}(x_k, \nu_k) - \alpha_k[h(x_k, \nu_k) + \langle \nabla_x h(x_k, \nu_k), \nu_k - x_k \rangle]$$

holds for any $k \in L$. Rewriting the above inequality we obtain

$$\Gamma^{\varrho_k}_{\alpha_k}(x_k, \nu_k) < \frac{-\alpha_k[h(x_k, \nu_k) + \langle \nabla_x h(x_k, \nu_k), \nu_k - x_k \rangle] + \eta\varepsilon_k}{1 - \eta} \quad \forall k \in L_2.$$

Because the RHS goes to 0 as $k \to +\infty$, we obtain that $x^*$ solves (EP) just arguing as in the former case.

□

Another important consequence of the above Lemma is that if $x_k$ doesn't solve (EP) eventually, the variable $k$ defined in Algorithm 2.2 will increase. In other words is not possibile to have an infinite loop of Steps: $1 \to 2 \to 3 \to 1 \to 2 \to 3$... cycling on some $x_k$ that doesn't solve (EP). In fact, consider the variables $a_{I_\alpha}, \rho_{I_\varrho}, \epsilon_{I_\varepsilon}$ and $y$: each time the GOTO 1 is taken at Step 3 C1 or C2 is false, in addition, the variables $a_{I_\alpha}, \rho_{I_\varrho}, \epsilon_{I_\varepsilon}$ decrease thus an infinite loop would generate a going to zero sequence and match the premises of the above Lemma.

**Theorem 2.4** (Correctness)
*Suppose that $f$ is c-monotone and let $x^*$ be a cluster point of the sequence $\{x_k\}$ generated by Algorithm 2.2, then $x^*$ solves (EP).*

*Proof.* Let $S \subset \mathbb{N}$ be the set of all indexes $k$ such that the GOTO at Step 3 has been taken at that $k$. If $|S| = |\mathbb{N}|$, then exist a subset of $\mathbb{N}$ satisfying the assumptions of Lemma 2.8, and thus, $x^*$ solves (EP). Instead, if $S$ is finite, we have that $\varrho_k = \varrho$ and

$\alpha_k = \alpha$, and C1, C2, C3 hold from some $k$ onwards. Indeed if $k > \max\{n \in S\}$ C1 and C2 hold and the only possibility to update $\varrho_k$ is that the condition $\neg C3$ at Step 4 is true, but this could happen only for a finite number of times. Otherwise, we would have a subsequence of $\lambda_k \in \Lambda_\alpha^{\varepsilon_k}(x_k)$ for some $x_k \to x^* \in D$, with $\varepsilon_k \to 0$ such that:

$$\begin{aligned}
&\lambda_k \to \lambda^* \in \Lambda_\alpha(x^*) \subset \mathbb{R}^m \\
&\frac{1}{\varrho_k} \leq \|\lambda_k^+\| \\
&\frac{1}{\varrho_k} \to \infty,
\end{aligned}$$

where the first statement follows directly from Definition 2.8, the second is the violation of $(C3)$ and the third is due to the going to zero of $\varrho_k$.

Hence we now deal with the case when $\varrho_k = \varrho$ and $\alpha_k = \alpha$, and C1, C2, C3 hold from some $k$ onwards. Then thanks to the line search at Step 8, the assumptions of Lemma 2.7 are met and so we have that $\|t_k^2 d_k\| \to 0$. Let $L$ be an infinite subset of $\mathbb{N}$ such that the limit of $x_k$ as $k \to \infty$ with $k \in L$ is $x^*$.

We distinguish two cases. If $\|d_k\| \to 0$ for $k \in L$, then we can choose a convergent subsequence of $\varepsilon_k$ approximated solutions $\nu_k \to \overline{\nu}$ for $k \in L$ such that $d_k = \nu_k - x_k$, (remember that $D$ is compact). Hence, since $\varepsilon_k \to 0$, then $\|\nu_\alpha(x_k) - \nu_k\| \to 0$ and therefore $\overline{\nu} = \nu_\alpha(x^*)$ and $d^* = \nu_\alpha(x^*) - x^*$ equal to zero actually means that $x^*$ is a solution of (EP).

Now, suppose that $t_k \to 0$. If for some infinite set $R \subseteq L$ we have that $t_k = 0$ we can reduce again to the case $\|d_k\| \to 0$ for $k \in R$ , hence we can limit ourselves to the case $t_k > 0$ from $k \geq \overline{k}$ onwards for some $\overline{k} \in \mathbb{N}$. By the way the line search at Step 8 is performed, there exist two sequences $\nu_k^+, \nu_k^- \in P(x)$ being respectively an $\varepsilon_k$ and an $\varepsilon_{k+1}$ approximated solutions of $(\mathcal{P}_{x_k}^\alpha)$ for which

$$\Gamma_\varrho^\alpha(x_{k-1} + \frac{t_k d_k}{\gamma}, \nu_k^+) - \Gamma_\varrho^\alpha(x_{k-1}, \nu_{k-1}^-) > -\beta(\frac{t_k}{\gamma})^2\|d_k\| + \delta_k,$$

at every step $k > \overline{k} \in \mathbb{N}$.

Thus, by Definition 2.7 we can affirm

$$\Gamma_\varrho^\alpha(x_{k-1} + \frac{t_k d_k}{\gamma}, \nu_\alpha(x_{k-1} + \frac{t_k d_k}{\gamma})) - \Gamma_\varrho^\alpha(x_{k-1}, \nu_\alpha(x_{k-1})) + \varepsilon_k > -\beta(\frac{t_k}{\gamma})^2\|d_k\| + \delta_k$$

holds for any $k > \overline{k}$. The mean value theorem guarantees the existence of $\theta_k \in (0, 1)$ such that

$$\Gamma_\varrho^\alpha(x_{k-1} + \frac{t_k d_k}{\gamma}, \nu_\alpha(x_{k-1} + \frac{t_k d_k}{\gamma})) - \Gamma_\varrho^\alpha(x_{k-1}, \nu_\alpha(x_{k-1})) = \langle \xi_k, \frac{t_k d_k}{\gamma} \rangle,$$

where $\xi_k$ is a generalized gradient of $\Gamma^{\alpha}_{\varrho}(\cdot, \nu_\alpha(\cdot))$ at $x_{k-1} + \theta_k \frac{t_k d_k}{\gamma}$. Because of the generalized gradient proprieties, we have

$$\Gamma^{\alpha\circ}_{\varrho}(x_{k-1} + \theta_k \frac{t_k d_k}{\gamma}, \nu_\alpha(x_{k-1} + \theta_k \frac{t_k d_k}{\gamma}); d_k) \geq \langle \xi_k, d_k \rangle \quad \forall k > \overline{k}.$$

Hence,

$$\Gamma^{\alpha\circ}_{\varrho}(x_{k-1} + \theta_k \frac{t_k d_k}{\gamma}, \nu_\alpha(x_{k-1} + \theta_k \frac{t_k d_k}{\gamma}); d_k) > -\beta \frac{t_k}{\gamma} \|d_k\| + \delta_k - \varepsilon_k \tag{2.9}$$

holds for any $k \geq \overline{k}$.

Exploiting the upper semicontinuity of the generalized derivative (see e.g. [Cla87]) taking the lim sup as $k \to +\infty$ with $k \in L$ of (2.9) we get

$$\Gamma^{\alpha\circ}_{\varrho}(x^*, \nu_\alpha(x^*); d^*) \geq 0. \tag{2.10}$$

Now, there can be two cases based wheter or not $x^*$ belongs to $C$.

Suppose that $x^* \in C$ and but it doesn't solve (EP).

Notice that $-\Gamma^{\alpha\circ}_{\varrho}(x^*, \nu_\alpha(x^*); d^*) \leq -\eta \Gamma^{\alpha}_{\varrho}(x^*, \nu_\alpha(x^*))$. Indeed C2 guarantees

$$\begin{aligned}
&- \Gamma^{\alpha}_{\varrho}(x_k, \nu_\alpha(x_k)) - \alpha[h(x_k, \nu_\alpha(x_k)) + \langle \nabla_x h(x_k, \nu_\alpha(x_k)), \nu_\alpha(x_k) - x_k \rangle] \\
&\leq -\Gamma^{\alpha}_{\varrho}(x_k, \nu_k^+) - \alpha[h(x_k, \nu_\alpha(x_k)) + \langle \nabla_x h(x_k, \nu_\alpha(x_k)), \nu_\alpha(x_k) - x_k \rangle] \\
&\leq -\eta(\Gamma^{\alpha}_{\varrho}(x_k, \nu_k^+) + \varepsilon_k) - \alpha[h(x_k, \nu_\alpha(x_k)) + \langle \nabla_x h(x_k, \nu_\alpha(x_k)), \nu_\alpha(x_k) - x_k \rangle] \\
&+ \alpha[h(x_k, \nu_k^+) + \langle \nabla_x h(x_k, \nu_k^+), \nu_k^+ - x_k \rangle],
\end{aligned}$$

from which, taking the limit as $k \to \infty$, $k \in L$, we obtain the following chain of inequalities

$$\begin{aligned}
&- \eta \Gamma^{\alpha}_{\varrho}(x^*, \nu_\alpha(x^*)) \\
&\geq -\Gamma^{\alpha}_{\varrho}(x^*, \nu_\alpha(x^*)) - \alpha[h(x^*, \nu_\alpha(x^*)) + \langle \nabla_x h(x^*, \nu_\alpha(x^*)), d^* \rangle] \\
&= -\Psi^{\alpha}_{\varrho}(x^*) - \alpha[h(x^*, \nu_\alpha(x^*)) + \langle \nabla_x h(x^*, \nu_\alpha(x^*)), d^* \rangle] \\
&\geq \Psi^{\alpha\circ}_{\varrho}(x^*; d^*) = \Gamma^{\alpha\circ}_{\varrho}(x^*, \nu_\alpha(x^*); d^*).
\end{aligned} \tag{2.11}$$

Where the last inequality follows from Lemma 2.5 (i). Therefore, if $x^*$ doesn't solve (EP) and $x^* \in C$,

(2.11) contradicts (2.10), since $-\eta \Gamma^{\alpha}_{\varrho}(x^*, \nu_\alpha(x^*)) = -\varphi_\alpha(x^*) < 0$.

Suppose that $x^* \notin C$ and it does not solve (EP). Condition C3 guarantees $\frac{1}{\varrho} \geq \|(\lambda_k^+)\| + \delta$, with $\lambda_k \in \Lambda_\alpha^{\varepsilon_k}(x_k)$. Hence, exploiting DEFINITION 2.8 we can take a subsequence converging to $\lambda^* \in \Lambda_\alpha(x^*)$, such that $\frac{1}{\varrho} \geq \|(\lambda^*)^+)\| + \delta$. Thus, by LEMMA 2.5(ii), $\Gamma_\varrho^{\alpha\circ}(x^*, \nu_\alpha(x^*); d^*) < 0$, contradicting (2.10).

□

# Chapter 3

# On the Computation of the Inner Problem's Solutions

ALGORITHM 2.2 generates a sequence of $x_k \in D$, and it needs both some $\varepsilon_k$ approximated solution $y_k$ and some $\lambda_k \in \Lambda_{\alpha_k}^{\varepsilon_k}(x_k)$ at each step.

In this chapter, we firstly study how the characteristics of $y_k$ and $\lambda_k$ are linked to a perturbation of the well known $KKT$ optimality conditions for $(\mathcal{P}_x^\alpha)$. Afterwards we propose two methods. The first, is derived from the Frank and Wolfe algorithm and it reduces to a sequence of linear programs, interestingly, theoretical results show that it could be arranged to converge rapidly in our framework. The second is derived from the Fiacco and McCormick's Barrier Method and it reduces to a single unconstrained nonlinear minimization problem.

All the considerations of this chapter are applicable with not much effort to the simpler case of ALGORITHM 1.3. Please notice that for the aims of this chapter, we can consider $x_k = x \in D, \varepsilon_k = \varepsilon > 0$ and $\alpha_k = \alpha > 0$ fixed.

## 3.1 A Subset of Approximated Dual Solutions

In this section we study the point to set map $U_\varepsilon$, which associates to a point $x \in \mathbb{R}^n$ a subset of $\varepsilon$ approximated dual solutions. $U_\varepsilon$ and the function $L^*$ defined below, are common tool which are exploited in literature (e.g. [Hog73a]) to study the asymptotic behaviour of properties of approximated dual/primal solutions as the approximation goes to zero. In our case they constitute the main tool to prove that a dual feasible vector $\lambda \in \mathbb{R}^m$ belongs to the set $\Lambda_\alpha^\varepsilon(x)$.

In order to express compactly the feasible region of $(\mathcal{P}_x^\alpha)$ and abstract from its real structure we introduce the functions $g_i$.

**Definition 3.1** ($g_i$)
We define the functions $g_i : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ for $i = 1, 2, \ldots, m$ as follows:

$$g_i(x, y) = c_i(x) + \langle \nabla c_i(x), y - x \rangle.$$

Where the functions $c_i$ are the ones defined in Assumption 2.1.

We can now express the polyhedron $P(x)$ as follows:

$$P(x) = \{y \in D \ : \ g_i(x, y) \leq 0\}.$$

Notice that in this way, with a mere substitution $g_i(x, y) = c_i(x)$ we can cover also the case of the algorithm Algorithm 1.3.

**Definition 3.2** ($L^*$)
We define the function $L^* : \mathbb{R}^{n+m} \to \mathbb{R}$ as:

$$L^*(x, u) = \max_{y \in D} - f_\alpha(x, y) - \sum_{i=1}^{m} u_i g_i(x, y)$$

The function $(y, \lambda) \mapsto f_\alpha(x, y) + \sum_{i=1}^{m} u_i g_i(x, y)$ is a partial Lagrangian of $(\mathcal{P}_x^\alpha)$. Indeed, it considers only the inequality-constraints that depends upon the considered $x$.

Remark 3.1. Let $(\lambda^*, \mu^*) \in \mathbb{R}^{m+r}$ be the Lagrangian multipliers for the problem

$$\max_{y \in P(x)} \ - f_\alpha(x, y).$$

Then $L^*(x, \lambda^*) = \psi_\alpha(x)$.

