University of Padova

Department of Mathematics "Tullio Levi-Civita" Master in Mathematics

Master thesis

A derivative-free method for bilevel optimization

Student
Matia Bojovic
Serial number: **2056746**

Supervisor
Francesco Rinaldi

Co-Supervisor
Damiano Zeffiro

Academic year 2022-2023
15/12/2023

Contents

A Derivative-free method for Bilevel Optimization

Matia Bojovic

December 2023

Abstract

We address bilevel programming problems when the lower-level problem is solved inexactly, and it is supposed to have access to an approximation of the lower-level minimizer up to a bounded numerical error, through a black-box oracle.
The following thesis represents a first attempt to tackle a mathematical problem that may be, in general, difficult to deal with. Even though in literature there are common assumptions that allow us to work with lower-level minimizers which may not be unique, we will assume uniqueness just to make reading a little easier.
Our methods will deal with both smooth and potentially nonsmooth objectives. Our main goal is to design algorithms that perform optimization at the upper-level, using derivative-free methods. The analysis will first concern the case where we have only box constraints. This case may be easy when compared to nonlinear constraints, but one of the reasons is that a good understanding of the box-constrained case seems to be necessary to tackle the more general one. Also, as can be seen from the current thesis, box constraints, together with the framework we are dealing with, already introduce several nontrivial issues and a considerable amount of technical details. Further in the thesis, we will investigate more general constraints. We will show some numerical results to exhibit the effectiveness of our approaches.

Keywords. Bilevel programming, derivative-free methods, black-box oracle, box and nonlinear constraints.

Chapter 1

Introduction

Bilevel optimization has been subject of increasing interest, thanks to its applications to hyperparameter tuning for machine learning algorithms and meta-learning together with the development of the computational power of contemporary machines, through the use of GPUs in particular.
In this work, we are interested in the following bilevel optimization problem:

$$\min_{(x,y)\in\mathcal{F}\times\mathcal{G}} f(x,y) \quad \text{s.t.} \quad y \in \arg\min_{z\in\mathcal{G}} \mathrm{g}(x,z) \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be at least continuous and $g : \mathbb{R}^{n+p} \to \mathbb{R}$ is such that the lower-level problem has a unique solution *y(x)* for any given $x \in \mathcal{F}$, where n, p are natural numbers.
Here, we explore derivative-free optimization (DFO) algorithms, which simply employ the objective value itself instead of derivatives of the upper-level objective function. The existence of a black-box oracle that can approximate $\tilde{y}(x)$ of *y(x)* for every given $x \in \mathcal{F}$ will be a crucial supposition. We are particularly interested in direct-search approaches, which sample the objective at well-selected places rather than building models of it.

Chapter 2

Background and Preliminaries

The following represents an attempt to introduce the main assumptions we will consider in the thesis, all along.
Depending on the specific framework we are dealing with, which will depend on the constraints rather than the amount of regularity assumed, more assumptions will be explored.
As we said before, we will assume the unicity of the lower-level minimizer, which is common and widely adopted in the context of bilevel programming.

Assumption 1. *For any x, the minimizer y(x) of the lower problem is unique.*

Under Assumption 1, the bilevel optimization problem in (1.1) can then be rewritten as

$$\min_{x \in \mathcal{F}} \quad \mathrm{F}(x) := f(x, y(x)) \tag{2.1}$$

In practice, however, one must typically use an iterative procedure to compute *y(x)*, and thus cannot expect to readily have an exact value of *y(x)*, but rather some approximation. More explicitly, we will use the following assumption.

Assumption 2. *For all $x \in \mathcal{F}$ we can compute an approximation $\tilde{y}(x)$ of y(x) such that*

$$\|\tilde{y}(x) - y(x)\| \leq \epsilon \tag{2.2}$$

The assumption above looks reasonable and not restrictive at all.
In the following, we will seek theoretical guarantees for our algorithms and a more restrictive assumption on the numerical error allowed will be necessary.
Briefly speaking, the approximation of the lower-level minimizer has to decrease geometrically to ensure convergence of our algorithms to stationary points of the problem in (1.1). Later on, in the next chapter, details will be given.

Assumption 3. *The function f is lower bounded by* f_{low}.

Assumption 4. *The function f is Lipschitz continuous with respect to y of Lipschitz constant* L_f *.*

While the remaining assumptions introduced in this section are not always needed, in the rest of this manuscript we always assume that Assumptions 1, 2, 3, and 4 hold.

Assumption 2 together with Assumption 4 imply that $\tilde{\mathrm{F}}(x) := f(x, \tilde{y}(x))$ is an approximation of *F(x)* with accuracy $\mathrm{L}_f\epsilon$. Indeed,

$$\|\tilde{F}(x) - F(x)\| = \|f(x, \tilde{y}(x)) - f(x, y(x))\| \leq \mathrm{L}_f \|\tilde{y}(x) - y(x)\| \leq \mathrm{L}_f\epsilon \qquad (2.3)$$

Some regularity on the true objective F will always be necessary for our analysis. We consider both the differentiable and the potentially nondifferentiable setting.

Assumption 5. *The function F is Lipschitz continuous with constant* L_F*.*

Assumption 6. *The function F is continuously differentiable with Lipschitz continuous gradient of Lipschitz constant* L*.*

Chapter 3

The box constrained case

In this chapter, we will first introduce a direct-search approach for bilevel optimization when we deal with box constraints for the upper variable, both in the smooth and nonsmooth case, i.e. respectively, under Assumptions 6 or 5.
Further in this thesis, nonlinear constraints will be considered in the analysis.
In this section, the feasible set of the problem in (2.1) is given by
$\mathcal{F} = \{x \in \mathbb{R}^n \ s.t. \ l \leq x \leq u\}$, where $l,\ u \in \mathbb{R}^n$, with $l < u$. We allow the case where some of the variables are possibly unbounded by permitting both $l_i = -\infty$ and $u_i = +\infty$ for some $i \in \{1, ...n\}$.

3.1 Smooth objective

In this subsection, we propose a class of derivative-free algorithms for bilevel optimization in the case where the true function F is a continuously differentiable function, in the case that some of (or all) the variables are bounded. In particular, we will assume Assumption 6 holds.
It may be wise to first focus on this case, where some regularity is provided, as we will focus on the nonsmooth case in the following sections.
The main idea behind our approach is performing suitable samplings of the objective function along the coordinate directions.
We define a stationary point of problem in (2.1) as a feasible point $\bar{x}$ that satisfies the following first-order necessary optimality condition:

$$\nabla F(\bar{x})^T (y - \bar{x}) \geq 0 \quad \forall y \in \mathcal{F} \tag{3.1}$$

meaning there is no feasible descent direction.
We recall that any stationary point $\bar{x}$ of $\mathcal{F}$ ($\bar{x} \in \mathcal{F}$), we have that

$$\nabla^{red} F(\bar{x}) = 0 \tag{3.2}$$

where the reduced gradient $\nabla^{red}F(x)$ is defined as follows

$$\nabla_i^{red}F(x) = \begin{cases} \max\{\nabla_i F(x), 0\} & \text{if } \mathrm{x}^i = u^i \quad (3.4) \\ \min\{\nabla_i F(x), 0\} & \text{if } \mathrm{x}^i = l^i \quad (3.5) \\ \nabla_i F(x) & \text{otherwise} \quad (3.6) \end{cases}$$

In this framework, the choice of the coordinate directions as explorable directions in the algorithm we will introduce fits well, as it allows us to cope with the presence of box constraints. This can be easily derived from the optimality conditions (3.1). In fact, if $\bar{x}$, feasible point, is not a stationary point of F, then there must exist a feasible point y and an integer $h \in \{1, ..., n\}$ such that $\nabla_h F(\bar{x})^T (y - \bar{x})^h < 0$. If $\bar{\alpha} = (y - \bar{x})^h > 0$, then, recalling that $\mathcal{F}$ is defined by box constraints, we have

$$\bar{\alpha} \nabla F(\bar{x})^T e_h < 0, \qquad \bar{x} + \bar{\alpha} e_h \in \mathcal{F}$$

The continuity of ∇F and the convexity of the feasible set $\mathcal{F}$ imply that there exists a positive value $\bar{\alpha}$ such that:

$$F(\bar{x} + \alpha e_h) < F(\bar{x}), \qquad \bar{x} + \alpha e_h \in \mathcal{F},$$

for all $\alpha \in (0, \bar{\alpha})$. The case $\alpha = (y - \bar{x})^h < 0$ ends up with the same conclusions with e_h replaced by $-e_h$. Hence, in correspondence to any feasible point $\bar{x}$, either it is a stationary point, or there is a coordinate direction along which (or along its opposite) there must exist feasible points where the function is strictly decreased (this property is not guaranteed for different sets of n linearly independent directions). Therefore, the main idea behind the proposed algorithm will be performing finer and finer samplings of the objective function along the directions given by the canonical basis in $\mathbb{R}^n$ and their opposite.
At the same time, it will be possible either to understand that a point is a good approximation of a stationary point of F or to determine a specific direction along which the objective function decreases, which is guaranteed by the previous considerations.
On this basis, we propose an algorithm model that evaluates the objective function along the coordinate directions, taking care of the numerical error carried by the lower-level minimization, with the aim of detecting a feasible direction where the objective is sufficiently decreased. Once such a direction has been found, a derivative-free line search technique is adopted in order to explore a sufficiently large step along it, so as to exploit the descent property of the search direction as much as possible.
In general, when we deal with continuously differentiable functions, a positive spanning set for the feasible region near the current iterate is needed to ensure a decrease in the objective. The canonical basis fits well in this framework as it allows it to conform to the local geometry of the boundary.
Even though some regularity for the true function F is assumed, we do not have

access to the gradient.
The use of the coordinate directions as search directions and the particular sampling technique adopted allow us to overcome the lack of gradient information and to ensure that every limit point of the sequence produced is a stationary point for the problem in (2.1). Formally, the algorithm model is described as follows.