*Proof.* Under Assumption 2.1, strong duality holds for the problem

$$\max_{y \in D} - f_\alpha(x, y) - \sum_{i=1}^{m} u_i g_i(x, y).$$

Hence, we have that

$$\begin{aligned} & L^*(x, \lambda^*) \\ & = \max_{y \in D} - f_\alpha(x, y) - \sum_{i=1}^{m} \lambda_i^* g_i(x, y) \\ & = \min_{\mu \in \mathbb{R}_+^{r_1}} \{ \max_{y \in \mathbb{R}^n} - f_\alpha(x, y) - \sum_{i=1}^{m} \lambda_i^* g_i(x, y) - \sum_{j=1}^{r_1} \mu_j h_j(y) : h_j(y) = 0, \ j = r_1, \ldots, r \} \end{aligned} \tag{3.1}$$

Where in the last equation we have exploited the $h_j$ from the definition of $D$ (see Assumption 2.1).

On the other hand, strong duality holds also for the problem

$$\max_{y \in P(x)} \; - f_\alpha(x, y),$$

and we get

$$\begin{aligned}\psi_\alpha(x) &= \min_{(\mu,\lambda)\in\mathbb{R}_+^{m+r_1}} \{\max_{y\in\mathbb{R}^n} -f_\alpha(x,y) - \sum_{i=1}^{m} \lambda_i g_i(x,y) - \sum_{j=1}^{r_1} \mu_j h_j(y) : h_j(y) = 0,\ j = r_1, \dots, r\} \\ &= - f_\alpha(x,y) - \sum_{i=1}^{m} \lambda_i^* g_i(x,y) - \sum_{j=1}^{r_1} \mu_j^* h_j(y).\end{aligned}$$

Since the minimum is attained at $(\lambda^*, \mu^*)$, $\mu^*$ has to minimize the RHS of (3.1). □

This remark will be exploited in the proof of Lemma 3.4.

**Lemma 3.1**

*(i) $L^*$ is a continuous function over $\mathbb{R}^{n+m}$.*

*(ii) $L^*(x, \cdot)$ is convex for any $x \in \mathbb{R}^n$.*

*Proof.*

(i) The point to set map $x \mapsto D$ is constant, and therefore continuous (see [Hog73c]). In addition $x \mapsto D$ is uniformly compact thanks to Assumption 2.1. The function $-f_\alpha(x,\cdot) - \sum_{i=1}^m u_i g_i(x,\cdot)$ is continuous, thus Theorem A.1 guarantee the continuity of $L^*$.

(ii) Let $\lambda \in [0,1]$ and $u_1, u_2 \in \mathbb{R}_+^m$ then

$$\begin{aligned}&L^*(x, \lambda u_1 + (1-\lambda)u_2) \\ &= \max_{y\in D} -f_\alpha(x,y) - \sum_{i=1}^{m} (\lambda u_1 + (1-\lambda)u_2) g_i(x,y) \\ &= \max_{y\in D} -\lambda f_\alpha(x,y) - \sum_{i=1}^{m} \lambda u_1 g_i(x,y) - (1-\lambda) f_\alpha(x,y) - \sum_{i=1}^{m} (1-\lambda) u_2 g_i(x,y) \\ &\le \max_{y\in D} -\lambda f_\alpha(x,y) - \sum_{i=1}^{m} \lambda u_1 g_i(x,y) + \max_{y\in D} -(1-\lambda) f_\alpha(x,y) - \sum_{i=1}^{m} (1-\lambda) u_2 g_i(x,y) \\ &= \lambda L^*(x,u_1) + (1-\lambda) L^*(x,u_2)\end{aligned}$$

□

**Definition 3.3**
For any $\varepsilon \geq 0$, we define the point to set map $U_\varepsilon : \mathbb{R}^n \to \mathbb{R}^m$ as follows:

$$U_\varepsilon(x) = \{u \in \mathbb{R}^m_+ | L^*(x,u) \leq \psi_\alpha(x) + \varepsilon\}$$

Notice that the above definition allows $\varepsilon$ to be zero. By definition the set $U_0(x)$ is the set of Lagrangian multipliers of the problem $(\mathcal{P}^\alpha_x)$. The following lemma, will be heavily exploited in the proof of Theorem 3.1.

**Lemma 3.2** ([Hog73b])
*If $\psi_\alpha(\overline{x})$ is finite, then the point set map $U_0 : \mathbb{R}^n \to \mathbb{R}^m$ is non empty and uniformly compact near $\overline{x}$, and $U_0$ is closed at $\overline{x}$.*

Now we will prove that uniformly compactness is achieved also in the case $\varepsilon > 0$.

**Lemma 3.3**
*Given any $\varepsilon \geq 0$, $U_\varepsilon$ is a closed point to set map.*

*Proof.* Just apply Theorem A.3 with $P(x) \equiv U_\varepsilon(x)$, $Y \equiv \mathbb{R}^m_+$ and $g(x,u) \equiv L^*(x,u)$. □

**Lemma 3.4**
*Given any $\varepsilon > 0$, $U_\varepsilon$ is an open point to set map.*

*Proof.* We invoke Theorem A.4 with $P(x) \equiv \{y \in Y \mid g(x,y) \leq 0\}$ for any $x \in \mathbb{R}^n$, $Y \equiv \mathbb{R}^m_+$ and $g(x,u) \equiv L^*(x,u) - \psi_\alpha(x) - \varepsilon$, which actually means $P(x) = U_\varepsilon(x)$. In order to apply Theorem A.4, we have to prove that $g$ is continuous on $x \times P(x)$, that $g(x,\cdot)$ is convex (both ensured by Lemma 3.1) and that for each fixed $x \in \mathbb{R}^n$, there exists $u \in Y$ such that $g(x,u) < 0$. This could be seen, observing that fixed $x \in \mathbb{R}^n$, there exists $u \in \mathbb{R}^m_+ :\ L^*(x,u) - \psi_\alpha(x) - \varepsilon < 0$. Indeed Remark 3.1 shows that for any $x$ we can choose the optimal Lagrangian multiplier $u^*$ and obtain $L^*(x,u^*) = \psi_\alpha(x)$. □

Lemma 3.3 and Lemma 3.4 guarantee the continuity of the point to set map $U_\varepsilon$.

**Corollary 3.1**
*For any $\varepsilon > 0$ the point to set map $U_\varepsilon$ is continuous.*

**Lemma 3.5**
*Let $\varepsilon > 0$ and suppose that $U_\varepsilon(\overline{x})$ is non empty. Then, the point to set map $U_\varepsilon : \mathbb{R}^n \to \mathbb{R}^m$ is non empty and uniformly compact near $\overline{x}$.*

*Proof.* Let $f : \mathbb{R}^{2n} \to \mathbb{R}$ be a constant function, let $\Omega(x) \equiv U_\varepsilon(x)$, $v(x) \equiv \sup_{y \in \Omega(x)} f(x,y)$ and $M(x) \equiv \{y \in \Omega(x) | v(x) \leq f(x,y)\}$. Since $f$ is constant, $M(x) = \Omega(x)$.

Moreover $f$ is continuous and quasiconcave and, Lemma 3.4 and Lemma 3.3, guarantee that $\Omega$ is closed on a neighborhood of $\overline{x}$ and open at $\overline{x}$.

Furthermore, $M(\overline{x}) = U_\varepsilon(\overline{x})$ is bounded: otherwise, there would exist $u_k \in U_\varepsilon(\overline{x})$ such that $\|u_k\| \to +\infty$, so, there could exist an index $i^* \in \{1, 2, \dots, m\}$ such that $u_{k_{i^*}} \to +\infty$. Let $\overline{y} \in D$ satisfy $g_i(\overline{x}, \overline{y}) < 0$ for any $i$ (the existence of such a vector $\overline{y}$ is ensured by Assumption 2.1), then we would have

$$L^*(\overline{x}, u_k) = \max_{y \in D} -f_\alpha(\overline{x}, y) - \sum_{i=1}^{m} u_{k_i} g_i(\overline{x}, y) \geq -f_\alpha(\overline{x}, \overline{y}) - \sum_{i=1}^{m} u_{k_i} g_i(\overline{x}, \overline{y}),$$

Therefore, since $-u_{k_i} g_i(\overline{x}, \overline{y}) \to z_i \in \mathbb{R}_+ \cup \{+\infty\}$ for any $i$, we would have that $L^*(\overline{x}, u_k) \to +\infty$. Obviously $L^*(\overline{x}, u_k) \leq \psi_\alpha(\overline{x}) + \varepsilon$ since $u_k \in U_\varepsilon(\overline{x})$, and thus the finiteness of $\psi_\alpha(\overline{x})$ would led to a contradiction.

Hence, all the assumptions of Theorem A.5 are met and the thesis follows. □

Our aim is to prove that $U_\varepsilon(x) \subseteq \Lambda_\alpha^\varepsilon(x)$. We already showed that $U_\varepsilon(x)$ is bounded. The following lemma shows that we can extract a converging subsequence, then exploiting Lemma 3.1, we will prove the desired inclusion.

**Theorem 3.1**
*Let $\{x_k\}$ and $\{\varepsilon_k\}$ be two sequences such that $\varepsilon_k > 0$, $\varepsilon_k \to 0$ and $x_k \to x^*$ for some $x^*$ at which $\psi_\alpha(x^*)$ finite, then $\cup_{k>0} U_{\varepsilon_k}(x_k)$ is bounded.*

*Proof.* By contradiction, we prove that the existence of a sequence $\{u_k\}$ such that $u_k \in \cup_{k>0} U_{\varepsilon_k}(x_k)$ and $\|u_k\| \to +\infty$ is not possible.

By Lemma 3.5, $U_{\varepsilon_k}(x_k)$ is bounded for any $k$. Thus, in order to prove that $\|u_k\| \to +\infty$ is impossible, we can just consider the case $u_k \in U_{\varepsilon_k}(x_k)$.

Suppose there exists a sequence $\{u_k\}$ such that $u_k \in U_{\varepsilon_k}(x_k)$ and $\|u_k\| \to +\infty$. Thanks to Lemma 3.2, we can construct a sequence $z_k \in U_0(x_k)$, such that $z_k \to z^* \in U_0(x^*)$.

Moreover, given an abitrarily large $M > 0$, we can construct also a sequence $\{w_k\}$ satisfying the following conditions:

(a) $w_k = \lambda_k z_k + (1 - \lambda_k) u_k$,

(b) $\lambda_k \in [0, 1]$,

(c) $w_k$ converges to some $w^*$ such that $M < \|w^*\| < +\infty$.

If $M < \|z^*\|$, it is enought to set $\lambda_k = 1$ to satisfy (a), (b) and (c). Hence, suppose $M > \|z^*\|$, and choose $\lambda_k = 1 - \frac{\delta}{\|u_k\|}$ with $\delta > 0$. We have

$$\begin{aligned} \|w_k\|^2 &= \lambda_k^2\|z_k\|^2 + (1-\lambda_k)^2\|u_k\|^2 + 2\lambda_k(1-\lambda_k)z_k^T u_k \\ &= (1 - \frac{\delta}{\|u_k\|})^2\|z_k\|^2 + \delta^2 + 2\delta(1 - \frac{\delta}{\|u_k\|})z_k^T(\frac{u_k}{\|u_k\|}) \\ &\leq (1 - \frac{\delta}{\|u_k\|})^2\|z_k\|^2 + \delta^2 + 2\delta(1 - \frac{\delta}{\|u_k\|})\|z_k\|, \end{aligned} \tag{3.2}$$

where we have exploited the well known Cauchy–Schwarz inequality. As a consequence, taking the limit in (3.2) as $k \to +\infty$, we obtain that $\|w^*\| < +\infty$. Furthermore, it holds

$$\|w_k\|^2 \geq (1 - \frac{\delta}{\|u_k\|})^2\|z_k\|^2 + \delta^2,$$

In fact, since $U_\varepsilon(x) \subset \mathbb{R}^m_+$ for any $\varepsilon \geq 0$ and any $x \in \mathbb{R}^n$ we have $2\lambda_k(1-\lambda_k)z_k^T u_k \geq 0$. Now, taking the limit as $k \to +\infty$ in the above inequality, we obtain

$$\|w^*\| \geq \|z^*\|^2 + \delta^2.$$

Since $\|u_k\| \to \infty$, we can set $\delta = \sqrt{M}$ and subsequencing to fulfil the requirement $0 \leq \lambda_k \leq 1$. Hence, we can assume to have sequence a $\{w_k\}$ that satisfies $(a), (b)$ and $(c)$.

By Lemma 3.1, we have that $L^*(x_k, \cdot)$ is convex. Therefore, it holds

$$L^*(x_k, w^k) \leq \lambda_k L^*(x_k, z_k) + (1-\lambda_k)L^*(x_k, u_k) \leq \lambda_k\psi_\alpha(x_k) + (1-\lambda_k)(\psi_\alpha(x_k) + \varepsilon_k),$$

where the last inequality follows from $z_k \in U_0(x_k)$ and $u_k \in U_{\varepsilon_k}(x_k)$. Moreover, we proved that $L^*$ and $\psi_\alpha$ are continuous functions. Thus, taking the limit we get $L^*(x^*, w^*) \leq \psi_\alpha(x^*)$, and hence $w^* \in U_0(x^*)$. Anyway, by construction we can make $\|w^*\|$ arbitrarily big contradicting the boundedness of $U_0(x^*)$, which is guaranteed by Lemma 3.2. □

We are finally ready to state the following corollary:

**Corollary 3.2**
$U_\varepsilon(x) \subseteq \Lambda^\varepsilon_\alpha(x)$.

*Proof.* Lemma 3.5 guarantees that Definition 2.7 a) holds. Consider a sequence $\lambda_k \in U_{\varepsilon_k}(x_k)$ with $x_k \to x^* \in D$ and $\varepsilon_k \to 0$. Theorem 3.1 guarantees that we can construct a converging subsequence $\lambda_{k_j} \to \lambda^*$. Furthermore since $L^*$ is continuous, we obtain that $\lambda^* \in U_0(x^*)$, i.e., $\lambda^*$ is a Lagrangian multiplier of $(\mathcal{P}^\alpha_x)$. Hence, also Definition 2.7 b) holds. □

## 3.2 Relaxating the Complementary Slackness Conditions

In this section we study a weaker case of KKT conditions (named $\varepsilon$KKT($\eta$) conditions), obtained relaxating the complementary slackness conditions. Afterwards, we prove that vectors which satisfy them provide suitable $\varepsilon$ approximated solution $y$ and $\lambda \in \Lambda_\alpha^\varepsilon(x)$.

**Definition 3.4** ($\varepsilon$KKT($\eta$))
Given $\eta \in \operatorname{int} \mathbb{R}^m_+$, and $\varepsilon > 0$ we say that a vector $(y, \lambda, \mu) \in \mathbb{R}^{n+m+r}$ satisfies the $\varepsilon\mathrm{KKT}_x(\eta)$ conditions if

$$\nabla_y f_\alpha(x,y) + \sum_{i=1}^{m} \lambda_i \nabla_y g_i(x,y) + \sum_{j=1}^{r} \mu_j \nabla h_j(y) = 0 \qquad (\varepsilon\mathrm{KKT}_x(\eta)1)$$

$$\sum_{i=1}^{m} \lambda_i g_i(x,y) \geq -\varepsilon \qquad (\varepsilon\mathrm{KKT}_x(\eta)2)$$

$$\mu_j h_j(y) = 0 \quad j = 1,2,\ldots,r_1 \qquad (\varepsilon\mathrm{KKT}_x(\eta)3)$$

$$g_i(x,y) \leq -\eta_i \quad i = 1,2,\ldots,m \qquad (\varepsilon\mathrm{KKT}_x(\eta)4)$$

$$y \in D \qquad (\varepsilon\mathrm{KKT}_x(\eta)5)$$

$$\lambda \in \mathbb{R}^m_+,\ \mu \in \mathbb{R}^{r_1}_+ \times \mathbb{R}^{r-r_1} \qquad (\varepsilon\mathrm{KKT}_x(\eta)6)$$

The following Lemma shows that the $\varepsilon\mathrm{KKT}_x(\eta)$ conditions are sufficient for $y$ to be an $\varepsilon$ approximated solution of $(\mathcal{P}_x^\alpha)$.

**Lemma 3.6**
*If $(y, \lambda, \mu) \in \mathbb{R}^{n+m+r}$ satisfies the $\varepsilon KKT_x(\eta)$ conditions, then $y$ is an $\varepsilon$ approximated solution of $(\mathcal{P}_x^\alpha)$.*

*Proof.* Let $y^*$ be a solution of $(\mathcal{P}_x^\alpha)$. By ($\varepsilon\mathrm{KKT}_x(\eta)4$) and ($\varepsilon\mathrm{KKT}_x(\eta)5$), $y \in P(x)$ and therefore $f_\alpha(x,y^*) \leq f_\alpha(x,y)$. Thus, it is enough to prove that $f_\alpha(x,y^*) + \varepsilon \geq f_\alpha(x,y)$. Exploiting the convexity of $g_i(x,\cdot)$, $h_j$ and ($\varepsilon\mathrm{KKT}_x(\eta)1$), we obtain

$$\begin{aligned} 0 &= \langle \nabla_y f_\alpha(x,y) + \sum_{i=1}^{m} \lambda_i \nabla_y g_i(x,y) + \sum_{i=1}^{r} \mu_i \nabla h_i(y), y^* - y \rangle \\ &\leq f_\alpha(x,y^*) - f_\alpha(x,y) + \sum_{i=1}^{m} \lambda_i (g_i(x,y^*) - g_i(x,y)) + \sum_{j=1}^{r} \mu_j (h_j(y^*) - h_j(y)). \end{aligned}$$

Since $y^* \in P(x)$ and $\lambda \in \mathbb{R}^m_+$, $\lambda_i g_i(x,y^*) \leq 0$ for any $i$. Moreover, it holds

$$\begin{aligned} \sum_{j=1}^{r} \mu_j (h_j(y^*) - h_j(y)) &= \sum_{j=1}^{r_1} \mu_j (h_j(y^*) - h_j(y)) \\ &\leq \sum_{j=1}^{r_1} \mu_j (-h_j(y)) = 0. \end{aligned}$$

Where the first equality holds since $y, y^* \in D$, the inequality holds since $\mu_j \geq 0$ and $y^* \in D$ and the last equality follows from ($\varepsilon$KKT$_x(\eta)$3). Therefore, we obtain

$$0 \leq f_\alpha(x, y^*) - f_\alpha(x, y) + \sum_{i=1}^{m} \lambda_i (g_i(x, y^*) - g_i(x, y)) + \sum_{j=1}^{r} \mu_j (h_j(y^*) - h_j(y))$$
$$\leq f_\alpha(x, y^*) - f_\alpha(x, y) - \sum_{i=1}^{m} \lambda_i g_i(x, y).$$

Where we implicitly exploit that $\lambda \in \mathbb{R}^m_+$, $y^* \in P(x)$ implies $\lambda_i g_i(x, y^*) \leq 0$. Finally applying ($\varepsilon$KKT$_x(\eta)$2) to the last inequality, we conclude that

$$0 \leq f_\alpha(x, y^*) - f_\alpha(x, y) - \sum_{i=1}^{m} \lambda_i g_i(x, y) \leq f_\alpha(x, y^*) - f_\alpha(x, y) + \varepsilon.$$

□

The remaining part of this section is devoted to prove that if $(y, \lambda, \mu) \in \mathbb{R}^{n+m+r}$ satisfies the $\varepsilon$KKT$_x(\eta)$ conditions, then $\lambda \in U_\varepsilon(x)$ (and thus $\lambda \in \Lambda^\varepsilon_\alpha(x)$).

**Proposition 3.1**

*Let $x \in \mathbb{R}^n$ and $\varepsilon > 0$ be given. If $(\overline{y}, \lambda, \mu) \in \mathbb{R}^{n+m+r}$ satisfies $\varepsilon KKT_x(\eta)$, then $\lambda \in U_\varepsilon(x)$.*

*Proof.* Given $x \in \mathbb{R}^n$, by definition it is required that $L^*(x, \lambda) \leq \psi_\alpha(x) + \varepsilon$ for $\lambda$ to belong to $U_\varepsilon(x)$.

Consider the problem

$$\min_{y \in D} f_\alpha(x, y) + \sum_{i=1}^{m} \lambda_i g_i(x, y). \tag{P($\lambda$)}$$

The convexity of $f_\alpha(x, \cdot) + \sum_{i=1}^m \lambda_i g_i(x, \cdot)$, guarantees that the KKT are sufficient to be a minimizer. KKT conditions for (P($\lambda$)) are indeed satisfied by $(\overline{y}, \lambda, \mu)$, since they are exactly ($\varepsilon$KKT$_x(\eta)$1), ($\varepsilon$KKT$_x(\eta)$3), ($\varepsilon$KKT$_x(\eta)$5) and ($\varepsilon$KKT$_x(\eta)$6). Therefore, $\overline{y}$ is a solution of (P($\lambda$)) and we have

$$\begin{aligned} -L^*(x, \lambda) = & \min_{y \in D} f_\alpha(x, y) + \sum_{i=1}^{m} \lambda_i g_i(x, y) \\ & = f_\alpha(x, \overline{y}) + \sum_{i=1}^{m} \lambda_i g_i(x, \overline{y}) \\ & \geq -\psi_\alpha(x) + \sum_{i=1}^{m} \lambda_i g_i(x, \overline{y}) \\ & \geq -\psi_\alpha(x) - \varepsilon. \end{aligned}$$

Where the first inequality follows from LEMMA 3.6 and the second from ($\varepsilon\text{KKT}_x(\eta)2$). □

## 3.3 Existence of Solutions

In this section we show that at least one vector satisfying conditions $\varepsilon\text{KKT}_x(\eta)$ for a given $\varepsilon$ always exists.

**DEFINITION 3.5**
Let $\eta \in \mathbb{R}^m_+$. We define the point-to-set map $P : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ as

$$P(x,\eta) = \{y \in D |\ g_i(x,y) + \eta_i \leq 0,\ i = 1,2,\ldots,m\}.$$

**DEFINITION 3.6**
Let $\eta \in \mathbb{R}^m_+$. We define the problem $(P^\eta_x)$ as

$$(P^\eta_x) \qquad \text{find } y \in \underset{y \in P(x,\eta)}{\arg\min}\ f_\alpha(x,y)$$

Under ASSUMPTION 2.1, the continuity of the functions involved in our problem guarantees the existence of $\overline{\eta} \in \text{int } \mathbb{R}^m_+$ such that whenever $\eta \in \mathbb{R}^m_+$ satisfies $\eta \leqq \overline{\eta}$ componentwise there exists $\overline{y} \in \mathbb{R}^n$ such that $g_i(x,\overline{y}) + \eta_i < 0$ and $h_j(\overline{y}) < 0$ hold for any $x \in \mathbb{R}^n$ and $i = 1,2,\ldots m$, $j = 1,2,\ldots,r_1$. Therefore, the set

$$V \equiv \{\eta \in \mathbb{R}^m_+ |\ \exists \overline{y} \in \mathbb{R}^n \forall x \in \mathbb{R}^n : g_i(x,\overline{y}) + \eta_i < 0,\ h_j(\overline{y}) < 0,\ i = 1,\ldots,m,\ j = 1,\ldots,r_1,\},$$

includes an open ball centered at 0. If we restrict to those $\eta \in V$ only, we can assume that the Slater's constraint-qualification holds and that the solutions of $(P^\eta_x)$ satisfy the KKT conditions. Namely, these conditions read:

$$\nabla_y f_\alpha(x,y) + \sum_{i=1}^{m} \lambda_i \nabla_y g_i(x,y) + \sum_{i=1}^{r} \mu_i \nabla h_i(y) = 0 \qquad (\text{KKT}^\eta_x\ 1)$$

$$\lambda_i g_i(x,y) = -\lambda_i \eta_i \quad i = 1,2,\ldots,m \qquad (\text{KKT}^\eta_x\ 2)$$

$$\mu_i h_i(y) = 0 \quad i = 1,2,\ldots,r_1 \qquad (\text{KKT}^\eta_x\ 3)$$

$$g_i(x,y) \leq -\eta_i \quad i = 1,2,\ldots,m \qquad (\text{KKT}^\eta_x\ 4)$$

$$y \in D \qquad (\text{KKT}^\eta_x\ 5)$$

$$\lambda \in \mathbb{R}^m_+,\ \mu \in \mathbb{R}^{r_1}_+ \times \mathbb{R}^{r-r_1} \qquad (\text{KKT}^\eta_x\ 6)$$

Notice that for any $\eta \in V$, $(P^\eta_x)$ admits a solution since the feasible region is non empty and compact and $f_\alpha(x,\cdot)$ is continuous. Furthermore, the solution is unique since $f_\alpha(x,\cdot)$ is strictly convex.

Thanks to the convexity assumptions, also strong duality holds.

Now, we are going to show some proprieties of the optimal multipliers for $(P_x^\eta)$. It is straightforward to see that

$$U(x,\eta) \equiv \{u \in \mathbb{R}_+^m |\ L(x,\eta,u) = \max_{u \geqq 0} L(x,\eta,u)\},$$

where

$$L(x,\eta,u) \equiv \inf_{y \in D} f_\alpha(x,y) + \sum_{i=1}^{m} u_i(g_i(x,y) + \eta_i),$$

is the set of Lagrange multipliers for $(P_x^\eta)$ corresponding to the constraints $g_i$.

**Lemma 3.7**
*For any $(x,\eta) \in \mathbb{R}^n \times V$, the point-to-set map $U : \mathbb{R}^{n+m} \to \mathbb{R}$ is non empty, uniformly bounded near $(x,\eta)$ and closed at $(x,\eta)$.*

*Proof.* Notice that

i) $\min_{y \in P(x,\eta)} f_\alpha(x,y)$ exists and it is finite thanks to the Weierstrass's theorem (since $f_\alpha(x,\cdot)$ is continuous and $P(x,y)$ is compact).

ii) $f_\alpha(x,\cdot)$ is convex and continuous and so are also the functions $(y,\eta) \mapsto g_i(x,y) + \eta$,

iii) Slater's constraint qualification holds for any $x \in \mathbb{R}^n$ and $\eta \in V$,

iv) the solution set $M(x,\eta) = \arg\min_{y \in P(x,\eta)} f_\alpha(x,y)$ is a subset of $D$ and therefore is bounded thanks to Assumption 2.1.

Then, the thesis follow directly from Theorem A.6 considering $f = -f_\alpha$ and $\Omega = P$, $X = \mathbb{R}^n \times V$, $Y = \mathbb{R}^m$. □

### Existence of an $\varepsilon$KKT$_x(\eta)$ vector

Obviously, the existence of a vector $(y,\lambda,\mu) \in \mathbb{R}^{n+m+r}$ satisfing KKT$_x^\eta$ conditions, implies the existence of an $\varepsilon$KKT$_x(\eta)$ vector with $\varepsilon = \sum_{i=1}^m \lambda_i \eta_i$. Intuitively, this leads us to think that for any $\varepsilon > 0$, there exists $\eta \in \operatorname{int} \mathbb{R}_+^m$ such that the system of Definition 3.4 admits at least one solution.