Algorithm DSS

Data. Choose $x_0 \in \mathcal{F}$, $\theta \in (0,1)$, $\gamma > 0$, $A \geq 0$ stepsize lower bound, $\tilde{\alpha}_0^i \in (A, +\infty)$, $d_i = e_i$ $i = \{1, ..., n\}$, where e_i are the vectors of the canonical basis in $\mathbb{R}^n$.

Step 0. $k = 0$, $i = 1$

Step1. Compute α_{max} s.t. $x_k + \alpha_{max} d_i \in \partial\mathcal{F}$ and set $\alpha = \min\{\alpha_{max}, \tilde{\alpha}_k^i\}$. If $\alpha > A$ and $\tilde{F}(x_k + \alpha d_i) \leq \tilde{F}(x_k) - \gamma(\alpha)^2$, go to Step 3.

Step2. Compute α_{max} s.t. $x_k - \alpha_{max} d_i \in \partial\mathcal{F}$ and set $\alpha = \min\{\alpha_{max}, \tilde{\alpha}_k^i\}$. If $\alpha > A$ and $\tilde{F}(x_k - \alpha d_i) \leq \tilde{F}(x_k) - \gamma(\alpha)^2$, set $d_i = -d_i$ and then go to Step 3, else, set $\alpha_k = 0, \tilde{\alpha}_{k+1}^i = \max\{A, \theta\tilde{\alpha}_k^i\}$ and go to Step 4.

Step3. Compute α_k by the Expansion Step$(d_i, \alpha, \alpha_{max}, \gamma)$ and set $\tilde{\alpha}_{k+1}^i = \alpha_k$.

Step4. Set $x_{k+1} = x_k + \alpha_k d_i, \tilde{\alpha}_{k+1}^j = \tilde{\alpha}_k^j \;\; \forall j \neq i$
Set $i = \mod(i, n) + 1$, $k = k + 1$ and go to Step 1.

In the following, the Expansion Step is presented.

Expansion Step$(d_i, \alpha, \alpha_{max}, \gamma)$

$\delta \in (0, 1)$.

Step1. Set $\tilde{\alpha} = \min\{\alpha_{max}, \frac{\alpha}{\delta}\}$
If $\alpha = \alpha_{max}$ or $\tilde{F}(x_k + \tilde{\alpha} d_i) > \tilde{F}(x_k) - \gamma(\tilde{\alpha})^2$ then set $\alpha_k = \alpha$ and stop.

Step2. Set $\alpha = \tilde{\alpha}$ and go to Step 1.

We give a brief explanation of how our algorithm works in the following.

In Step 1 the direction d_i is examined with the aim of finding (if possible) a feasible point where the objective function is decreased according to a sufficient decrease condition. First, it is computed the maximum feasible steplength α_{max} which can be performed along the direction d_i starting from the point x_k. Then, the trial stepsize α is determined by choosing the minimum between α_{max} and $\tilde{\alpha}_k^i$. The scalar $\tilde{\alpha}_k^i$ has been computed on the basis of the behavior of the objective function along the same direction shown at the previous iterations. Therefore, $\tilde{\alpha}_k^i$ should take into account the sensitivity of the objective function with respect to the i-th variable, and hence it should provide a promising initial stepsize for the direction d_i. Finally, provided that the stepsize is above the lower bound A, it is verified if the moving of length α along d_i produces a feasible point where the function is sufficiently reduced. If such a point is produced then a linesearch technique is performed along d_i to provide a suitable stepsize α_k (Step 3). Otherwise, the direction -d_i is considered (Step 2).

Step 2 is similar to Step 1, with d_i replaced by -d_i. In this case, if the trial point $x_k - \alpha d_i$ does not produce a sufficient decrease of F then the stepsize α_k is set equal to zero and the scalar $\tilde{\alpha}_k^i$ is reduced (observe that the stepsize is always above the threshold A). In this way, when the directions d_i and -d_i will be considered again by the algorithm, the initial stepsize will be chosen in an interval containing possibly smaller values.

At Step 3 a suitable large stepsize α_k is computed by a derivative-free line search technique. The aim of this step is to exploit the good descent direction d_i identified in Step 1 or Step 2. Then, the scalar $\tilde{\alpha}_{k+1}^i$ is set equal to α_k . The motivation for this choice derives from the fact that the stepsize α_k produced by a line search technique should identify promising values for the initial stepsize when the direction d_i (or -d_i) will be investigated.

At Step 4 the new point x_{k+1} is produced and, for the next iteration, a coordinate direction is selected by following the cyclic order.

3.1.1 Convergence Analysis

In the following, we focus on the theoretical properties of the algorithm model. In particular, we show that any accumulation point of the sequence generated by the proposed algorithm is a stationary point of problem in (2.1).
For the convergence analysis, if not specified otherwise, we set our attention on the case $A = 0$, when there is no lower bound for the algorithm' stepsizes.
First of all, we want to remark the algorithm is well-defined, meaning that the Expansion Step terminates in a finite number of iterations.

Proposition 3.1.1. *The line search terminates in a finite number of iterations.*

Proof. Recall that f is lower bounded by f_{low} by assumption (see Chapter 2).

It follows that $\tilde{F}$ is bounded as well. Indeed, by definition,
$\tilde{F}(x) = f(x, \tilde{y}(x)) \geq f_{low}$.
By contradiction, assume that for a given d_i,
$x_k + \delta^{-j}\alpha d_i \in F \ \ \forall\, j$ and $\tilde{F}(x_k + \delta^{-j}\alpha d_i) < \tilde{F}(x_k) - \gamma(\delta^{-j}\alpha) \ \ \forall\, j$.
This violates the assumption that $\tilde{F}$ is bounded below. □

We now report a technical result which is key to proving the main convergence result. The proof can be found in [14].

Proposition 3.1.2. *We have that*

(*i*) $\lim_{k\to+\infty} \alpha_k = 0$

(*ii*) $\lim_{k\to+\infty} \tilde{\alpha}_k^i = 0 \ \ \forall i \in \{1, ..., n\}$

Before proving the main result, let's state, as anticipated in Chapter 2, an assumption on the numerical error allowed in the lower-level approximation for the problem in (1.1).

Assumption 7. *Let $\{x_k\}_k$ be the sequence generated by the algorithm. At the k-th iteration of the algorithm, when the black-box oracle is performed, the maximal numerical error allowed is ϵ_k, where*

$$\epsilon_k := \begin{cases} \epsilon, & \textit{if } \ k = 0 \\ \theta^2 \epsilon_{k-1}, & \textit{otherwise} \end{cases} \tag{3.6}$$

The geometric decrease for the numerical error will make sense soon.
Assumption 7 can be relaxed, setting the multiplying factor as $\theta^{2-l(k)}$ with $l(k) \to 0$ as $k \to \infty$.

We are ready for the main convergence result. Briefly speaking, any accumulation point of the sequence generated by the proposed algorithm is a stationary point, meaning that (3.2) holds.

Proposition 3.1.3. *Let $\{x_k\}_k$ be the sequence generated by the proposed algorithm.*
There exist positive constants $c_1, c_2 > 0$ such that

$$\|\nabla^{red} F(x_k)\| \leq c_1 \underset{l\in\{1,...,n\}}{max} \{\tilde{\alpha}_k^l\} + \frac{c_2 L_f \epsilon_k}{\underset{l\in\{1,...,n\}}{min} \{\tilde{\alpha}_k^l\}} \ \ \forall k \geq n \tag{3.7}$$

It follows directly from (3.7) that

$$\|\nabla^{red} F(x_k)\| \to 0 \ \textit{as} \ k \to +\infty \tag{3.8}$$

.

Proof. Fix $i \in \{1, ..., n\}$, let $k \geq n$ and let $i(k)$ be the biggest index where the direction e_i (and/or -e_i) has been explored. By instructions of the algorithm, $k - i(k) \leq n$.
We distinguish three cases:

(i) $x_k^i = u^i$

(ii) $x_k^i = l^i$

(iii) $l^i < x_k^i < u^i$

Focus first on (i).
We may have:

$(ia) : x_{i(k)}^i = u^i$

or

$(ib) : x_{i(k)}^i < u^i$

In (ia), we have $\alpha_{i(k)} = 0, \quad \tilde{\alpha}_{i(k)+1}^i = \tilde{\alpha}_k^i$ and

$$F\left(x_{i(k)} - \tfrac{\tilde{\alpha}_k^i}{\theta} e_i\right) > F(x_{i(k)}) - \gamma\left(\tfrac{\tilde{\alpha}_k^i}{\theta}\right)^2 - 2L_f\epsilon_k$$

By the Mean Value Theorem:

$$-\tfrac{\tilde{\alpha}_k^i}{\theta} \nabla F\left(u_{i(k)}\right)^T e_i > -\gamma\left(\tfrac{\tilde{\alpha}_k^i}{\theta}\right)^2 - 2L_f\epsilon_k$$

where $u_{i(k)} = x_{i(k)} - \lambda_{i(k)} \frac{\tilde{\alpha}_k^i}{\theta} e_i$, $\lambda_{i(k)} \in (0, 1)$.

$$[\nabla F(u_{i(k)}) - \nabla F(x_k) + \nabla F(x_k)]^T e_i < \gamma \tfrac{\tilde{\alpha}_k^i}{\theta} + \tfrac{2L_f\epsilon_k\theta}{\tilde{\alpha}_k^i}$$

from which, taking into account the Lipschitz assumption on ∇F, it follows

$$\begin{aligned} \nabla F(x_k)^T e_i &< \gamma \frac{\tilde{\alpha}_k^i}{\theta} + \frac{2L_f\epsilon_k\theta}{\tilde{\alpha}_k^i} + L\|x_k - u_{i(k)}\| \\ &\leq \gamma \frac{\tilde{\alpha}_k^i}{\theta} + \frac{2L_f\epsilon_k\theta}{\tilde{\alpha}_k^i} + L\|x_k - x_{i(k)}\| + L\frac{\tilde{\alpha}_k^i}{\theta} \end{aligned} \tag{3.9}$$