**Theorem 3.2**
*Let $\varepsilon > 0$, then exist $\overline{\eta} \in \operatorname{int} \mathbb{R}_+^m$ such that for any $\eta \in \prod_{i=1}^m [0, \overline{\eta}_i]$ exists a vector $(y,\lambda,\mu) \in \mathbb{R}^{n+m+r}$ satisfying the $\varepsilon KKT_x(\eta)$ conditions.*

*Proof.* Given $\varepsilon > 0$ we construct a sequence such that $\eta_k \in V$, $\eta_k \to 0$ with $\lambda_k \in U(x, \eta_k)$. Thanks to LEMMA 3.7, we can take a subsequence of $\lambda_k$ converging to $\lambda^*$. Therefore, $\sum_{i=1}^m \lambda_{ki}\eta_{ki} \to 0$. Thus, there exists an index $k$ such that $\sum_{i=1}^m \lambda_{ki}\eta_{ki} \leq \varepsilon$ since $\varepsilon > 0$. This guarantees that $\varepsilon\text{KKT}_x^{\eta_k}$ admits a solution. □

## 3.4 Practical Methods to Compute Approximated Primal/-Dual Solutions

A possible way to compute an $\varepsilon$ approximated solution is to solve directly the equations coming out from $\varepsilon\text{KKT}_x(\eta)$ conditions. Unluckily, as for the KKT conditions, this could be difficult or inefficient in some cases. Therefore, we develop a very general method based on the Frank-Wolfe algorithm.

### 3.4.1 Frank & Wolfe Method Review

The Frank-Wolfe method [FW56] is a well known method to solve constrained optimization problems. When applied to the problem $(\mathcal{P}_x^\alpha)$, it generates two sequence of points of $P(x)$: $z_k$ and $y_k$:

$$\begin{aligned} z_k &\in \arg\min_{z \in P(x)} f_\alpha(x, y_{k-1}) + \langle \nabla_y f_\alpha(x, y_{k-1}), z - y_{k-1} \rangle \\ y_k &= (1 - \alpha_{k-1})y_{k-1} + \alpha_{k-1} z_{k-1}, \end{aligned} \tag{3.5}$$

where $\{\alpha_k\}$ is a sequence in $[0, 1]$ such that $\alpha_k \to 0$ and $y_0 \in P(x)$.

**PROPOSITION 3.2**
*Define $\varepsilon_k = f_\alpha(x, y_k) - f_\alpha(x, y_{k-1}) - \langle \nabla_y f_\alpha(x, y_{k-1}), z_k - y_{k-1} \rangle$ then $y_k$ is an $\varepsilon_k$ approximated solution of $(\mathcal{P}_x^\alpha)$for any $k > 0$.*

*Proof.* The sequence of points $y_j$, $j \in \mathbb{N}$ produced by the Frank and Wolfe algorithm belongs to the set $P(x)$. Thus, the only property to be checked is that

$$f_\alpha(x, \nu_\alpha(x)) \geq f_\alpha(x, y_k) - \varepsilon_k.$$

Exploiting the convexity of $f_\alpha(x, \cdot)$ we have that

$$f_\alpha(x, \nu_\alpha(x)) \geq f_\alpha(x, y_{k-1}) + \langle \nabla_y f_\alpha(x, y_{k-1}), \nu_\alpha(x) - y_{k-1} \rangle \geq f_\alpha(x, y_{k-1}) + \langle \nabla_y f_\alpha(x, y_{k-1}), z_k - y_{k-1} \rangle,$$

where the last inequality is due to the optimality of $z_k$. The definition of $\varepsilon$ allows writing the above inequality as

$$f_\alpha(x, y_k) - \varepsilon_k = f_\alpha(x, y_{k-1}) + \langle \nabla_y f_\alpha(x, y_{k-1}), z_k - y_{k-1} \rangle \leq f_\alpha(x, y(x)). \tag{3.6}$$

□

Frank and Wolfe [FW56] proved that if $f_\alpha(x,\cdot)$ is convex and D is compact $y_k \to \nu_\alpha(x)$ as $k \to \infty$. This implies that the error $\varepsilon_k \to 0$. Thanks to the above proposition, breaking the Frank-Wolfe procedure when $\varepsilon_k \leq \varepsilon$, $y_k$ provides the desired $\varepsilon$ approximated solution.

#### 3.4.1.1 How to Obtain Bounded Approximated Dual Solutions

A possible way to obtain elements of $\Lambda_\alpha^\varepsilon(x)$ is to gather them from the solution of the problem

$$\min_{z\in P(x)} f_\alpha(x,y_{k-1}) + \langle \nabla_y f_\alpha(x,y_{k-1}), z - y_{k-1}\rangle$$

that is solved at every iteration of the Frank-Wolfe procedure. In particular we can use as approximated dual solution the solution of the problem

$$\lambda_x \in \arg\max_{\lambda\in\mathbb{R}^m_+} \min_{z\in D} \langle \nabla_y f_\alpha(x,y_{\hat{k}-1}) + f_\alpha(x,y_{\hat{k}-1}), z - y_{\hat{k}-1}\rangle + \sum_{i=1}^m \lambda_i g_i(x,z) \tag{3.7}$$

where $y_{\hat{k}}$ satisfies

$$\varepsilon \geq \varepsilon_{\hat{k}} = f_\alpha(x,y_{\hat{k}}) - f_\alpha(x,y_{\hat{k}-1}) - \langle \nabla_y f_\alpha(x,y_{\hat{k}-1}), z_{\hat{k}} - y_{\hat{k}-1}\rangle. \tag{3.8}$$

For clarity, suppose that our algorithm calls the routine below (see Algorithm 3.1) at each iteration with parameters $x_k$ and $\varepsilon_k$ in input to obtain the approximated primal and dual solution of $\mathcal{P}^\alpha_{x_k}$.

Algorithm 3.1: Truncated Frank & Wolfe

```
TruncatedFrankAndWolfe(x, ε)
  Choose y_0 ∈ P(x), set i = 0.
  do
    Compute (z_i, λ_i), primal and dual solution of the problem
                  min_{z∈P(x)} f_α(x, y_i) + ⟨∇_y f_α(x, y_i), z − y_i⟩.
    α_i = 2/(i+2)
    y_{i+1} = (1 − α_i) y_i + α_i z_i
    error = f_α(x, y_{i+1}) − f_α(x, y_i) − ⟨∇_y f_α(x, y_i), z_i − y_i⟩
    i = i+1
  while (error > ε)
return (y_i, λ_{i−1})
```

Multipliers obtained in this way enjoy the fundamental propriety of belonging to the set $U_\varepsilon(x)$ given in Definition 3.3 as stated by the following Lemma.

**Theorem 3.3**

*Suppose that $\lambda \in \mathbb{R}^m_+$ satisfies (3.7), then $\lambda \in U_\varepsilon(x)$.*

*Proof.* Consider the function $L^*$ given in DEFINITION 3.2. Exploiting the convexity of $f_\alpha(x,\cdot)$, we obtain

$$\begin{aligned} L^*(x,\lambda) &= \max_{z\in D} -f_\alpha(x,z) - \sum_{i=1}^{m} \lambda_i g_i(x,z) \\ &\leq \max_{z\in D} -f_\alpha(x,y_{\hat{k}-1}) - \langle \nabla_y f_\alpha(x,y_{\hat{k}-1}), z - y_{\hat{k}-1}\rangle - \sum_{i=1}^{m} \lambda_i g_i(x,z) \\ &= -f_\alpha(x,y_{\hat{k}-1}) - \langle \nabla_y f_\alpha(x,y_{\hat{k}-1}), z_{\hat{k}} - y_{\hat{k}-1}\rangle. \end{aligned}$$

Thanks to (3.8) we get

$$f_\alpha(x,y_{\hat{k}-1}) + \langle \nabla_y f_\alpha(x,y_{\hat{k}-1}), z - y_{\hat{k}-1}\rangle = f(x,y_{\hat{k}}) - \varepsilon_{\hat{k}} \geq \min_{y\in P(x)} f_\alpha(x,y) - \varepsilon_{\hat{k}} = -\psi_\alpha(x) - \varepsilon_{\hat{k}}.$$

Multiplying the above inequality by -1, we get

$$L^*(x,\lambda) \leq \psi_\alpha(x) + \varepsilon_{\hat{k}} \leq \psi_\alpha(x) + \varepsilon,$$

which is the characteristic property of the vector $\lambda$ belonging to $U_\varepsilon(x)$. □

This result, coupled with THEOREM 3.1, ensures that the procedure described in ALGORITHM 3.1 is suitable to be embedded in algorithm ALGORITHM 2.2.

#### 3.4.1.2 On the Computational Cost

The Truncated Frank-Wolfe algorithm explained in the above section has the advantage of involving only the solution of linear optimization programs. In fact, thanks to the linearization technique introduced in the previous chapter, all the constraints functions in $(\mathcal{P}_x^\alpha)$ are linear. Moreover, the Frank and Wolfe algorithm solves an optimization program with a linear objective function at each step.

On the other hand, the well known drawback of the Frank-Wolfe algorithm is its "slow" convergence rate. Indeed we have $f_\alpha(x,y_k) - f_\alpha(x,y_\alpha(x)) \leq O(\frac{1}{k})$ ([FW56] and [DH78]).

Anyway in a recent paper ([JLJ14]) Simon Lacoste-Julien and Martin Jaggi showed that the Frank-Wolfe variant with *Away Steps* converges with a geometric rate ($O(\varrho^{-k})$) for any *strongly convex* objective function if furthermore the feasible region is a polytope.

This is indeed our case: the strong convexity requirement is always met due to the properties of $h$, as well as the geometric properties of the feasible region.

The Away Steps variant was proposed in [Wol70]. While in the standard Frank-Wolfe algorithm the direction at each iteration $k$ is given by the vertex $z_k$, the Away Steps variant allows us to exploit also a subset of the vertices which have already been considered in the previous iterations. To do so, a data structure containing a subset of those

vertices $\{z_i | i \leq k\}$ is mantained. At each iteration $k$ a new vertex could be added or there could be a "drop step", which causes the deletion of one or more vertices. After computing the standard Frank-Wolfe direction $z_k - y_k$, the algorithm chooses the best descent direction available in the data structure, i.e. $z_1 - y_k$, is better than $z_2 - y_k$ if

$$\langle \nabla_y f_\alpha(x, y_k), z_1 \rangle \leq \langle \nabla_y f_\alpha(x, y_k), z_2 \rangle.$$

It is easy to check that the Away Steps variant enjoys the same good properties of the standard Frank-Wolfe algorithm and therefore we can use a Truncated Away Steps Algorithm correctly.

Summarizing, exploiting a Frank and Wolfe method (in its Away Steps variant) and truncating it when the error gets under our treshold $\varepsilon$, we get an $\varepsilon$ approximated solution of $(\mathcal{P}_x^\alpha)$ and one vector $\lambda \in \Lambda_\alpha^\varepsilon(x)$. In addition, the algorithm has a good theoretical behaviour due to its geometric rate of convergence and to the fact that the iteration cost could be ammortized exploiting linear programming solvers.

### 3.4.2 Unconstrained Minimization Method

We introduced Algorithm 3.1 mainly because of its linearization property. Although its theoretical "good behaviour" practical performances could be very different.

Beyond the Frank-Wolfe approach, another possibility that could lead to good performances is the *Uncostrained Minimization Method*.

Given $t > 0$, consider the problem

$$\min\{f_\alpha(x, y) - \frac{1}{t} \sum_{i=1}^{m} \log(-g_i(x, y)) : \ y \in D, \ g_i(x, y) < 0 \ i = 1, 2, \ldots, m\}. \quad \text{(CP(t))}$$

Since $G \equiv \{y \in \mathbb{R}^n | \ g_i(x, y) < 0 \ i = 1, 2, \ldots, m\}$ is an open convex set, if $\overline{y} \in G \cap D$ solves (CP(t)), either

$$\nabla_y [f_\alpha(x, \overline{y}) - \frac{1}{t} \sum_{i=1}^{m} \log(-g_i(x, \overline{y}))] = 0,$$

or $\overline{y}$ is a boundary point of $G \cap D$. Optimization algorithms can benefit from this fact. As an example, in the case when $D = \mathbb{R}^n$, (CP(t)) can be solved as an unconstrained optimization problem searching for $y \in \mathbb{R}^n$ satisfying

$$\nabla_y [f_\alpha(x, \overline{y}) - \frac{1}{t} \sum_{i=1}^{m} \log(-g_i(x, \overline{y}))] = 0.$$

Enhanced techniques are possible also in the case when $D$ is given by linear equalities and in general when it is a polyhedron strengthening the hypothesis accordingly.

The solution of (CP(t)) is linked to the $\varepsilon$KKT($\eta$) conditions by the following theorem.

**Lemma 3.8**
*If $y^*$ solves (CP(t)), then there exist $\lambda \in \mathbb{R}^m_+$, $\mu \in \mathbb{R}^{r_1}_+ \times \mathbb{R}^{r-r_1}$ and $\eta \in int\, \mathbb{R}^m_+$ such that $(y^*, \lambda, \mu)$ satisfies the $\varepsilon$KKT($\eta$) conditions with $\varepsilon = \frac{m}{t}$.*

*Proof.* Since $-log(-(\cdot))$ is a convex increasing function, its composition with $g_i(x, \cdot)$ preserves the convexity. Since the Slater's constraint qualification holds, the KKT conditions for (CP(t)) have to be satisfied by some $\mu \in \mathbb{R}^{r_1}_+ \times \mathbb{R}^{r-r_1}$:

$$\nabla_y f_\alpha(x, y^*) + \sum_{i=1}^{m} \frac{\nabla_y g_i(x, y^*)}{-t g_i(x, y^*)} + \sum_{j=1}^{r} \mu_j \nabla h_j(y^*) = 0 \tag{3.9a}$$

$$\mu_j h_j(y^*) = 0 \quad j = 1, 2, \ldots, r_1 \tag{3.9b}$$

$$y^* \in G \cap D. \tag{3.9c}$$

Defining $\lambda_i = \frac{1}{-t g_i(x, y^*)}$ for $i = 1, 2, \ldots, m$, we are done. In fact ($\varepsilon$KKT$_x(\eta)$1), ($\varepsilon$KKT$_x(\eta)$3), ($\varepsilon$KKT$_x(\eta)$5) are satisfied respectively because of (3.9a), (3.9b) and (3.9c). In addition we have that $\lambda_i g_i(x, y) = \frac{1}{-t}$ and therefore also ($\varepsilon$KKT$_x(\eta)$2) is satisfied since

$$\sum_{i=1}^{m} \lambda_i g_i(x, y) = -\frac{m}{t}. \tag{3.10}$$

Finally, $y^* \in G$ guarantees ($\varepsilon$KKT$_x(\eta)$6) and the existence of $\eta \in$ int $\mathbb{R}^m_+$ such that ($\varepsilon$KKT$_x(\eta)$4) holds.

□

It is straightforward that due to this lemma, we can provide suitable $\varepsilon$ approximated solutions solving a single problem in the form of (CP(t)) with $t = m/\varepsilon$, taking advantage of the lack of constraints.

#### 3.4.2.1 Historical Notes

The name comes from the fact that it reduces a constrained problem in the form of ($\mathcal{P}^\alpha_x$) to unconstrained optimization problems, except for the linear constraints which define $D$ (see Assumption 2.1). Moreover, in turn they could be partially absorbed into the objective function obtaining

$$\min_{Hy=0} f_\alpha(x, y) - \frac{1}{t} \sum_{i=1}^{m} \log(-g_i(x, y)) - \frac{1}{t} \sum_{j=1}^{r_1} \log(-h_j(y)),$$

where $H \in \mathbb{R}^{n \times r - r_1}$.

Usually, the Unconstrained Minimization Method, is not used as it is to solve exactly an optimization problem through a unique unconstrained problem. Indeed it is used to solve a sequence of problems in the form (CP(t)) for increasing values of $t$, exploiting every time the previous solution point as the starting point. Since the solutions of $(CP(t_k))$ are $\frac{m}{t_k}$ approximated solutions of $(\mathcal{P}_x^\alpha)$, for $t_k \to \infty$ the method converges to an optimal solution. The $\frac{m}{t_k}$ approximated solutions are internal points of the feasible region: for this reason the sequence of points solution of $(CP(t_k))$ is called *central-path*. This kind of methods, was originally proposed by Fiacco and McCormick in the 1960s. These methods are also called *Barrier Methods* or *Sequential Unconstrained Minimization Technique* (SUMT) and belong to the family of *Interior Point Methods*.

# Chapter 4

# Equilibrium Prices Forecasting in Cloud Computing

In this chapter we describe a concrete equilibrium problem which can be formulated in the (EP) format, and we apply the methods proposed in the previous chapters aiming at solving it. The first section describes the domain of the application (Cloud Computing). In the second section we give a description of the problem from a Game Theoretic point of view. In the third section we briefly discuss the assumptions and the application of the descent methods. In the last section we show some numerical results.

## 4.1 Domain Presentation

### A Cloud Computing Application

According to Armbrust at al. [AFG+10] "cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services". Therefore, cloud computing involves computational resources being sold on demand (*utility computing*) as a service. These computational resources can belong to different levels of abstraction from software (software as a service or SaaS) to hardware (platform or infrastructure as a service or PaaS/IaaS).

The impact of cloud computing in modern economy is huge. Selling computational resources on demand allows converting capital expenses to operating expenses (CapEx to OpEx) and cut down the over/under provisioning problem. For these reasons currently the cloud computing offer is becoming day by day wider.

As cloud-based services increase and become more dynamic, new challenges and problems for the management of the provider systems arise. As a result Game Theory have

been used extensively to model the competitive behaviour of the service providers and users, see for example Shue et al. [SFS13] and Ardagna et al. [APP11].

In this chapter we consider the point of view of an IaaS provider which sells virtual machines and communication bandwidth to the users. The users act as if their virtual machines were real computing platforms with memory and computational capacity. On the other side, the provider assigns each virtual machine to a physical server which runs a virtualization software emulating the behaviour of the virtual machines. The virtual machines belonging to the same user can send and receive messages, which in turn are transmitted between the physical machines over the provider's network. In our scenario the provider has already allocated the virtual machines of his tenants and has a stochastic knowledge of the communication bandwidth intensity that the users will buy as a function of the transmission price. The objective of the provider is to choose proper network routing and transmission prices in order to achieve an efficient allocation and high revenues. On the other hand, the users are interested in accessing the best communication bandwidth at the most convenient price. The system is modeled as a game and the algorithms developed in the previous chapters are applied to find an equilibrium.

### System Model

We can model the server network as a graph $G = (N, A)$, whose nodes are the physical machines. We assume to have $|N| = m$ physical machines denoted as $PM_i$ for $i = 1, \dots, m$. $T$ is the number of tenants, $VM(i)$ is the set of virtual machines bought by tenant $i$. The generic virtual machine is denoted as $VM_j$.

Each virtual machine has been allocated on one physical machine and has associated a parameter $\mu_j$ that represents its computational capacity. We can think of $\mu_j$ as the mean number of data that the virtual machine can can process in the unit of time.

For each pair of communicating virtual machines $VM_s, VM_r \in VM(i)$, $\lambda_{sr}$ denotes the transmission bandwidth bought for communication from $VM_s$ to $VM_r$. We can think of $\lambda_{sr}$ as the maximum number of data that $VM_s$ can send to $VM_r$ in the unit of time. $K$ is the number of communicating machine pairs and $\Lambda(i)$ is the set of communication pairs associated to tenant $i$.

The arc $(i, j) \in A$ has associated a capacity, denoted as $x_{ij}^k$, that is reserved for the transmission of data of the pair $k$. The total capacity of the arc $(i, j)$ is denoted by $u_{ij}$. The unitary price paid for the transmission $k$ is denoted by $c_k$. In addition, we denote the cost for routing a unit of flow through the arc $(i, j)$ by $b_{ij}$.

We have the following quantities:

- $\sum_{(i,j)\in A} \sum_{k=1}^{K} b_{ij} x_{ij}^k$, the total routing cost in the server network.

- $\sum_k \lambda_k c_k$, the income of the provider.
- $\sum_k x_{ij}^k$, the total reserved capacity of the arc $(i, j)$.
- $u_{ij} - \sum_k x_{ij}^k$, the free capacity of the arc $(i, j)$.

## 4.2 Game Theoretic Formulation

We model the behaviour of the tenants and the service provider as a game. The service provider's strategy concerns the allocation of the flows in the network $(x_{ij}^k)$ in order to satisfy the demands of the tenants and the transmission prices $(c_k)$. A tenant's strategy, instead, concerns the deciding of the transmission demand $(\lambda_k)$ between his virtual machines. Since we assume the virtual machines to be already assigned to the physical machines, each $\lambda_k$ has associated an origin-destination pair of physical machines.

### Provider's Best Response

Formally, given vector of transmission demands $\lambda \in \mathbb{R}^K$, the response of the service provider is

$$
\begin{aligned}
\min \sum_{(i,j)\in A} \sum_{k=1}^{K} b_{ij} x_{ij}^k - \sum_k r_k \lambda_k c_k & \\
\sum_{j\in BS(i)} x_{ji}^k - \sum_{j\in FS(i)} x_{ij}^k = \beta_i^k \quad & i = 1, 2, \ldots, m,\ k = 1, 2, \ldots, K \\
\sum_{k=1}^{K} x_{ji}^k \leq u_{ij} \qquad & (i, j) \in A \\
\sum_{k=1}^{K} \frac{c_k}{K} \leq B & \\
x_{ij}^k \geq 0 \qquad & (i, j) \in A,\ \ k = 1, 2, \ldots, K \\
0 \leq c_k \leq \bar{c} \quad & k = 1, 2, \ldots, K
\end{aligned}
\tag{PR}
$$

where

$$
\beta_i^k = \begin{cases} -\lambda_k & \text{if } i \text{ is the origin of flow } k \\ \lambda_k & \text{if } i \text{ is the destination of flow } k \\ 0 & \text{otherwise} \end{cases},
$$

$BS(i)$ and $FS(i)$ denote respectively the backward and forward star of node i and $r_k$ is a positive constant that can be used to model particular server's policies (e.g, QoS classes).

The first set of constraints provides a correct allocation of the flows $\lambda_k$ in the network, in order to satisfy the demands. The subsequent capacity constraints guarantee that

the arc capacity $u_{ij}$ is not exceeded. The first term of the objective function requires the routing to be of minimum cost with respect to the prices $b_{ij}$. Notice that the set of constraints relative to the variables $x_{ij}^k$ requires the solution of an embedded *multicommodity minimum cost flow problem* in which $K$ is the number of commodities.

The provider prices are constrained to be lower or equal to $\overline{c}$ and the mean price $\sum_{k=1}^{K} \frac{c_k}{K}$ cannot exceed the fixed positive constant $B$.

#### Tenant's Best Response

The response of each tenant depends only on the prices. This is because he does not have knowledge of the network. The purpose of the tenant is to minimize his expenses. In addition, in order to guarantee a good communication between his virtual machines, the tenant tries to maximize the minimum bandwidth. Formally, given the provider strategy $c \in \mathbb{R}^K$, the response of the tenant $i$ is given by

$$\begin{aligned} \min \quad & \sum_{k \in \Lambda(i)} \lambda_k c_k - \lambda_{min} \\ & \sum_{k \in S(j) \cap \Lambda(i)} \lambda_k \leq \mu_j \quad j \in VM(i) \\ & \lambda_k \geq \lambda_{min} \quad k \in \Lambda(i) \\ & \underline{W}_k \leq \lambda_k \leq \overline{W}_k \quad k \in \Lambda(i) \end{aligned} \tag{TR}$$

where $S(j)$ denotes the set of pairs that have the virtual machine $VM_j$ as destination.

$\lambda_{min}$ is an auxiliary variable: due to the constraints $\lambda_k \geq \lambda_{min}$, the best value it can get whenever $\lambda_k$'s are given is always the value of the minimum $\lambda_k \in \Lambda(i)$. The constraint $\sum_{k \in S(j) \cap \Lambda(i)} \lambda_k \leq \mu_j$ requires that the intensity of data transmitted in the time unit to machine $VM_i$ does not exceed the quantity of data it can compute. $\underline{W}_k, \overline{W}_k \in \mathbb{R}_+$ are lower and upper bound on the bandwidth allocation to the tenant.

### 4.2.1 Enhanced Formulation

#### Bounding Network Delay

The game formulated above would be perfect to test **??** and **??**, aiming at finding equilibria. Anyway, we need nonlinear constraints in order to test properly algorithm Algorithm 2.1 and Algorithm 2.2,. Therefore, we decided to modify the provider response (PR) adding a constraints on the maximum delay to cross arcs in the network. The delay function we consider is the one introduced by the U.S. Bureau of Public Roads [oPR64], defined as

$$D_{ij}(x) = f_{ij}[1 + 0.15(\frac{\sum_k x_{ij}^k}{u_{ij}})^4],$$

where $f_{ij}$ is the free flow travel time on link $(i,j)$ per unit of time. $D_{ij}$ extimates the mean time required to cross the arc $(i,j)$ when the utilization factor of the arc is equal to $\frac{\sum_k x_{ij}^k}{u_{ij}}$. The provider best response, we call it (PRE) (Provider Response Enhanced), becomes

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} \sum_{k=1}^{K} b_{ij} x_{ij}^k - \sum_k r_k \lambda_k c_k \\
& \sum_{j\in BS(i)} x_{ji}^k - \sum_{j\in FS(i)} x_{ij}^k = \beta_i^k \quad i = 1,2,\ldots,m,\ k = 1,2,\ldots,K \\
& \sum_{k=1}^{K} x_{ji}^k \le u_{ij} \quad (i,j) \in A \\
& D_{ij}(x) \le \overline{D}_{ij} \quad (i,j) \in A \\
& \sum_{k=1}^{K} \frac{c_k}{K} \le B \\
& x_{ij}^k \ge 0 \qquad (i,j) \in A,\quad k = 1,2,\ldots,K \\
& 0 \le c_k \le \bar{c} \quad k = 1,2,\ldots,K
\end{aligned}
\tag{PRE}
$$

where $\overline{D}_{ij}$ is a positive constant bounding the delay for crossing arc $(i,j)$.

### Software Application Scenario

All the descent methods we have described in the previous chapters implicitely require a *centralized* execution. In fact, all the players' parameters have to be accessible to the algorithm. This could be in contrast with some enviromental policies, as in cloud computing, where for security reasons is often preferred a distributed approach.

Our proposal is to apply the algorithms at the server side, with simulation purposes. Instead of computing equilibria that require all players' informations, the provider estimates the parameters of the users (e.g., from past sessions) and solve (EP). The aim is to provide the server with a stochastic knowledge of the equilibria.

In our case, all the parameters $\overline{W}_k, \underline{W}_k$ and $\mu_j$ are private informations of the tenant and hence are not known to the provider. Therefore, we suppose that the provider treats them as *random variables with known distribution*. Instead of requiring $\lambda_k$, to satisfy the constraints

$$
\begin{cases}
\lambda_k \le \overline{W}_k \\
\lambda_k \ge \underline{W}_k,
\end{cases}
\tag{4.1}
$$

it is required that the probability that the random variables $\overline{W}_k$ and $\underline{W}_k$ satisfy the constraints is greater than $\epsilon \in ]0,1[$:

$$\begin{cases} \mathbb{P}\{\lambda_k \leq \overline{W}_k\} \geq \epsilon \\ \mathbb{P}\{\lambda_k \geq \underline{W}_k\} \geq \epsilon. \end{cases} \tag{4.2}$$

In the same way, introducing the random variable $\mu_j$, we substitute the constraints

$$\sum_{k \in S(j) \cap \Lambda(i)} \lambda_k \leq \mu_j \quad j \in VM(i) \tag{4.3}$$

with

$$\mathbb{P}\{\sum_{k \in S(j) \cap \Lambda(i)} \lambda_k \leq \mu_j\} \geq \epsilon \quad j \in VM(i). \tag{4.4}$$

Optimization programs that include this kind of constraints are also known as *chance (or probabilistic) constrained programs*. Chance constrained programming have been introduced by Charnes and Cooper [CC59] in a linear programming framework.

Due to the simple structure of the constraints (4.1), the constraints (4.2) can be handled easily.