We have

$$x_k = x_{i(k)} + \textstyle\sum_{j=0}^{k-i(k)-1} \alpha_{i(k)+j} d_{i(k)+j}$$

For each j such that $\alpha_{i(k)+j} \neq 0$, recalling instructions of the algorithm, we have that there exists an index $l \in \{1, ..., n\}$ such that
$\tilde{\alpha}_{i(k)+j+1}^l = \alpha_{i(k)+j}$, and $\tilde{\alpha}_k^l = \tilde{\alpha}_{i(k)+j+1}^l$. Therefore, it follows

$$\tilde{\alpha}_k^l \le \max_{l\in\{1,\dots,n\}}\{\tilde{\alpha}_k^l\}$$

and we can write

$$\|x_k - x_{i(k)}\| \le \max_{l\in\{1,\dots,n\}}\{\tilde{\alpha}_k^l\}.$$

From (3.9) we get

$$\nabla_i{}^{red}F(x_k) < \frac{(\gamma + L(n+1))}{\theta}\max_{l\in\{1,\dots,n\}}\{\tilde{\alpha}_k^l\} + \frac{2L_f\epsilon_k\theta}{\min\limits_{l\in\{1,\dots,n\}}\{\tilde{\alpha}_k^l\}} \tag{3.10}$$

In case (ib), we have $\alpha_{i(k)} \ne 0$, $\tilde{\alpha}_{i(k)+1}^i = \alpha_{i(k)} = \tilde{\alpha}_k^i$ and

$$F\left(x_{i(k)} + \tilde{\alpha}_k^i e_i\right) \le F(x_{i(k)}) - \gamma(\tilde{\alpha}_k^i)^2 + 2L_f\epsilon_k \le F(x_{i(k)}) + \gamma(\tilde{\alpha}_k^i)^2 + 2L_f\epsilon_k$$

Then, by applying the Mean Value Theorem, we obtain

$$\nabla F(v_{i(k)})^T e_i < \gamma\tilde{\alpha}_k^i + \tfrac{2L_f\epsilon_k}{\tilde{\alpha}_k^i}$$

where $v_{i(k)} = x_{i(k)} + \lambda_{i(k)}\tilde{\alpha}_k^i e_i$, with $\lambda_{i(k)} \in (0,1)$. Then by repeating the previous reasonings we obtain

$$\nabla_i{}^{red}F(x_k) < (\gamma + L(n+1))\max_{l\in\{1,\dots,n\}}\{\tilde{\alpha}_k^l\} + \frac{2L_f\epsilon_k}{\min\limits_{l\in\{1,\dots,n\}}\{\tilde{\alpha}_k^l\}} \tag{3.11}$$

Case (ii) is analogous to Case (i), so that conditions (3.10) and (3.11) hold.

Case (iii) splits into two possible subcases:

$(iiia)$: $x_{i(k)}^i = x_k^i$

or

$(iiib)$: $x_{i(k)}^i \ne x_k^i$

In case $(iiia)$, from the prescribed instructions, it follows that

$$F\left(x_{i(k)} + \frac{\tilde{\alpha}_k^i}{\theta}e_i\right) \ge F(x_{i(k)}) - \gamma\left(\frac{\tilde{\alpha}_k^i}{\theta}\right)^2 - 2L_f\epsilon_k \tag{3.12}$$

$$F\left(x_{i(k)} - \frac{\tilde{\alpha}_k^i}{\theta}e_i\right) \ge F(x_{i(k)}) - \gamma\left(\frac{\tilde{\alpha}_k^i}{\theta}\right)^2 - 2L_f\epsilon_k \tag{3.13}$$

By the Mean Value Theorem, we obtain

$$\nabla F(u_{i(k)})^T e_i \ge -\gamma\left(\frac{\tilde{\alpha}_k^i}{\theta}\right) - \frac{2L_f\epsilon_k\theta}{\tilde{\alpha}_k^i} \tag{3.14}$$

$$\nabla F(v_{i(k)})^T e_i \leq \gamma\left(\frac{\tilde{\alpha}_k^i}{\theta}\right) + \frac{2L_f\epsilon_k\theta}{\tilde{\alpha}_k^i} \tag{3.15}$$

where $u_{i(k)} = x_{i(k)} + \lambda^1_{i(k)}\frac{\tilde{\alpha}_k^i}{\theta}e_i, \quad v_{i(k)} = x_{i(k)} - \lambda^2_{i(k)}\frac{\tilde{\alpha}_k^i}{\theta}e_i$, with $\lambda^1_{i(k)}, \lambda^2_{i(k)} \in (0,1)$.
From (3.14), taking into account the Lipschitz assumption on ∇F we get

$$\begin{aligned}
\nabla F(x_k)^T e_i &\geq -\gamma\left(\frac{\tilde{\alpha}_k^i}{\theta}\right) - \frac{2L_f\epsilon_k\theta}{\tilde{\alpha}_k^i} - L\|x_k - u_{i(k)}\| \\
&\geq \nabla F(x_k)^T e_i - \gamma\left(\frac{\tilde{\alpha}_k^i}{\theta}\right) - \frac{2L_f\epsilon_k\theta}{\tilde{\alpha}_k^i} - L\|x_k - x_{i(k)}\| - L\frac{\tilde{\alpha}_k^i}{\theta} \\
&\geq -\gamma\left(\frac{\tilde{\alpha}_k^i}{\theta}\right) - \frac{2L_f\epsilon_k\theta}{\tilde{\alpha}_k^i} - nL\max_{l\in\{1,\ldots,n\}}\{\tilde{\alpha}_k^l\} - \frac{\tilde{\alpha}_k^i}{\theta} \\
&= -(\gamma + L)\frac{\tilde{\alpha}_k^i}{\theta} - \frac{2L_f\epsilon_k\theta}{\tilde{\alpha}_k^i} - nL\max_{l\in\{1,\ldots,n\}}\{\tilde{\alpha}_k^l\}
\end{aligned}$$

Hence, it follows

$$\nabla_i^{red} F(x_k) \geq -\frac{(\gamma + L(n+1))}{\theta}\max_{l\in\{1,\ldots,n\}}\{\tilde{\alpha}_k^l\} - \frac{2L_f\epsilon_k\theta}{\min\limits_{l\in\{1,\ldots,n\}}\{\tilde{\alpha}_k^l\}} \tag{3.16}$$

From (3.15), by repeating the same reasonings, we obtain

$$\nabla_i^{red} F(x_k) \leq \frac{(\gamma + L(n+1))}{\theta}\max_{l\in\{1,\ldots,n\}}\{\tilde{\alpha}_k^l\} + \frac{2L_f\epsilon_k\theta}{\min\limits_{l\in\{1,\ldots,n\}}\{\tilde{\alpha}_k^l\}} \tag{3.17}$$

From (3.16) and (3.17) it follows

$$|\nabla_i{}^{red} F(x_k)| \leq \frac{(\gamma + L(n+1))}{\theta}\max_{l\in\{1,\ldots,n\}}\{\tilde{\alpha}_k^l\} + \frac{2L_f\epsilon_k\theta}{\min\limits_{l\in\{1,\ldots,n\}}\{\tilde{\alpha}_k^l\}} \tag{3.18}$$

Let's consider now (*iiib*). Without loss of generality, we can assume that in this case we have $\alpha_{i(k)} \neq 0, \tilde{\alpha}^i_{i(k)+1} = \tilde{\alpha}_k^i = \alpha_{i(k)}$,

$$F\left(x_{i(k)} + \tilde{\alpha}_k^i e_i\right) \leq F(x_{i(k)}) - \gamma(\tilde{\alpha}_k^i)^2 + 2L_f\epsilon_k$$

and

$$F\left(x_{i(k)} + \bar{\alpha}_k^i e_i\right) > F(x_{i(k)}) - \gamma(\bar{\alpha}_k^i)^2 - 2L_f\epsilon_k$$

with $\bar{\alpha}_k^i = \frac{\tilde{\alpha}_k^i}{\delta_{i(k)}}$ where $\delta_{i(k)} = \delta$ if $x_{i(k)} + \frac{\tilde{\alpha}_k^i}{\delta}e_i \in F$ and $\delta_{i(k)} \in (\delta, 1)$ otherwise. By applying the Mean Value Theorem, we can write

$$\nabla F(u_{i(k)})^T e_i \leq -\gamma\tilde{\alpha}_k^i + 2\frac{L_f\epsilon_k}{\tilde{\alpha}_k^i} \leq \gamma\tilde{\alpha}_k^i + 2\frac{L_f\epsilon_k}{\alpha_k^i}$$

and

$$\nabla F(v_{i(k)})^T e_i > -\gamma \frac{\tilde{\alpha}_k^i}{\delta_{i(k)}} - 2\frac{L_f \epsilon_k \delta_{i(k)}}{\tilde{\alpha}_k^i}$$

where $u_{i(k)} = x_{i(k)} + \lambda_{i(k)}^1 \alpha_{i(k)}^i e_i$,$v_{i(k)} = x_{i(k)} + \lambda_{i(k)}^2 \alpha_{i(k)}^i e_i$ with $\lambda_{i(k)}^1, \lambda_{i(k)}^2 \in (0,1)$.