**Definition 4.1**
Let $F_X$ be the cumulative distribution function of a random variable $X : \Omega \to \mathbb{R}$ where $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space. Then given $\epsilon \in (0,1]$, the $\epsilon$ quantile of $X$ is defined as

$$q_{\epsilon X} = \arg\inf\{x \in \mathbb{R} \mid F_X(x) \geq \epsilon\}.$$

Since the cumulative distribution function $F_X$ is continuous and monotone non decreasing, if $F_X$ is invertible with inverse $F_X^{-1}$, we have $q_{\epsilon X} = F_X^{-1}(\epsilon)$. Furthermore it holds that if $x \leq q_{\epsilon X}$ then $F_X(x) \leq \epsilon$.

Exploiting the quantile we can replace the constraints (4.2), which can be nonlinear, by simple box constraint as shown by the following theorem.

**Theorem 4.1**
$\lambda_k \in \mathbb{R}$ *satisfies* (4.2) *iff*

$$q_{\epsilon \underline{W}_k} \leq \lambda_k \leq q_{1-\epsilon \overline{W}_k}.$$

*Proof.* Let $F_{\overline{W}_k}$ and $F_{\underline{W}_k}$ be the cumulative distribution function of $\overline{W}_k$ and $\underline{W}_k$ respectively. The first constraint of (4.2) rewrites

$$\begin{aligned} 1 - \mathbb{P}\{\lambda_k \geq \overline{W}_k\} &\geq \epsilon \\ 1 - F_{\overline{W}_k}(\lambda_k) &\geq \epsilon \\ F_{\overline{W}_k}(\lambda_k) &\leq 1 - \epsilon. \end{aligned}$$

DEFINITION 4.1 guarantees that the above condition is satisfied by $\lambda_k$ iff $\lambda_k \leq q_{1-\epsilon\overline{W}_k}$. Similarly the second constraint of (4.2) is equivalent to $F_{\underline{W}_k}(\lambda_k) \geq \epsilon$. This condition is satisfied iff $\lambda_k \geq q_{\epsilon\underline{W}_k}$. □

Repeating the same reasoning with constraints (4.4) we obtain the equivalent theorem:

**THEOREM 4.2**
$\lambda_k \in \mathbb{R}$ *satisfies* (4.4) *iff*

$$\sum_{k \in S(j) \cap \Lambda(i)} \lambda_k \leq q_{1-\epsilon\ \mu_j}.$$

The best response of the tenant $i$ becomes

$$\begin{aligned} \min \sum_{k \in \Lambda(i)} & \lambda_k c_k - \lambda_{min} \\ \sum_{k \in S(j) \cap \Lambda(i)} & \lambda_k \leq q_{1-\epsilon\ \mu_j} \quad j \in VM(i) \\ & \lambda_k \geq \lambda_{min} \quad k \in \Lambda(i) \\ & q_{\epsilon\underline{W}_k} \leq \lambda_k \leq q_{1-\epsilon\overline{W}_k} \quad k \in \Lambda(i). \end{aligned} \tag{TRE}$$

Notice that in (TRE) all the parameters are known to the provider. Indeed $q_{1-\epsilon\ \mu_j}, q_{\epsilon\underline{W}_k}, q_{1-\epsilon\overline{W}_k}$ could be statistically extimated by the observations collected during previous sessions with the user $i$.

In what follows we will refer to the game given by (PRE) and (TRE) as the *prices forecasting game* and to the problem of finding an equilibrium as the *equilibrium prices forecasting problem* (EPFP).

## 4.3 From Theory to Application

### 4.3.1 Generalized Games

There is a very relevant difference between the game definition given in DEFINITION 1.2 and the game presented in the previous section. Indeed, looking at (PRE) we can notice that the strategy set of the provider depends on the variables of the other players. This is due to the constraints

$$\sum_{j \in BS(i)} x_{ji}^k - \sum_{j \in FS(i)} x_{ij}^k = \beta_i^k \quad i = 1, 2, \ldots, m,\ k = 1, 2, \ldots, K,$$

since $\beta_i^k$ depends on $\lambda_k$. This is in contrast with DEFINITION 1.2, since it requires that the strategy set of each player is independent of the others player's variables.

In this section we introduce the notion of *Generalized Game* and of *Generalized Nash Equilibrium Problem* (Debreu [Deb52]), which is a more general class of problems that cover our case. As we will see, due to the particular structure of the game, the GNEP can be found exploiting a Nikaido-Isoda bifunction as before: thus, we can apply the theory developed in the past chapters.

**Definition 4.2** (Generalized Game)
A generalized game $G$ is a triple $G = \langle P, \{S_p\}_{p\in P}, \{u_p\}_{p\in P}\rangle$ where

- $P = \{1, 2, \ldots, N\}$ is the set of players,
- $S_p$ is a point to set map $S_p : \mathbb{R}^{n-p} \to \mathbb{R}^{n_p}$,
- $u_p : \prod_{j=1}^{N} S_j \to \mathbb{R}$ is the payoff function of player $p$.

Clearly, the difference with Definition 1.2 lays in $S_p$. While in Definition 1.2 $S_p$ is a set meaning that the player $p$ can always play any strategy $s_p \in S_p$, in Definition 4.2 $S_p$ is a point to set map with the meaning that the strategy played by $p$ depends on the strategies $s_{-p}$ choosen by the other players:

$$s_p \in S_p(s_{-p}).$$

Hence, the best response of player $p$ to the other players' strategy $s_{-p} \in \prod_{i\neq p} \mathbb{R}^{n_i}$ is

$$\min\{u_p(s_p, s_{-p}) \mid s_p \in S_p(s_{-p})\}.$$

The Generalized Nash equilibrium problem (GNEP) consist in finding strategies $(s_1, s_2, \ldots, s_N)$ which are a Generalized Nash equilibrium, defined as follows.

**Definition 4.3** (Generalized Nash Equilibrium)
Let $G$ be a game and let

$$\mathcal{S} = \{x \in \prod_{p=1}^{N} \mathbb{R}^{n_p} |\ x_p \in S_p(x_{-p}) \quad p = 1, 2, \ldots, N\}.$$

A point $x^* \in \mathcal{S}$ is called Generalized Nash Equilibrium for the game $G$ iff

$$u_p(y, x^*_{-p}) \geq u_p(x^*) \quad \forall (y, x^*_{-p}) \in \mathcal{S}$$

holds for any player $p$.

The *Jointly Convex* situation (Rosen [Ros65]) provides an interesting case.

**Assumption 4.1** (Convexity Assumption)
*A (GNEP) is said to satisfy the Convexity Assumption if for every player $p$ and every $x_{-p} \in \prod_{i \neq p} Im\ S_i$, the payoff function $u_p(\cdot, x_{-p})$ is convex and the set $S_p(x_{-p})$ is closed and convex.*

**Definition 4.4** (Jointly Convex)
Consider a GNEP satisfying the Convexity Assumption. We say that it is *jointly convex* if there exists some closed convex set $X \in \prod_{i=1}^{N} \mathbb{R}^{n_i}$ such that we have that

$$S_p(x_{-p}) = \{x_p \in \mathbb{R}^{n_p} \mid (x_p, x_{-p}) \in X\} \quad \forall x_{-p} \in \prod_{i \neq p} \mathbb{R}^{n_i}$$

holds for any player $p$.

It is remarkable that if the sets $S_p(x_{-p})$ are given by inequalites in the form

$$S_p(x_{-p}) = \{x_p \in \mathbb{R}^{n_p} \mid g_p(x_p) \leq 0,\ g(x_p, x_{-p}) \leq 0\}$$

for some $g : \prod_{i=1}^{N} \mathbb{R}^{n_i} \to \mathbb{R}^m$, $g_p : \mathbb{R}^{n_p} \to \mathbb{R}^{m_p}$, and , $m, m_1, \ldots, m_P \geq 1$ then a GNEP satisfying the Convexity Assumption is jointly convex iif $g, g_1, \ldots, g_P$ are convex over their respective domain.

The interesting thing about jointly convex (GNEP)s it that can be reformulated as an (EP) exploiting the Nikaido Isoda bifunction (1.4). In fact if $NI$ is the Nikaido Isoda bifunction and $X$ is the convex set of strategies as in Definition 4.4, then solving the (GNEP) is equivalent to solve the following (EP):

$$\text{find } x \in X \ :\ NI(x, y) \geq 0\ \forall y \in X.$$

For further details on (GNEP) the reader can refer to the detailed survey by Facchinei and Kanzow [FK07].

### 4.3.2 Application of the Descent Methods

We introduced the Equilibrium Prices Forecasting problem in order to have a concrete setting to test the algorithms described in the previous chapters. In this paragraph we show that the (GNEP) associated with the EPFP problem matches all our assumptions.

#### Joint Convexity

Firstly, notice that the (GNEP) is jointly convex. Indeed, both the provider and the tenants' payoff functions (see (PRE) and (TRE)) are affine with respect to their optimization variables. In addition, all the constraints in (PRE) and (TRE) are given by

inequalities or equalities and are convex with respect to all the variables. Hence, the corresponding (GNEP) satisfies the Convexity Assumption and is jointly convex.

#### Assumptions on the Nikaido Isoda Bifunction

The Nikaido Isoda bifunction $NI : \mathbb{R}^{K|A|+K+K} \times \mathbb{R}^{K|A|+K+K} \to \mathbb{R}$ associated with (EPFP) is

$$NI(X,Y) = f_{prov}(Y_x, X_\lambda, Y_c) - f_{prov}(X_x, X_\lambda, X_c) + \sum_{t=1}^{T} f^t_{ten}(X_x, Y_\lambda, X_c) - f^t_{ten}(X_x, X_\lambda, X_c)$$

where $f_{prov}$ is the payoff function of the provider, i.e.,

$$f_{prov}(x, \lambda, c) = \sum_{(i,j)\in A} \sum_{k=1}^{K} b_{ij} x^k_{ij} - \sum_k r_k \lambda_k c_k,$$

and $f^t_{ten} \to \mathbb{R}$ is the payoff function of the tenant $t$, i.e.,

$$f^t_{ten}(x, \lambda, c) = \sum_{k\in\Lambda(t)} \lambda_k c_k - \lambda^t_{min}$$

and $X_x \in \mathbb{R}^{K|A|}, X_\lambda \in \mathbb{R}^K, X_c \in \mathbb{R}^K$ are the components of the vector $X$ related respectively to the variables $x^k_{ij}$, $\lambda_k$ and $c_k$ ($Y_x \in \mathbb{R}^{K|A|}, Y_\lambda \in \mathbb{R}^K, Y_c \in \mathbb{R}^K$ are defined accordingly).

$NI$ is composed by affine functions. Obviously, it is differentiable and $NI(X, \cdot)$ is convex for any $X \in \mathbb{R}^{K|A|+K+K}$. In addition, it can be proven (see [BP14]), that c-monotonicity is guaranteed if $NI(\cdot, Y)$ is concave for any $Y \in \mathbb{R}^{K|A|+K+K}$. This is indeed the case, since $NI(\cdot, Y)$ is affine.

## 4.4 Numerical Results

The platform we decide to adopt is MATLAB R2011a. We implement the following algorithms:

**BCP:** Algorithm 1.2,

**BCP-ET:** Algorithm 1.3,

**COAP:** Algorithm 2.1,

**COAP-ET:** Algorithm 2.2.

The implementation includes an OO framework for optimization problems, a random generator of instances of (EPFP) and MATLAB coding of several nonlinear optimization algorithms.

In order to test properly the algorithms, we decided to solve their inner problems ($P_x^\alpha$) and ($\mathcal{P}_x^\alpha$) using the same procedure: the Sequential Unconstrained Minimization Technique described in subsection 3.4.2. We have seen that the SUMT solves a sequence of problems in the form (CP($t_k$)) with $t_k \to +\infty$. In our implementation $t_k = 10 \cdot 20^k$ and the central path problem (CP($t_k$)) is solved via the MATLAB function `fmincon` exploing the *Successive Quadratic Programming* method. COAP and BCP require an optimal solution, so the choice $10^{-6}$ represent the threshold for "true optimality". Instead, BCP-ET and COAP-ET stop the computation whenever an $\varepsilon$ approximated solution is found. This obvioulsy impacts on the number of average iterations of the SUMT method and therefore on the average computational time.

After some preliminary tests we set parameters of the algorithm as follows:

| Parameter | Value | Sequence | Value |
|---|---|---|---|
| $\eta$ | 0.8 | $\alpha_k$ | $\frac{1}{k^2}$ |
| $\beta$ | 0.7 | $\varrho_k$ | $\frac{1}{k^5}$ |
| $\gamma$ | 0.9 | $\epsilon_k$ | $\frac{1}{k^2}$ |
| | | $\delta_k$ | $\frac{25}{k^2}$ |

Firstly we tested the algorithm on 10 instances of (EPFP) generated fully at random for fixed number of tenant (10) and network nodes (8). The table below reports the average size of the subprobem ($P_x^\alpha$).

| | Dimension |
|---|---|
| Variables | 352 |
| Nonlinear Constraints | 33 |
| Linear Inequalities | 41 |
| Linear Equalities | 80 |

The computation is ended when $h(x_k, y_\alpha(x_k)) < 10^{-2}$. The mean of the results is shown in the following table.

| Algorithm | Time | Sub-Problems | Sub-Problem Time | Jumps |
|---|---|---|---|---|
| BCP | 219s | 20 | 8.14s | 14 |
| COAP | 211s | 20 | 8.03s | 14 |
| BCP-ET | 144s | 20 | 4.6s | 15 |
| COAP-ET | 138s | 20 | 4.49s | 16 |

where

**Time** is the average completion time of the algorithm (in seconds),

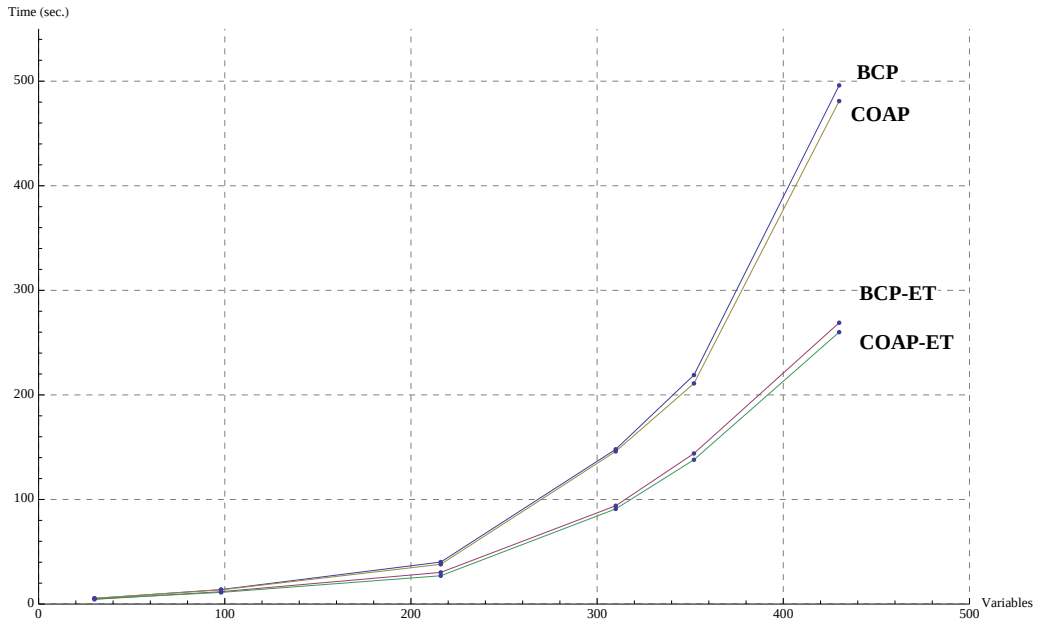
**Sub-Problems** is the number of times a problem in the form ($P_x^\alpha$) and ($\mathcal{P}_x^\alpha$) is solved,

**Sub-Problem Time** is the average time required for solving these problems,

**Jumps** is the mean number of GOTO taken due to a dissatisfaction of a condition. Remember that a GOTO is taken every time the parameter $\alpha$ or $\varrho$ are updated.

While the general behaviour of the algorithm does not change (number of jumps and of sub-problem solved), we can notice a sensible decrement in the completion time that almost halves. This is due to the reduction of the average time required to solve the inner problems. Notice that the time to solve ($P_x^\alpha$) and ($\mathcal{P}_x^\alpha$) is nearly the same. This is due to the SUMT method: indeed, all the inequality constraints are taken into the objective function during the computation and the nonlinear constraints influence minimally only the evaluation of the objective function and of its gradient.

In order to approximate the asymptotic behaviour of the algorithms we plan an "incremental size" test. We generate 6 instances of (EPFP) of different sizes, maintaining constant the ratio between number of nodes and number of flows. The following graph relates the number of variables to the completion time of the algorithms.




It can be noticed that we have an encouraging speed up in both error tolerant versions also for different problem sizes.

The error tolerant versions perform as well as the tolerance threshold for $h(x, y_\alpha(x))$ is high. The number of iterations increases when lowering the threshold and consequently the average time required to find a sub-optimal solution of the inner problem increases. The following table helps to understand the decay of the error tolerant benefits as the threshold decreases.

It has been generated solving an instance of (EPFP) of 8 nodes and 10 flows via BCP and BCP-ET. We have taken the average time needed to solve the inner problem at every iteration, considering the iterations that achieve almost the same value $h(x, y_\alpha(x))$. We can do this since the behaviour of the two algorithms is very similar. Anyway, we can always enforce this behaviour choosing proper sequence $\{\varepsilon_k\}$. Hence the ratio of the average inner problem solution time for the different iterations is an estimator of the potential speed up for different threshold.

Notice that since the behaviour of COAP/COAP-ET is comparable with the one of BCP/BCP-ET due to the SUMT algorithm, we have consired only BCP and BCP-ET in this test.

| Iteration | Avg IPT BCP | Avg IPT BCP-ET | Ratio |
|---|---|---|---|
| 1 | 7.71s | 3.30s | 2.33 |
| 2 | 7.34s | 4.38s | 1.67 |
| 3 | 7.44s | 4.48s | 1.66 |
| 4 | 7.58s | 4.48s | 1.69 |
| 5 | 7.73s | 4.54s | 1.70 |
| 6 | 8.81s | 6.07s | 1.45 |
| 7 | 9.15s | 6.48s | 1.48 |
| 8 | 9.19s | 6.46s | 1.42 |
| 9 | 9.04s | 7.55s | 1.19 |
| 10 | 9.09s | 7.64s | 1.18 |
| 11 | 8.87s | 7.62s | 1.16 |

where

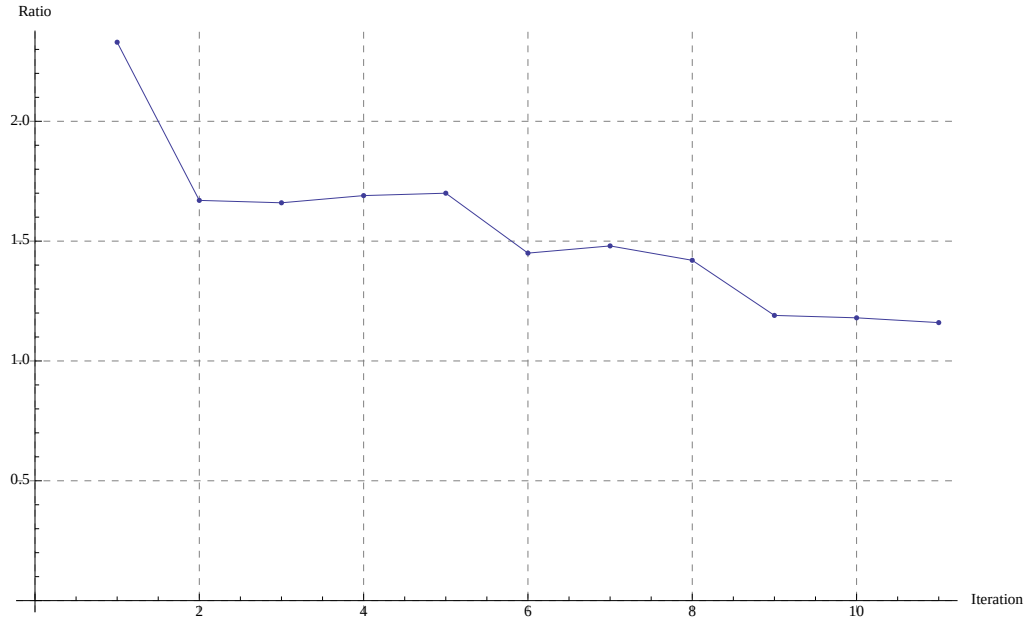
**Iteration** is the number of iterations,

**Avg IPT BCP** is average time required by BCP for solving the inner problem,

**Avg IPT BCPET** is average time required by BCPET for solving the inner problem,

**Ratio** is the ratio $\frac{\text{Avg IPT BCP}}{\text{Avg IPT BCPET}}$.

In the following graph shows how the Ratio changes as a function of the number of iterations




As expected, the improvement provided by the error tolerant approach vanish as the number of iterations increases.

Summarizing, numerical results suggest to apply the error tolerant extensions when it is required a mid-low accurancy ($h(x, y_\alpha(x))$ threshold) of the solutions of (EP) or in general when the number of iterations is not too high. But this is rather intuitive, indeed, since the speed up provided by the error tolerant approach lasts for a finite number of iterations it is useless to apply BCP-ET or COAP-ET when (EP) requires a high number of iterations. On the other hand, under this above conditions, numerical results show that the performace can improve moderately.

# Appendix A

# Point-to-Set Maps in Mathematical Programming

The following paragraph briefly reviews some results due to Hogan about the theory of points-to-set maps, that are used throughout all the thesis.

**Definition A.1** ([Hog73c])
Let $X$ and $Y$ be topological spaces. Let $\Omega$ be a point-to-set map $\Omega : X \to Y$, i.e., a function $\Omega : X \to 2^Y$. Then, we define $\Omega$ to be:

a) *open* at a point $\overline{x} \in X$ iff $\{x_k\} \subset X$, $x_k \to \overline{x}$ and $\overline{y} \in \Omega(\overline{x})$ imply the existence of an integer $m$ and a sequence $\{y_k\} \subset Y$ such that $y_k \to \overline{y}$ and $y_k \in \Omega(x_k)$ for $k \geq m$.

b) *closed* at a point $\overline{x} \in X$ iff $\{x_k\} \subset X$, $x_k \to \overline{x}$, $y_k \in \Omega(x_k)$ and $y_k \to \overline{y}$ imply $\overline{y} \in \Omega(\overline{x})$.

c) *continuous* at a point $\overline{x}$ iff is closed and open at $\overline{x}$.

   $\Omega$ is open, closed of continuous on $X$ if it has the corresponding property for every $x \in X$.

d) $\Omega$ is said to be *uniformly compact*(*bounded*) near $\overline{x}$ iff there is a neighbour $N$ of $\overline{x}$ such that the set $\overline{\cup_{x \in N} \Omega(x)}$ is, respectively, compact(bounded).

**Theorem A.1** ([Hog73c])
*Let $\Omega$ be a point-to-set map $\Omega : X \to Y$, and let $f : X \times Y \to [-\infty, +\infty]$ . Consider the following function $v : X \to [-\infty, +\infty]$*

$$v(x) = \sup_{y \in \Omega(x)} f(x, y).$$

*If $\Omega$ is continuous and uniformely compact at $\overline{x} \in X$ and $f$ is continuous on $\overline{x} \times \Omega(\overline{x})$, then $v$ is continuous at $\overline{x}$.*

**Theorem A.2** (Danskin [Dan67] [Hog73b])
*Let $\Omega, f, v, X, Y$ be defined as in Theorem A.1. Suppose that $\nabla_x f$ exists, $f$ and $\nabla_y f$ are continuous over $X \times Y$, $X$ is compact, and $\Omega$ is constant. Then, the directional derivative $v'(\overline{x}; d)$ exists for any $\overline{x}, d \in X$ and*

$$v'(\overline{x}; d) = \max_{y \in M(\overline{x})} \langle \nabla_x f(\overline{x}, y), d \rangle$$
$$\text{where} \quad M(\overline{x}) = \{y \in \Omega(\overline{x}) |\ v(\overline{x}) \leq f(\overline{x}, y)\}.$$

**Theorem A.3** ([Hog73c])
*Let $P$ be a point-to-set map $P : X \to Y$ defined as follows:*

$$P(x) = \{y \in Y : g(x, y) \leq 0\},$$

*where $g(x, y) : X \times Y \to [-\infty, +\infty]^m$. If each component of $g$ is lower semicontinuous on $\overline{x} \times Y$, then $P$ is closed at $\overline{x}$.*

**Theorem A.4** ([Hog73c])
*Let $P$ be a point-to-set map and $g$ a function defined as in Theorem A.3. If $Y$ is convex and normed, if each component of $g$ is continuous on $\overline{x} \times P(\overline{x})$ and $g(x, \cdot)$ is convex for each fixed $x \in X$, and if there exists $\overline{y} \in Y$ such that $g(\overline{x}, \overline{y}) < 0$, then $P$ is open at $\overline{x}$.*

**Theorem A.5** ([Hog73c])
*Let $\Omega, f, v, M$ be as defined in Theorem A.2. Suppose in addition that $Y$ is a subset of a finite-dimensional normed space, $f(x, \cdot)$ is quasiconcave in $y$ for fixed $x$ and continuous on $X \times Y$, $\Omega$ is closed on a neighborhood of $\overline{x}$ and open at $\overline{x}$, and $\Omega(\overline{x})$ is convex for each $x$ in a neighborhood of $\overline{x}$. Then, $M(\overline{x})$ is non empty and compact if and only if $M(x)$ is nonempty and uniformly compact near $\overline{x}$.*

**Theorem A.6** ([Hog73b])
*Let $Y$ be a closed convex set, and let the point-to-set maps $\Omega, M : X \to Y$ and the function $v : X \to Y$ be defined as follows:*

$$\Omega(x) = \{y \in Y \mid g(x, y) \leqq 0\},$$
$$v(x) = \sup_{y \in \Omega(x)} f(x, y),$$
$$M(x) = \{y \in Y \mid v(x) \leq f(x, y)\},$$

*where $g(x, \cdot)$ and $-f(x, \cdot)$ are convex. In addition, let $\overline{x} \in X$ and suppose that $g, f$ are continuous on $N_{\overline{x}} \times Y$, where $N_{\overline{x}}$ is a neighborhood of $\overline{x}$. Let $U(x)$ be the set of Lagrangian multipliers associated with the problem which defines $v(x)$. If $M(\overline{x})$ is nonempty and bounded and there is a point $\overline{y} \in Y$ such that $g(\overline{x}, \overline{y}) < 0$, then $U$ and $M$ are nonempty and bounded on a neighborhood of $\overline{x}$ and $U$ is closed at $\overline{x}$.*

# Appendix B

# Nonlinear Programming

### Convexity

**Definition B.1** (Convex Set)
A set $C \subseteq \mathbb{R}^n$ it is said to be *convex* if given any $x, y \in C$

$$\lambda x + (1-\lambda)y \in C \quad \forall \lambda \in [0,1].$$

**Definition B.2** (Convex Function)
Let $C$ be a convex subset of $\mathbb{R}^n$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is called

- *convex* over $C$ if for any $x, y \in C$

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \quad \forall \lambda \in [0,1].$$

- *strictly convex* over $C$ if for any $x, y \in C$

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y) \quad \forall \lambda \in [0,1].$$

- *strongly convex* over $C$ if $\exists m > 0$ such that for any $x, y \in C$

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) - \frac{1}{2} m \lambda (1-\lambda) \|x - y\|_2^2 \quad \forall \lambda \in [0,1].$$

A function $f$ such that $-f$ is convex, strictly or strongly convex is called respectively *concave*, *strictly concave* or *strongly concave*. The following implications are obvious from the above definitions:

$$f \text{ strongly convex } \Rightarrow f \text{ strictly convex } \Rightarrow f \text{ convex.} \tag{B.1}$$

**Theorem B.1**
*Let $C \subseteq \mathbb{R}^n$ be a convex set. A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is*

– *convex over a set $C$ iff for any $x, y \in C$*

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle. \tag{B.2}$$

– strictly *convex over a set $C$ iff for any $x, y \in C$*

$$f(x) > f(y) + \langle \nabla f(y), x - y \rangle.$$

– strongly *convex over a set $C$ iff $\exists m > 0$ such that for any $x, y \in C$*

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{m}{2} \|x - y\|_2^2.$$

**Optimization Problems**

Let $f$ be a function $\mathbb{R}^n \to \mathbb{R}$ and $D$ be a subset of $\mathbb{R}^n$, then we define the optimization problem (OP), denoted as

$$\min_{x \in D} f(x), \tag{OP}$$

as the problem of finding a point $\overline{x} \in \mathbb{R}^n$ satisfying

i) $\overline{x} \in D$,

ii) $f(\overline{x}) \leq f(y)\ \forall y \in D$.