Taking into account the Lipschitz assumption on ∇F we can write

$$\begin{aligned}
\nabla F(x_k)^T e_i &\leq \gamma \tilde{\alpha}_k^i + \frac{2L_f\epsilon_k}{\tilde{\alpha}_k^i} + L\|x_k - u_{i(k)}\| \\
&\leq \gamma \tilde{\alpha}_k^i + \frac{2L_f\epsilon_k}{\tilde{\alpha}_k^i} + L\|x_k - x_{i(k)}\| + L\tilde{\alpha}_{i(k)+1}^i \\
&\leq (\gamma + L)\tilde{\alpha}_k^i + \frac{2L_f\epsilon_k}{\tilde{\alpha}_k^i} + nL \max_{l\in\{1,\dots,n} \{\tilde{\alpha}_k^l\}
\end{aligned}$$

$$\begin{aligned}
\nabla F(x_k)^T e_i &\geq -\gamma \frac{\tilde{\alpha}_k^i}{\delta_{i(k)}} - \frac{2L_f\epsilon_k\delta_{i(k)}}{\tilde{\alpha}_k^i} - L\|x_k - v_{i(k)}\| \\
&\geq -\gamma \frac{\tilde{\alpha}_k^i}{\delta} - \frac{2L_f\epsilon_k}{\tilde{\alpha}_k^i} - L\|x_k - x_{i(k)}\| - L\tilde{\alpha}_k^i \\
&\geq -(\gamma + L)\frac{\tilde{\alpha}_k^i}{\delta} - \frac{2L_f\epsilon_k}{\tilde{\alpha}_k^i} - nL \max_{l\in\{1,\dots,n} \{\tilde{\alpha}_k^l\}
\end{aligned}$$

Then, we have

$$|\nabla_i^{red} F(x_k)| \leq \frac{(\gamma + L(n+1))}{\delta} \max_{l\in\{1,\dots,n\}} \{\tilde{\alpha}_k^l\} + \frac{2L_f\epsilon_k}{\min\limits_{l\in\{1,\dots,n\}} \{\tilde{\alpha}_k^l\}} \tag{3.19}$$

Finally, from (3.16), (3.17),(3.19), we obtain

$$\|\nabla^{red} F(x_k)\| \leq n^{1/2} \left(\frac{(\gamma + L(n+1))}{\max\{\theta, \delta\}} \max_{l\in\{1,\dots,n\}} \{\tilde{\alpha}_k^l\} + \frac{2L_f\epsilon_k}{\min\limits_{l\in\{1,\dots,n\}} \{\tilde{\alpha}_k^l\}} \right) \forall k \geq n$$

and this proves (3.7).

If there is no lower bound for the stepsize, then, by the bound above given by (3.7), the asymptotic behavior of the algorithm stepsizes given by (3.1.2) and Assumption 7 concerning the numerical noise produced for our reduced-upper-level problem, it follows that (3.8) holds true, and this concludes the proof. □

3.1.2 Stopping Condition

Now, we describe the stopping criterion employed. In practice, we look for an algorithm that produces a reasonably optimal solution in finite time.
To this scope, we consider now the case $A > 0$ and the proposed algorithm stops if $\tilde{\alpha}_k^j = A \;\; \forall j \in \{1, ..., n\}$.
We start by proving that our method terminates a finite number of iterations.

Proposition 3.1.4. *Let $\{x_k\}_k$ be the sequence generated by Algorithm DSS. The algorithm terminates in finite time.*

Proof. $\{\tilde{F}(x_k)\}_k$ is non-decreasing. In particular $\tilde{F}(x_k) = \tilde{F}(x_{k+1})$ after an unsuccessful step and

$$\tilde{F}(x_{k+1}) \leq \tilde{F}(x_k) - \gamma A^2$$

after a successful one. Hence, we have at most

$$\frac{\left(\tilde{F}(x_0) - \inf_{x \in F} \tilde{F}(x)\right)}{\gamma A^2} \leq \frac{\tilde{F}(x_0) - f_{low} + 2L_f \epsilon}{\gamma A^2}$$

successful steps.
By the instructions of the algorithm, eventually the stepsizes $\{\tilde{\alpha}_k^i\}_k$ will reach the lower bound A (in finite time, due to the geometric contraction by the factor θ and the finiteness of successful steps).
Hence, the algorithm terminates in a finite time. □

The result in (3.7) still holds true in the case where $A > 0$.
One could follow the very same steps of the proof of (3.7), replacing $\frac{\tilde{\alpha}_k^i}{\theta}$ with $\tilde{\alpha}_{i(k)}^i$, since in this framework, when an iteration fails, the stepsize is reduced, but a strictly positive lower bound, i.e. A, is active.
Another issue emerges as we consider (*iiia*) since we should also include the case where the point $x_{i(k)}$ is too close to the boundary and either e_i or -e_i are not explorable. One may observe that this issue can be easily avoided since one ends up in one of the cases already treated in the original proof.

Proposition 3.1.5. *Let $A > 0$.*
Let $\{x_k\}_k$ be the sequence generated by the proposed algorithm.
There exist positive constants $c_1, c_2 > 0$ such that

$$\|\nabla^{red} F(x_k)\| \leq c_1 \underset{l \in \{1,\dots,n\}}{max} \{\tilde{\alpha}_k^l\} + \frac{c_2 L_f \epsilon_k}{\underset{l \in \{1,\dots,n\}}{min} \{\tilde{\alpha}_k^l\}} \;\; \forall k \geq n \tag{3.20}$$

As anticipated, in practice we end in finite time, so the following guarantees us that the point returned by the algorithm is reasonably optimal. Combining the bound given by Proposition 3.1.4 and Proposition 3.1.5, the following holds true.

Proposition 3.1.6. *Algorithm DSS terminates in $\bar{k}$ iterations, with $\bar{k} \geq n$. At the last iteration, our model produces an element $x_{\bar{k}}$ such that*

$$\|\nabla^{red} F(x_{\bar{k}})\| \leq c_1 A + \frac{c_2 L_f \epsilon}{A} \tag{3.21}$$

Corollary 3.1.1. *Let $A = \sqrt{\frac{4c_2 L_f \epsilon}{c_1}}$. Algorithm DSS terminates in $\bar{k}$ iterations, with $\bar{k} \geq n$. At the last iteration, our model produces an element $x_{\bar{k}}$ such that*

$$\|\nabla^{red} F(x_{\bar{k}})\| \leq \sqrt{c_1 c_2 L_f \epsilon} \tag{3.22}$$

Proof. Just plug A defined as above in (3.21) and the thesis holds. □

3.2 Nonsmooth objective

In the following, we aim to solve the problem in (2.1) when the true function F (though nonsmooth) is Lipschitz continuous and that first-order information is unavailable or impractical to obtain. In this subsection, only box constraints are considered, while more general constraints will be explored in the next chapter. We will assume, later on, that two different classes of constraints exist, namely, difficult general nonsmooth constraints ($h(x) \leq 0$) and simple bound constraints on the problem variables ($l \leq x \leq u$). The main idea consists of getting rid of the nonlinear constraints by means of an exact penalty approach, but we will get back to that.

First of all, we want to highlight that the assumption introduced in the previous subsection, which concerns the geometrical decrease of the numerical error allowed in the lower-level, will be needed as well here, so from now on we tacitly assume that Assumption 7 holds.

We now recall the necessary definitions.
First of all, just to make it easier for the reader, we introduce some notation we will use further in the thesis.

Definition 3.2.1. Let $\mathcal{F}$ be given by box constraints as we mentioned above. Given problem in (2.1) and any point $x \in \mathcal{F}$,
$D(x) = \{d \in \mathbb{R}^n : d_i \geq 0 \text{ } if \text{ } x_i = l_i, \text{ } d_i \leq 0 \text{ } if \text{ } x_i = u_i, \text{ } d_i \in \mathbb{R} \text{ } if \text{ } l_i < x_i < u_i\}$
is the cone of feasible directions at x with respect to $\mathcal{F}$.

We also report a technical proposition whose proof can be found in [3].

Proposition 3.2.1. *Given problem in (2.1), let $\{x_k\}_k \subset \mathcal{F}$ for all k and $\{x_k\}_k \to \bar{x}$ for $k \to \infty$. Then, for k sufficiently large,*

$$D(\bar{x}) \subseteq D(x_k)$$

Let's give the definition of Clarke stationarity, which is relevant for unconstrained problems.

Definition 3.2.2. Given a point $x \in \mathbb{R}^n$, the Clarke directional derivative of function F along direction $d \in \mathbb{R}^n$ is given by

$$F^{Cl}(x; d) = \limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{F(y + td) - F(y)}{t} \tag{3.23}$$

Definition 3.2.3. Given the problem $\min_{x \in \mathbb{R}^n} F(x)$, x is a Clarke stationary point if

$$F^{Cl}(x; d) \geq 0 \quad \forall\ d \in \mathbb{R}^n \tag{3.24}$$

The necessary optimality conditions for the problem in (2.1) can be instead characterized in terms of the Clarke–Jahn generalized directional derivative of the objective function, which takes into consideration the constraints (see [7]).

Definition 3.2.4. Given a point $x \in \mathcal{F}$, the Clarke–Jahn generalized directional derivative of function F along direction d is given by

$$F^{\circ}(x; d) = \limsup_{\substack{y \to x, y \in F \\ t \downarrow 0, y + td \in F}} \frac{F(y + td) - F(y)}{t} \tag{3.25}$$

We recall that every local minimum of problem in (2.1) satisfies the following definition.