The requirement i) is called *feasibility* while ii) is called *optimality*. A solution of (OP) is also called *global minimum* of (OP) in order to distinguish it from *local minima* of (OP), which are feasible points that satisfy the optimality requirement only in their neighborhoods.

**Definition B.3** (Local Minimum of (OP))
A point $x \in D$ is a *local minimum* of (P) if $\exists \varepsilon > 0$

$$f(x) \leq f(y) \quad \forall y \in B(x, \varepsilon) \cap D, \tag{B.3}$$

where $B(x, \varepsilon)$ denotes the open ball of radius $\varepsilon$ centered at $x$.

Clearly a global minimum is also a local minimum but not viceversa.

**Theorem B.2**
*Suppose that $f$ is convex over $D$. If $\overline{x}$ is a local minimum of (OP), then it is also a global minimum of (OP).*

**Theorem B.3**
*Suppose that $f$ is strictly convex over $D$. Then, if there exists at least one local minimum of (OP), it is unique.*

### Unconstrained Optimization

In this paragraph we deal with the case $D = \mathbb{R}^n$. In this case the problem (OP) is called Unconstrained Optimization Problem (UOP) and it is denoted as

$$\min_{x \in \mathbb{R}^n} f(x). \tag{UOP}$$

**Definition B.4** (Stationary Point of (UOP))
Suppose that $f$ is differentiable. Then, a point $\overline{x}$ is called Stationary Point of (UOP) if

$$\nabla f(x) = 0. \tag{B.4}$$

**Theorem B.4** (Fermat (1637))
*If $\overline{x}$ is a local minimum of (UOP), then it is a stationary point of (UOP).*

As a consequence of (B.2) and of Theorem B.4 we obtain the following theorem

**Theorem B.5**
*Suppose that $f$ is differentiable and convex over $\mathbb{R}^n$, then $\overline{x}$ solves (UOP) iff*

$$\nabla f(\overline{x}) = 0. \tag{B.5}$$

The interesting fact is that if $f$ is differentiable and convex, the problem (UOP) collapse in the problem of solving a system of equations. Therefore, in principle, any algorithm to solve nonlinear systems (e.g., Newton-Raphson) could be used.

### Constrained Optimization

In this paragraph we generalize the above results removing the assumption $D = \mathbb{R}^n$. The first results concern a convex feasible region $D$.

**Definition B.5**
Suppose that $D$ is convex and that $f$ is differentiable over $D$. Then a point $\overline{x} \in D$ is called stationary point of (OP) if

$$\langle \nabla f(\overline{x}), x - \overline{x} \rangle \geq 0 \quad \forall x \in D. \tag{B.6}$$

**Theorem B.6**
*Suppose that $D$ is convex and that $f$ is differentiable over $D$. Then if a point $\overline{x} \in D$ is a local minimum of (OP) it is a stationary point of (OP).*

As a consequence of Theorem B.2, Theorem B.6 and (B.2) we obtain the following result.

**Theorem B.7**
*Suppose that $D$ is convex and that $f$ is differentiable and convex over $D$. Then a point $\overline{x} \in D$ is a global minimum of (OP) iff is a stationary point of (OP).*

More interesting results could be obtained supposing that $D$ is explicitly described by inequalities. Let functions $f, g_i, h_j : \mathbb{R}^n \to \mathbb{R}$ be differentiable over $\mathbb{R}^n$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, r$. In addition, suppose that

$$D = \{x \in \mathbb{R}^n |\ g_i(x) \leq 0,\ h_i(x) = 0\} \tag{B.7}$$

Then, the the problem (OP) reads

$$\begin{aligned} \min\ & f(x) \\ & g_i(x) \leq 0 \quad i = 1, 2, \ldots, m \\ & h_j(x) = 0 \quad j = 1, 2, \ldots, r \end{aligned} \tag{COP}$$

and the following theorem holds.

**Theorem B.8** (Fritz-John)
*If $\overline{x}$ is a local minimum for (COP) then $\exists\ \theta, \lambda_1, \ldots, \lambda_m \geq 0$ and $\mu_1, \ldots, \mu_r \in \mathbb{R}$ such that*

$$\begin{aligned} & \theta \nabla f(\overline{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\overline{x}) + \sum_{j=1}^{r} \mu_j h_j(\overline{x}) = 0 \\ & \lambda_i g_i(\overline{x}) = 0 \quad i = 1, 2, \ldots, m \\ & g_i(\overline{x}) \leq 0 \quad i = 1, 2, \ldots, m \\ & h_j(\overline{x}) = 0 \quad j = 1, 2, \ldots, r \end{aligned} \tag{FJ}$$

*with at least one nonzero element between $\theta, \lambda_1, \ldots, \lambda_m, \mu_1, \ldots, \mu_r$.*

**Definition B.6** (Set of Active Constraints)
Given $x \in \mathbb{R}^n$, we define the (index) set of active inequality constraints $I(x)$ as

$$I(x) = \{i |\ g_i(x) = 0\}.$$

**Definition B.7** (SCQ)
Let $\overline{x}$ be a point of $\mathbb{R}^n$. If $g_i$ is convex and $h_j$ is affine for any $j$ and $i \in I(\overline{x})$, the vectors $\nabla h_j(\overline{x})$ are linearly independent and there exists $y \in \mathbb{R}^n$ such that $g_i(y) < 0$ and $h_j(y) = 0$ for any $j = 1, \ldots, r, \quad i \in I(\overline{x})$, then $\overline{x}$ satisfies the Slater constraint qualification.

**Definition B.8** (MFCQ)
Let $\overline{x}$ be a point of $\mathbb{R}^n$. If the vectors $\nabla h_j(\overline{x})$ are linearly independent and if there

exists $d \in \mathbb{R}^n$ such that

$$\begin{aligned} &\langle \nabla g_i(\overline{x}), d \rangle < 0 \quad i \in I(\overline{x}) \\ &\langle \nabla h_j(\overline{x}), d \rangle = 0 \quad j = 1, 2, \dots, r, \end{aligned}$$

then $\overline{x}$ satisfies the Mangasarian-Fromovitz constraint qualification.

Notice that $SCQ$ implies $MFCQ$.

**Theorem B.9** (Karush-Kuhn-Tucker)
*Let $\overline{x}$ be a point satisfying MFCQ constraint qualification. If $\overline{x}$ is a local minimum for* (COP)*, then there exist $\lambda_1, \dots, \lambda_m \geq 0$ and $\mu_1, \dots, \mu_r \in \mathbb{R}$ such that*

$$\begin{aligned} &\nabla f(\overline{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\overline{x}) + \sum_{j=1}^{r} \mu_j h_j(\overline{x}) = 0 \\ &\lambda_i g_i(\overline{x}) = 0 \quad i = 1, 2, \dots, m \\ &g_i(\overline{x}) \leq 0 \quad i = 1, 2, \dots, m \\ &h_j(\overline{x}) = 0 \quad j = 1, 2, \dots, r \end{aligned} \tag{KKT}$$

The above optimality necessary conditions are also called *KKT conditions* and they are a particular case of Theorem B.8. For the sake of brevity, we reported only MFCQ and SCQ but there are other assumptions on $\overline{x}$, called *constraint qualifications*, that allows us to move from Fritz-John conditions to the KKT conditions.

While in general KKT conditions are only *necessary* conditions to be a global minimum, in the particular case when $f, g_i$ are convex for $i = 1, \dots, m$ and $h_j$ are affine for $j = 1, \dots, r$, they are also *sufficient*. The $\lambda_i$ satisfying (KKT) are called *Lagrange multipliers*.

### Duality

**Definition B.9** (Lagrangian)
We define the Lagrangian function of (COP), $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^r \to \mathbb{R}$ as

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{r} \mu_j h_j(x).$$

Notice that the first KKT condition

$$\nabla f(\overline{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\overline{x}) + \sum_{j=1}^{r} \mu_j h_j(\overline{x}) = 0 \tag{B.8}$$

could be also written $\nabla_x \mathcal{L}(\overline{x}, \lambda, \mu) = 0$, with the immediate interpretation that a local minimum has to be a stationary point of the function $\mathcal{L}(\cdot, \lambda, \mu)$.

**Definition B.10** (Lagrange Dual Function)
We define the function $g : \mathbb{R}^m_+ \times \mathbb{R}^r \to \mathbb{R}$ as the optimum value of the unconstrained minimization problem of minimizing $\mathcal{L}(\cdot, \lambda, \mu)$ over $\mathbb{R}^n$, i.e.

$$g(\lambda, \mu) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \mu).$$

**Theorem B.10**
*Given $\lambda \in \mathbb{R}^m_+$ and $\mu \in \mathbb{R}^r$, let $x^*$ be the solution of (COP). Then, it holds*

$$g(\lambda, \mu) \leq f(x^*).$$

**Definition B.11**
We define the *Lagrange dual problem* as the constrained optimization problem given by

$$\max\{g(\lambda, \mu) : \ \lambda \in \mathbb{R}^m_+, \quad \mu \in \mathbb{R}^r\}. \tag{LD}$$

**Theorem B.11** (Strong Duality)
*Suppose that $f, g_i$ are convex and $h_j$ are affine for any $i, j$. If $x^*$ solves (COP) and satisfies SCQ and, in addition $(\lambda^*, \mu^*)$ solves (LD), then*

$$g(\lambda^*, \mu^*) = \mathcal{L}(x^*, \lambda^*, \mu^*) = f(x^*).$$

*and the vector $(x^*, \lambda^*, \mu^*)$ the satisfies KKT conditions.*

# Bibliography


[AFG+10] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Communications of the ACM*, 53:50–58, 2010.

[APP11] D. Ardagna, B. Panicucci, and M. Passacantando. A game theoretic formulation of the service provisioning problem in cloud systems. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 177–186, New York, NY, USA, 2011. ACM.

[BCP09] G. Bigi, M. Castellani, and M. Pappalardo. A new solution method for equilibrium problems. *Optimization Methods and Software*, 24:895–911, 2009.

[BCPP13] G. Bigi, M. Castellani, M. Pappalardo, and M. Passacantando. Existence and solution methods for equilibria. *European Journal of Operational Research*, 227:1–11, 2013.

[BP12] G. Bigi and M. Passacantando. Gap functions and penalization for solving equilibrium problems with nonlinear constraints. *Computational Optimization and Applications*, 53:323–346, 2012.

[BP14] G. Bigi and M. Passacantando. Twelve monotonicity conditions arising from algorithm for equilibrium problems. *Optimization Methods and Software*, page DOI:10.1080/10556788.2014.900552, 2014.

[CC59] A. Charnes and W. W. Cooper. Chance constrained programming. *Management Science*, 6(1):73–79, 1959.

[Cla87] F.H. Clarke. *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1987.

[Dan67] J. M. Danskin. *The Theory of Max Min*. Springer-Verlag, 1967.

[Deb52] G. Debreu. A social equilibrium existence theorem. *Proc Natl Acad Sci*, 38:886–893, 1952.

[DH78] J. C. Dunn and S. Harshbarger. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62:432–444, 1978.

[FK07] F. Facchinei and C. Kanzow. Generalized nash equilibrium problems. *4OR*, 5(3):173–210, 2007.

[Fuk92] M. Fukushima. Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. *Mathematical Programming*, 53:99–110, 1992.

[FW56] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Res. Logis. Quart.*, 3:95–110, 1956.

[Hog73a] W. W. Hogan. Applications of a general convergence theory for outer approximation algorithms. *Mathematical Programming*, 5:151–168, 1973.

[Hog73b] W. W. Hogan. Directional derivatives for extremal-value functions with applications to the completely convex case. *Operations Research*, 21:188–209, 1973.

[Hog73c] W. W. Hogan. Point-to-set maps in mathematical programming. *SIAM Review*, 15:591–603, 1973.

[JLJ14] M. Jaggi and S. Lacoste-Julien. An affine invariant linear convergence analysis for Frank-Wolfe algorithms. *arXiv:1312.7864v2*, 2014.

[LPS13] D. Di Lorenzo, M. Passacantando, and M. Sciandrone. A convergent inexact solution method for equilibrium problems. *Optimization Methods and Software*, DOI:10.1080/10556788.2013.796376, 2013.

[Mas00] G. Mastroeni. On auxiliary principle for equilibrium problems. Technical report, Department of Mathematics of Pisa University, 2000.

[Mas03] G. Mastroeni. Gap functions for equilibrium problems. *Journal of Global Optimization*, 27:411–426, 2003.

[NI55] H. Nikaido and K. Isoda. Note on noncooperative convex games. *Pac J Math*, 5:807–815, 1955.

[oPR64] Bureau of Public Roads. *Traffic Assignment Manual. U.S. Dept. of Commerce, Urban Planning Division.* Washington DC, 1964.

[Ros65] J.B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33:520–534, 1965.

[SFS13] D. Shue, M.J. Freedman, and A. Shaikh. Fairness and isolation in multi-tenant storage as optimization decomposition. *SIGOPS Oper. Syst. Rev.*, 47(1):16–21, January 2013.

[Wol70] P. Wolfe. Convergence theory in nonlinear programming. In J Abadie, editor, *Integer and Nonlinear Programming*, pages 1–23. North-Holland, 1970.

[ZM94] D.L. Zhu and P. Marcotte. An extended descent framework for variational inequalities. *Journal of Optimization Theory and Applications*, 80:349– 366, 1994.