Definition 3.2.5. Given problem in (2.1), x is a Clarke–Jahn stationary point if

$$F^{\circ}(x; d) \geq 0 \quad \forall\ d \in D(x) \tag{3.26}$$

Since the true objective function F is possibly not continuously differentiable on $\mathcal{F}$, a finite number of search directions is not sufficient to investigate the local behavior of F on $\mathcal{F}$. In this framework, it is common and also necessary to assume density properties on particular subsequences of the search directions. In this way, we are able to prove convergence to stationary points of problem in (2.1). To this purpose, here we propose a very simple derivative-free algorithm for solving the nonsmooth problem in (2.1), namely, Algorithm DSN$_{simple}$.
In this algorithm, we use a predefined sequence of search directions $\{d_k\}_k$. Then, we investigate the behavior of the function F along the direction d_k by means of the line search procedure Projected Continuous Search. Given the current iterate x_k at step k, the latter procedure first evaluates the function at $[x_k \pm \tilde{\alpha}_k d_k]_{[l,u]}$, where$[\cdot]_{[l,u]}$ denotes the projection on the box.
In case a sufficient reduction of the function value is obtained, an extrapolation along the search direction is performed, so that a suitable steplength α_k is computed, and is used as a tentative steplength for the next iteration, i.e., $\tilde{\alpha}_{k+1} = \tilde{\alpha}_k$. On the other hand, if at $[x_k \pm \tilde{\alpha}_k d_k]_{[l,u]}$ we do not obtain a sufficient reduction of the function value, then the tentative steplength at the next

iteration is suitably reduced by a scale factor, i.e., $\tilde{\alpha}_{k+1} = \theta\tilde{\alpha}_k$, $\theta \in (0, 1)$. More formally, the resulting algorithm and the proposed line search procedure are summarized in the next schemes.

Algorithm DSN$_{simple}$

Data. Choose $x_0 \in \mathcal{F}$, $\theta \in (0,1)$, $A \geq 0$ stepsize lower bound, $\tilde{\alpha_0} \in (A, +\infty)$, $\{d_k\}_k$ as above.

Step 0. $k = 0$.

Step1. Compute α_k and $\tilde{\alpha}_k$ by the Projected Continuous Search.

Step2. If $\alpha_k = 0$ then set $\tilde{\alpha}_{k+1} = \max\{A, \theta\tilde{\alpha}_k\}$
Else set $\tilde{\alpha}_{k+1} = \alpha_k$ and $x_{k+1} = [x_k + \tilde{\alpha}_k d_k]_{[l,u]}$.
Set $k = k + 1$ and go to Step 1.

In the following, the Projected Continuous Search is presented.

Projected Continuous Search($\tilde{\alpha}, y, p; \alpha, p^+$)

$\gamma > 0$, $\delta \in (0,1)$.

Step0. Set $\alpha = \tilde{\alpha}$.

Step1. If $\tilde{F}\left([y + \alpha p]_{[l,u]}\right) \leq \tilde{F}(y) - \gamma(\alpha)^2$ then set $p^+ = p$ and go to Step 4.

Step2. If $\tilde{F}\left([y - \alpha p]_{[l,u]}\right) \leq \tilde{F}(y) - \gamma(\alpha)^2$ then set $p^+ = p$ and go to Step 4.

Step3. Set $\alpha = 0$, return α and $p^+ = p$.

Step4. Let $\beta = \alpha/\delta$.

Step5. If $\tilde{F}\left([y + \alpha p^+]_{[l,u]}\right) > \tilde{F}(y) - \gamma(\beta)^2$ return α, p^+.

Step6. Set $\alpha = \beta$ and go to Step 4.

For clarity, we note that the Projected Continuous Search procedure takes in input $\tilde{\alpha}, y$, and p (that is, the arguments before the semicolon) and gives in output α and p^+ (that is, the arguments after the semicolon).
In the following results, we analyze the global convergence properties of Algo-

rithm DSN$_{simple}$.

3.2.1 Convergence Analysis

We start by proving in the next proposition that the Projected Continuous Search cannot cycle.

Proposition 3.2.2. *The line search cannot infinitely cycle between Step 4 and Step 6.*

Proof. Let us consider the Projected Continuous Search. We proceed by contradiction assuming that an infinite monotonically increasing sequence of positive numbers $\{\beta_j\}_j$ exists such that

$$\tilde{F}([y + \beta_j p^+]_{[l,u]}) \leq \tilde{F}(y) - \gamma(\beta_j)^2$$

By Assumption 3 in Chapter 2, f is lower bounded, hence $\tilde{F}$ is lower bounded as well, so the above relation contradicts the boundness of $\tilde{F}$ and we conclude. □

Now, as in the smooth case, from now on we focus on the case $A = 0$, and in the following proposition we state that the stepsizes computed by the Projected Continuous Search procedure eventually go to zero. The reader can find the proof in [1].

Proposition 3.2.3. *We have that*

(i) $\lim_{k\to+\infty} \alpha_k = 0$

(ii) $\lim_{k\to+\infty} \tilde{\alpha}_k = 0$

Using the latter result we can provide the next technical lemma, which will be necessary to prove the main global convergence result for Algorithm DSN$_{simple}$. This lemma shows that the projection operator does not sensibly deteriorate the asymptotic properties of the directions d_k. More precisely, performing a steplength η_k along d_k and assuming that η_k goes to zero, it results that eventually the new point $[x_k + \eta_k d_k]_{[l,u]}$ differs from x_k and the scaled actual step $([x_k + \eta_k d_k]_{[l,u]} - x_k)/\eta_k$ enjoys the same asymptotic properties of d_k.
The reader may refer to the proof provided in [1].

Proposition 3.2.4. *Let $\{x_k\}_k$ be the sequence produced by Algorithm DSN$_{simple}$, let $\{d_k\}_k$ be the sequence of search directions used by the proposed model, and let $\{\eta_k\}_k$ be a sequence such that $\eta_k > 0$ for all k. Further, let K be a subset of indices such that*

$$\lim_{k\to+\infty, k\in K} x_k = \bar{x} \tag{3.27}$$

$$\lim_{k\to+\infty, k\in K} d_k = \bar{d} \tag{3.28}$$

$$\lim_{k\to+\infty, k\in K} \eta_k = 0 \tag{3.29}$$

with $\bar{x} \in F$ and $\bar{d}$ feasible direction for F in $\bar{x}$, $\bar{d} \neq 0$. Then, (*i*) *for all* $k \in K$ *sufficiently large,* $[x_k + \eta_k d_k]_{[l,u]} \neq x_k$

(*ii*) $\lim_{\substack{k \to +\infty \\ k \in K}} \left(\frac{[x_k + \eta_k d_k]_{[l,u]} - x_k}{\eta_k} \right) = \bar{d}$

Finally, we are now ready to prove the main convergence result for Algorithm DSN$_{simple}$. We highlight that according to the following proposition, every limit point of the sequence of iterates $\{x_k\}_k$, produced by Algorithm DSN$_{simple}$, is a stationary point for the problem in (2.1).

Proposition 3.2.5. *Let* $\{x_k\}_k$ *be the sequence produced by Algorithm DSN*$_{simple}$*. Let* $\bar{x}$ *be any accumulation point of* $\{x_k\}_k$ *and* K *be a subset of indices such that*

$$\lim_{k \to +\infty, k \in K} x_k = \bar{x}$$

If the subsequence $\{d_k\}_{k \in K}$ *is dense in the unit sphere, then* $\bar{x}$ *is Clarke–Jahn stationary for problem in* (2.1)*.*

Proof. By contradiction, let $\bar{d} \in S^{n-1}$ feasible direction such that

$$F^\circ(\bar{x}, \bar{d}) < 0 \tag{3.30}$$

where we denote with S^{n-1} the unit sphere in $\mathbb{R}^n$.
By recalling the instructions of the Projected Continuous Search, if the condition at Step 1 is satisfied, we have $\alpha_k > 0$ and

$$\tilde{F}\left(\left[x_k + \left(\frac{\alpha_k}{\delta} \right) d_k \right]_{[l,u]} \right) > \tilde{F}(x_k) - \gamma \left(\frac{\alpha_k}{\delta} \right)^2 \tag{3.31}$$

otherwise, we have

$$\tilde{F}\left([x_k + \tilde{\alpha}_k d_k]_{[l,u]} \right) > \tilde{F}(x_k) - \gamma \tilde{\alpha}_k^2 \tag{3.32}$$

Now, for every index $k \in K$, let us set

$$\eta_k = \begin{cases} \alpha_k/\delta & \textit{if } (3.30) \textit{ holds} \\ \tilde{\alpha}_k & \textit{if } (3.31) \textit{ holds} \end{cases}$$

and let v_k be defined as in Proposition (3.2.4), that is,

$$v_k = \frac{[x_k + \eta_k d_k]_{[l,u]} - x_k}{\eta_k}$$

The instructions of Algorithm DSN$_{simple}$ and the definition of η_k guarantee that $\eta_k > 0$ for all $k \in K$. Moreover, by Proposition (3.2.3),

$$\lim_{k \to +\infty} \eta_k = 0 \tag{3.33}$$

Up to subsequences, it follows that

$$\lim_{k\to+\infty, k\in K} x_k = \bar{x} \tag{3.34}$$

$$\lim_{k\to+\infty, k\in K} d_k = \bar{d} \tag{3.35}$$

Hence, by (3.33), (3.34) and (3.35), hypothesis of the technical result given in (3.2.4) are satisfied.
Therefore, we have that $v_k \neq 0$ for $k \in K$ and sufficiently large, so that relations in (3.31) and (3.32) can be equivalently expressed as

$$\tilde{F}(x_k + \eta_k v_k) > \tilde{F}(x_k) - \gamma(\eta_k)^2 \tag{3.36}$$

and, hence, it follows that

$$F(x_k + \eta_k v_k) > F(x_k) - \gamma(\eta_k)^2 - 2L_f \epsilon_k \tag{3.37}$$

that is, recalling that $\eta_k > 0$,

$$\frac{F(x_k + \eta_k v_k) - F(x_k)}{\eta_k} > -\gamma\eta_k - \frac{2L_f \epsilon_k}{\eta_k} \tag{3.38}$$

for $k \in K$ and sufficiently large. Then we can write

$$\limsup_{\substack{x_k\to x, x_k\in F\\ t\downarrow 0, x_k+t\bar{d}\in F}} \frac{F(x_k + t\bar{d}) - F(x_k)}{t} \geq \limsup_{k\to+\infty, k\in K} \frac{F(x_k + \eta_k\bar{d}) - F(x_k)}{\eta_k}$$

$$\begin{aligned}
&= \limsup_{k\to+\infty, k\in K} \frac{F(x_k + \eta_k\bar{d}) + F(x_k + \eta_k v_k) - F(x_k + \eta_k v_k) - F(x_k)}{\eta_k} \\
&\geq \limsup_{k\to+\infty, k\in K} \left(\frac{F(x_k + \eta_k v_k) - F(x_k)}{\eta_k} - L\|\bar{d} - v_k\| \right) \\
&\geq \limsup_{k\to+\infty, k\in K} \left(-\gamma\eta_k - \frac{2L_f\epsilon_k}{\eta_k} - L\|\bar{d} - v_k\| \right) = 0
\end{aligned}$$

where the last equality follows by Assumption 7 together with Proposition (3.2.4). This contradicts (3.30) and concludes the proof. □

3.2.2 A more efficient algorithm for the nonsmooth case

A possible way to improve the efficiency of Algorithm DSN_{simple} is to take advantage of the experience in the smooth case. We can draw inspiration from Algorithm DSS, where the objective function is repeatedly investigated along the directions $\pm e_1, \dots, \pm e_n$ in order to capture the local behavior of the objective function. In fact, the use of a set of search directions, which is constant with iterations, allows us to store the actual and tentative steplengths, i.e., α^i and $\tilde{\alpha}^i$, respectively, that roughly summarize the sensitivity of the function along those directions. Thus, when the function is further investigated along such search directions, we can exploit information gathered in the previous searches along them.

In the following, we propose a new algorithm, where we first explore the coordinate directions, and then a further direction d_k is explored. In particular, the sampling along the coordinate directions is performed by means of the continuous search procedure used for Algorithm DSS. Concerning the above definition of the new version of Algorithm DSN_{simple}, we remark that we will focus on its asymptotic convergence properties.

Algorithm CS-DSN

Data. Choose $x_0 \in \mathcal{F}$, $\theta \in (0,1)$, $A \geq 0$ stepsize lower bound, $\tilde{\alpha}_0 > A$, $\tilde{\alpha}_0^i > A$, $d_0^i = e_i$ for $i = 1, \dots, n$, and a sequence $\{d_k\}_k$ of search directions as for Algorithm DSN_{simple}.

Step 0. $k = 0$.

Step1. Set $y_k^1 = x_k$.

For $i = 1, \dots, n$:

Compute α and d_{k+1}^i by the Expansion Step$\big(\tilde{\alpha}_k^i, y_k^i, d_k^i; \alpha, d_{k+1}^i\big)$.

If $(\alpha = 0)$, then set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta\tilde{\alpha}_k^i$;

else, set $\alpha_k^i = \alpha$ and $\tilde{\alpha}_{k+1}^i = \alpha$.

Set $x_k^i = y_k^i + \alpha_k^i d_{k+1}^i$.

Step2. Compute α_k and $\tilde{\alpha}_k$ by the Projected Continuous Search.

Step3. If $\alpha_k = 0$ then set $\tilde{\alpha}_{k+1} = \max\{A, \theta\tilde{\alpha}_k\}$
Else set $\tilde{\alpha}_{k+1} = \alpha_k$ and $x_{k+1} = [x_k + \tilde{\alpha}_k d_k]_{[l,u]}$.
Set $k = k + 1$ and go to Step 1.

---

where the Expansion Step and the Projected Continuous Search are the very same presented, respectively, in Algorithm DSS and Algorithm DSN_{simple}.

The following three propositions concern the convergence analysis of Algorithm CS-DSN. The proofs are omitted since one has just to follow the very same steps adopted to prove the same results in the previous sections.

Proposition 3.2.6. *Both the Expansion Step and Projected Continuous Search cannot infinitely cycle between Step 6 and Step 8.*

The proposition that follows concerns the convergence to zero of the steplengths in Algorithm CS-DSN.

Proposition 3.2.7. *Let* $\{\alpha_k^i\}_k$, $\{\tilde{\alpha}_k^i\}_k$, $\{\alpha_k\}_k$, *and* $\{\tilde{\alpha}_k\}$ *be the sequences produced by Algorithm CS-DSN; then all the sequences converge to 0 as* $k \to \infty$.

Proposition 3.2.8. *Let* $\{x_k\}_k$ *be the sequence produced by Algorithm CS-DSN. Let* $\bar{x}$ *be any accumulation point of* $\{x_k\}_k$ *and* K *be a subset of indices such that*

$$\lim_{k \to +\infty, k \in K} x_k = \bar{x}$$

If the subsequence $\{d_k\}_{k \in K}$ *is dense in the unit sphere, then* $\bar{x}$ *is Clarke–Jahn stationary for problem in* (2.1).

Chapter 4

The nonsmooth nonlinearly constrained case

In the following chapter, we consider constrained problems when $\mathcal{F}$ is given by both bound and nonlinear constraints.
In particular $\mathcal{F} = \{x \in \mathbb{R}^n \ s.t. \ l \leq \ x \leq u, \ h(x) \leq 0\}$, where the vectors l and u correspond respectively to lower and upper bounds on the variables $x \in \mathbb{R}^n$ and satisfy the additional condition $l < u$. We also assume throughout the thesis that $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are Lipschitz continuous functions, though they may be possibly non-differentiable.
It is convenient to introduce the two following sets and the reason will be clear soon.
We denote with $\mathcal{X} = \{x \in \mathbb{R}^n \ s.t. \ l \leq \ x \leq u\}$ the set of bound constraints, while we denote with $\mathcal{H} = \{x \in \mathbb{R}^n \ s.t. \ h(x) \leq 0\}$ the set given by nonlinear constraints, so that $\mathcal{F} = \mathcal{X} \cap \mathcal{H}$. Hence, the reduced-upper problem has the form

$$\min_{x \in \mathcal{X} \cap \mathcal{H}} \quad \mathrm{F}(x) \tag{4.1}$$

As in the previous chapters, Assumption 2 and Assumption 4 hold, so that we do not have access to the true evaluations of the function, but the numerical error is bounded. The same notation is used in the following.
In [1], the very same problem is treated, assuming to have access to the true evaluations of F.
In the following, we see the main results in [1] for this framework.

4.1 Preliminary Results

The nonlinearly constrained problem we are considering can be handled by partitioning the constraints into two different sets, the first one defined by general inequality constraints, and the second one consisting of simple bound constraints.

Then, for this kind of problem, we can state necessary optimality conditions that explicitly take into account the presence of these two different sets of constraints. The following propositions extend the results in [4] to the case where inequality constraints and an additional convex set of constraints are present.
This preliminary part concerns the optimality condition considered for the problem in (4.1), rather than the methods adopted.
These results do not rely on the employment of true function's evalutations, hence we can borrow the proofs provided in [1]. In the following, the i-th component of h will go by h_i.

Proposition 4.1.1. *Let $x^* \in \mathcal{H}$ be a local minimum of problem in (4.1). Then, there exist not all zero multipliers ${\lambda_0}^*, {\lambda_1}^*, ..., {\lambda_m}^* \in \mathbb{R}$ with*

$${\lambda_0}^* \geq 0,\ {\lambda_i}^* \geq 0 \text{ and } {\lambda_i}^* h_i(x^*) = 0\ \forall\ i = 1, ..., m$$

s.t. for every $d \in D(x^*)$

$$\max\left\{\zeta^T d\,:\,\zeta \in {\lambda_0}^* \partial F(x^*) + \sum_{i=1}^{m} {\lambda_i}^* \partial h_i(x^*)\right\} \geq 0 \tag{4.2}$$

During the proof of Proposition (4.1.1), in particular, it is proven that an element $\xi \in \lambda_0 \partial F(x^*) + \sum_{i \in I_0(x^*)} \lambda_i \partial g_i(x^*)$ such that $\zeta^T d \geq 0$ for all $d \in D(x^*)$ exists.
Hence, the following result follows.

Lemma 4.1.1. *Let $x^* \in \mathcal{H}$ be a local minimum of the problem in (4.1). Then, multipliers ${\lambda_0}^*, {\lambda_1}^*, ..., {\lambda_m}^* \in \mathbb{R}$ with*

$${\lambda_0}^* \geq 0,\ {\lambda_i}^* \geq 0 \text{ and } {\lambda_i}^* h_i(x^*) = 0\ \forall\ i = 1, ..., m,$$

and a vector $\bar{\xi} \in {\lambda_0}^* \partial F(x^*) + \sum_{i=1}^{m} {\lambda_i}^* \partial h_i(x^*)$ exists such that

$$\bar{\xi}^T d \geq 0$$

for every $d \in D(x^*)$.

As usual, by adding a version of the Mangasarian–Fromowitz constraint qualification condition for nonsmooth problems, we can prove KKT's necessary optimality conditions.

Corollary 4.1.1. *Let $x^* \in \mathcal{H}$ be a local minimum of problem (4.1) and assume that a direction $d \in D(x^*)$ exists such that for all $i \in \{1, ..., m\,:\,h_i(x^*) = 0\}$,*

$$(\xi^{h_i})^T d < 0\ \forall\ \xi^{h_i} \in \partial h_i(x^*). \tag{4.3}$$

Then, there exist multipliers ${\lambda_1}^*, ..., {\lambda_m}^* \in \mathbb{R}$ with

$$\lambda_i{}^* \geq 0 \text{ and } \lambda_i{}^* h_i(x^*) = 0 \;\forall\; i = 1, ..., m$$

such that for every $d \in D(x^*)$

$$\max\left\{\xi^T d \,:\, \xi \in \partial F(x^*) + \sum_{i=1}^{m} \lambda_i{}^* \partial h_i(x^*)\right\} \geq 0 \tag{4.4}$$

As regards the stationarity conditions for problem (4.1), taking into account the above results, we can now give the following definition.

Definition 4.1.1. Given problem (4.1), the feasible point $\bar{x}$ is a stationary point of (4.1) if multipliers $\bar{\lambda}_1, ..., \bar{\lambda}_m \in \mathbb{R}$ exist, with

$$\bar{\lambda}_i \geq 0 \text{ and } \bar{\lambda}_i h_i(\bar{x}) = 0 \;\forall\; i = 1, ..., m$$

such that for every $d \in D(\bar{x})$

$$\max\left\{\xi^T d \,:\, \xi \in \partial F(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \partial h_i(\bar{x})\right\} \geq 0 \tag{4.5}$$

4.2 The penalty approach

Given problem (4.1), we introduce the penalty function

$$\mathcal{Z}_\varepsilon(x) = F(x) + \frac{1}{\varepsilon}\sum_{i=1}^{m} \max\{0, h_i(x)\}$$

and define the penalized problem

$$\min_{x \in \mathcal{X}} \quad \mathcal{Z}_\varepsilon(x) \tag{4.6}$$

Remark 4.2.1. Observe that since F and h_i, $i = 1, ..., m$, are Lipschitz continuous, with Lipschitz constants L_F and L_{g_i} , $i = 1, ..., m$, the penalty function $\mathcal{Z}_\varepsilon$ is Lipschitz continuous too, with Lipschitz constant

$$L \leq L_F + \frac{1}{\varepsilon}\sum_{i=1}^{m} L_{h_i}$$

where we denote with L_{h_i} the Lipschitz constant of h_i.

Remark 4.2.2. Note that problem (4.6), for any $\varepsilon > 0$, has the same structure and properties as problem (2.1).

We further note that our penalty approach differs from the ones previously proposed in the literature (see, e.g., [12] and references therein), since only the general nonlinear constraints are penalized. The minimization of the penalty function is then carried out on the set defined by the bound constraints. We

report in the following proposition the equivalence between problem (4.6) and the nonlinearly constrained problem (4.1).
In order to carry out the theoretical analysis, we use an extended version of the Mangasarian–Fromowitz constraint qualification condition for nonsmooth problems.

Assumption 8. *Given problem* (4.1)*, for any* $x \in \mathcal{X} \setminus \mathcal{H}$ *a direction* $d \in D(x)$ *exists such that*

$$(\xi^{h_i})^T d < 0$$

for all $\xi^{h_i} \in \partial h_i(x)$*,* $i \in \{1, ..., m : h_i(x) \geq 0\}$*.*

We are ready for a crucial result in our convergence analysis.

Proposition 4.2.1. *Let Assumption 8 hold. Given problem* (4.1) *and considering problem* (4.6)*, a threshold value* $\varepsilon^* > 0$ *exists such that for every* $\varepsilon \in (0, \varepsilon^*]$*, every Clarke–Jahn stationary point* $\bar{x}$ *of problem* (4.6) *is stationary (according to Definition 4.1.1) for problem* (4.1)*.*

We will get back soon with the proof of the result stated above.

Now we report the algorithm adopted for solving problem (4.6), which is obtained from Algorithm DSN$_{simple}$ by replacing F with $\mathcal{Z}_\varepsilon$ for given $\varepsilon > 0$.

Algorithm DSN$_{con}$

Data. Choose $x_0 \in \mathcal{X}$, $\theta \in (0, 1)$, $A \geq 0$ stepsize lower bound, $\tilde{\alpha}_0 \in (A, +\infty)$, $\{d_k\}_k$ as in Algorithm DSN$_{simple}$.

Step 0. $k = 0$.

Step1. Compute α_k and $\tilde{\alpha}_k$ by the Projected Continuous Search.

Step2. If $\alpha_k = 0$ then set $\tilde{\alpha}_{k+1} = \max\{A, \tilde{\alpha}_k\}$
Else set $\tilde{\alpha}_{k+1} = \alpha_k$ and $x_{k+1} = [x_k + \tilde{\alpha}_k d_k]_{[l,u]}$.
Set $k = k + 1$ and go to Step 1.

In the following, the Projected Continuous Search is presented, which basically coincides with the one presented in the box-constrained framework, but F is substituted by $\mathcal{Z}_\varepsilon$.
Following the same spirit adopted in the previous chapters, we denote

$$\tilde{\mathcal{Z}}_\varepsilon(x) = \tilde{F}(x) + \frac{1}{\varepsilon} \sum_{i=1}^{m} \max\{0, h_i(x)\}$$

Projected Continuous Search($\tilde{\alpha}, y, p; \alpha, p^+$)

$\gamma > 0$, $\delta \in (0, 1)$.

Step0. Set $\alpha = \tilde{\alpha}$.

Step1. If $\tilde{\mathcal{Z}}_\varepsilon\left([y + \alpha p]_{[l,u]}\right) \leq \tilde{\mathcal{Z}}_\varepsilon(y) - \gamma(\alpha)^2$ then set $p^+ = p$ and go to Step 4.

Step2. If $\tilde{\mathcal{Z}}_\varepsilon\left([y - \alpha p]_{[l,u]}\right) \leq \tilde{\mathcal{Z}}_\varepsilon(y) - \gamma(\alpha)^2$ then set $p^+ = p$ and go to Step 4.

Step3. Set $\alpha = 0$, return α and $p^+ = p$.

Step4. Let $\beta = \alpha/\delta$.

Step5. If $\tilde{\mathcal{Z}}_\varepsilon\left([y + \alpha p^+]_{[l,u]}\right) > \tilde{\mathcal{Z}}_\varepsilon(y) - \gamma(\beta)^2$ return α, p^+.

Step6. Set $\alpha = \beta$ and go to Step 4.

Remark 4.2.3. Observe that Algorithm DSN$_{con}$ can be used to solve the constrained problem (4.1) provided that the penalty parameter ε is sufficiently small, as the following proposition states.

Proposition 4.2.2. *Let Assumption 8 hold and let $\{x_k\}_k$ be the sequence produced by Algorithm DSN$_{con}$. Let $\bar{x}$ be any limit point of $\{x_k\}_k$ and K be the subset of indices such that*

$$\lim_{k \to +\infty, k \in K} x_k = \bar{x}$$

If the subsequence $\{d_k\}_{k \in K}$ is dense in the unit sphere, then a threshold value ε^* exists such that for all $\varepsilon \in (0, \varepsilon^*]$, $\bar{x}$ is stationary for problem (4.1).

Proof. The proof follows from Propositions 3.2.5 and 4.2.1. □

We saw that Proposition 4.2.1 was crucial for the convergence analysis of the proposed algorithm.
In the following, we first introduce some technical results and then we will proceed to prove Proposition 4.2.1.
We first prove that any Clarke stationary point of problem (4.6) is stationary for problem (4.1). Then we give the proof of Proposition 3.6.

We begin by recalling, from [7], the definition of Clarke stationary point for a bound constrained problem, namely, a point $\bar{x} \in \mathcal{X}$ such that

$$\mathcal{Z}_\varepsilon^{Cl}(\bar{x}, d) \geq 0 \;\forall\; d \in D(\bar{x})$$

Furthermore, we assume throughout this section that Assumption 8 holds.
In the following, a first key result is stated, whose proof can be found in [1].

Proposition 4.2.3. *Given problem* (4.1) *and considering problem* (4.6)*, a threshold value* $\varepsilon^* > 0$ *exists such that, for every* $\varepsilon \in (0, \varepsilon^*]$*, the function* $\mathcal{Z}_\varepsilon$ *has no Clarke stationary points in* $\mathcal{X} \setminus \mathcal{H}$*.*

Now we report three further results concerning the exactness of $\mathcal{Z}_\varepsilon(x)$ from [12].

Proposition 4.2.4. *A threshold value* $\varepsilon^* > 0$ *exists such that for any* $\varepsilon \in (0, \varepsilon^*]$*, every local minimum point of problem* (4.6) *is also a local minimum point of problem* (4.1)*.*

Proposition 4.2.5. *A threshold value* $\varepsilon^* > 0$ *exists such that for any* $\varepsilon \in (0, \varepsilon^*]$*, every global minimum point of problem* (4.6) *is also a global minimum point of problem* (4.1)*, and conversely.*

In order to give stationarity results for problem (4.6), we have the following proposition.

Proposition 4.2.6. *For any* $\varepsilon > 0$*, every Clarke stationary point* $\bar{x}$ *of problem* (4.6)*, such that* $\bar{x} \in \mathcal{H}$*, is also a stationary point of problem* (4.1)*.*

Proof. Since $\bar{x}$ is, by assumption, a Clarke stationary point of problem (4.6), then, by definition of Clarke stationarity, we know that for all $d \in D(\bar{x})$,

$$\max\left\{\xi^T d : \xi \in \partial \mathcal{Z}_\varepsilon(\bar{x})\right\} \geq 0$$

that is, for all $d \in D(\bar{x})$ there exists $\xi_d \in \partial \mathcal{Z}_\varepsilon(\bar{x})$ such that $(\xi_d)^T d \geq 0$. Now, we recall that

$$\partial \mathcal{Z}_\varepsilon(x) \subseteq \partial F(x) + \frac{1}{\varepsilon} \sum_{i \in I(x)} \beta_i \partial h_i(x)$$

for some β_i, $i \in I(x)$, such that $\sum_{i \in I}(x)\beta_i = 1$ and $\beta_i \geq 0$ for all $i \in I(x)$. Hence, we have that $\xi_d \in \partial F(\bar{x}) + \frac{1}{\varepsilon}\sum_{i \in I}(\bar{x})\beta_i \partial h_i(\bar{x})$. Then, denoting $\lambda_i = \beta_i/\varepsilon$, $i \in I(\bar{x})$, we can write for all $d \in D(\bar{x})$,

$$\max\left\{\xi^T d \,:\, \xi \in \partial F(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \partial h_i(\bar{x})\right\} \geq 0$$

□

Finally, we can prove Proposition 4.2.1.

Proof of Proposition 4.2.1. Since $\bar{x}$ is Clarke-Jahn stationary for problem (4.6), we have, by definition,

$$\mathcal{Z}_\varepsilon^\circ(\bar{x}, d) \geq 0 \ \ \forall\, d \in D(\bar{x}). \tag{4.7}$$

Then, we have that

$$\limsup_{y \to \bar{x}} \frac{\mathcal{Z}_\varepsilon(y + td) - \mathcal{Z}_\varepsilon(y)}{t} = \mathcal{Z}_\varepsilon^{Cl} \geq \mathcal{Z}_\varepsilon^\circ \ \forall\, d \in D(\bar{x}),$$

which, by (4.7), gives,

$$\mathcal{Z}_\varepsilon^{Cl}(\bar{x}, d) \geq 0 \ \ \forall\, d \in D(\bar{x}).$$

Now, the proof follows by considering Proposition 4.2.3 and Proposition 4.2.4.
□

Chapter 5

Numerical illustration

In this section, we evaluate the performance of the proposed algorithms on a collection of nonlinear bilevel optimization problems. The proposed algorithms were implemented in Python.
In our implementation, the lower-level problem is solved using the command `minimize` from the library `scipy.optimize`. For the lower-level problem, the tolerance ϵ was set to 10^{-3}. The solvers were evaluated using groups of small-scale bilevel optimization problems which were built from scratch. In particular, quadratic strictly convex functions (in the variable (x, y)) were chosen for both upper and lower-levels for the first group of 30 problems. The dimensionality of the tested instances for the first collection of problems, with respect to the upper-level problem, did not exceed 3 variables.
Later on, the performance of the proposed algorithms will be tested on another stack of 15 problems, where quadratic strictly convex functions are used for both upper and lower-levels, but here it is explored a higher dimensionality, namely 4, 5, 6, with respect to the upper-level variable.

Algorithm DSS, Algorithm DSN_{simple}, and Algorithm CS-DSN were compared with MADS, a Mesh Adaptive direct-search method, in particular with the Orthomads version which uses a modified variant of MADS with orthogonal search directions. This variant differs from the classic version as the polling directions are chosen deterministically, ensuring that the results of a given run are repeatable and it only performs poll steps.
Since both Algorithm DSN_{con} and Orthomads rely on the same penalization approach and the same convergence theory (i.e the notion of stationarity in terms of the Clarke-Jahn directional derivatives), Algorithm DSN_{con} was not compared in the process as we could expect the same results we will obtain comparing Algorithm DSN_{simple} and Orthomads.

The computational analysis is carried out by using well-known tools from the literature, that is data and performance profiles (see, e.g., [8] for further details). We briefly recall their definitions. Given a set S of algorithms and a set P of problems, for $s \in S$ and $p \in P$, let $t_{p,s}$ be the number of function evaluations required by algorithm s on problem p to satisfy the condition

$$\tilde{F}(x_k) \leq \tilde{F}_{\text{low}} + \alpha(\tilde{F}(x_0) - \tilde{F}_{\text{low}}),$$

where $\alpha \in (0, 1)$ and $\tilde{F}_{\text{low}}$ is the best objective function value achieved by any solver on problem p.

Then, the performance and data profiles of solver s are defined as follows:

$$\rho_s(\gamma) = \frac{1}{|P|} \left| \left\{ p \in P : \frac{t_{p,s}}{\min\{t_{p,s'} : s' \in S\}} \leq \gamma \right\} \right|,$$

and

$$d_s(\kappa) = \frac{1}{|P|} \left| \{ p \in P : t_{p,s} \leq \kappa(n_p + 1) \} \right|,$$

where n_p is the dimension of problem p. We used a budget of 1000 upper-level function evaluations in our experiments.
In the following, we compare separately our algorithms with Orthomads.
The parameters during the optimization were set as follows:

- $\theta = \frac{1}{2}$, $\delta = \frac{1}{2}$, $\gamma = 10^{-3}$,
- $\tilde{\alpha}_0^i = \min\left\{1, |x_0^i|\right\}$ where x_0 is the upper-level starting point for Algorithm DSS and with x_0^i we denote the i-th component,
- $\tilde{\alpha}_0 = \min\left\{1, \min_i\{|x_0^i|\}\right\}$ for Algorithm DSN$_{simple}$, Algorithm CS-DSN,
- The parameter α that rules the reasonable optimality of the solver on a given problem in (5) is set to 10^{-3},
- In Algorithm DSN$_{simple}$, Algorithm CS-DSN we implemented a mapping based on the Sobol sequence, which is a pseudorandom generator widely used in practice.

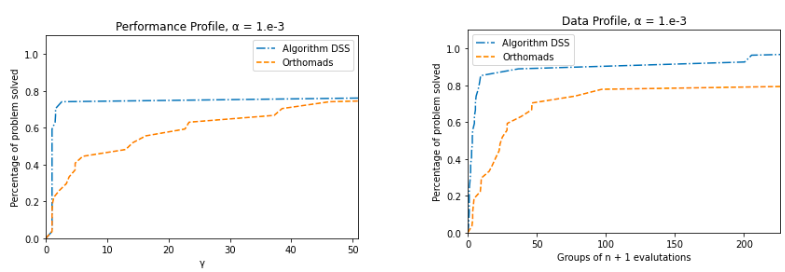

Figure 5.1: Performance and Data Profiles when Algorithm DSS and Orthomads are compared on the first 30 problems, where n is at most 3.

The plots above bring with them remarkable information about how Algorithm DSS performs.
The data profile plot highlights that Algorithm DSS performs better than Orthomads, as it is able to solve most of the problems in the sense of (5). Algorithm DSS looks quite efficient in small dimensions but we should not expect such effectiveness on higher dimensions, because it is a method whose performances are strongly related to the smoothness of the objective F. On the contrary, Orthomads instead relies on the notion of Clarke directional derivatives, and the convergence to a stationary point is guaranteed in a generic nonsmooth framework.
In a generic nonsmooth framework, we should not expect Algorithm DSS to perform well. Indeed, the possibility of reducing the objective function along at least a coordinate direction at each iteration relies completely on the smoothness of F.
In small dimensions, however, even in a nonsmooth context, the explorable space is not wide at all, hence the directions $\pm e_1, ..., \pm e_n$ s capture the local behavior of the objective function.
Moreover, most of the first 30 problems have in the lower-level a quadratic function which is strictly convex in the variable y, and the Implicit Function Theorem ensures that the true solution of the lower-level $y(x)$ is regular, at the least locally. This combined with the choice of smooth upper-level functions lends us the regularity required for the function F of the reduced-upper-level formulation necessary from a theoretical point of view from Algorithm DSS. The performance profile plot shows that Algorithm DSS performs better than Orthomads overall on the first 30 problems.

The following plot compares Algorithm DSN$_{simple}$ and Orthomads on the same collection of 30 problems.

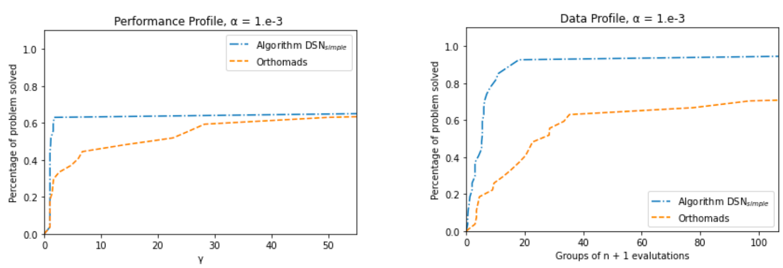

Figure 5.2: Performance and Data Profiles when Algorithm DSN$_{simple}$ and Orthomads are compared on the first 30 problems, where n is at most 3.

In small dimensions, Algorithm DSN$_{simple}$ performs better the Orthomads. In higher dimensions, we should not expect a drop in performance.
Indeed, we derived our explorable direction from a Sobol sequence, inheriting all the suitable properties.

In particular, Sobol sequences are an example of quasi-random low-discrepancy sequences, meaning that they cover the space uniformly.
As a consequence, Algorithm DSN$_{simple}$ does not leave large feasible cones unexplored.
Orthomads relies as well on a dense sequence in the unit sphere as search directions, and it uses at each iteration an orthogonal positive spanning set of polling directions in order to avoid large angles between the 2n directions.

As anticipated, we built another collection where the dimension n of the upper-level gets higher, namely 5 problems where $n = 4$, 5 problems where $n = 5$, and 5 problems where $n = 6$. At least theoretically, we should expect a drop in performance from the first algorithm in favor of Orthomads. In the following, we compare Algorithm DSS with Orthomads.

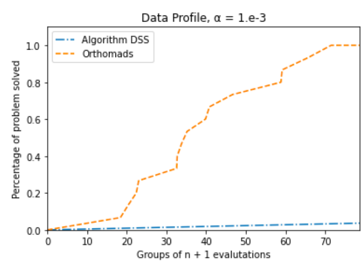

Figure 5.3: Data Profiles when Algorithm DSS and Orthomads are compared on the new collection of problems.

As we could expect, Algorithm DSS is not a good candidate to solve these kind of problems. Now we test the performance of Algorithm DSN$_{simple}$ on the new stack of problems.

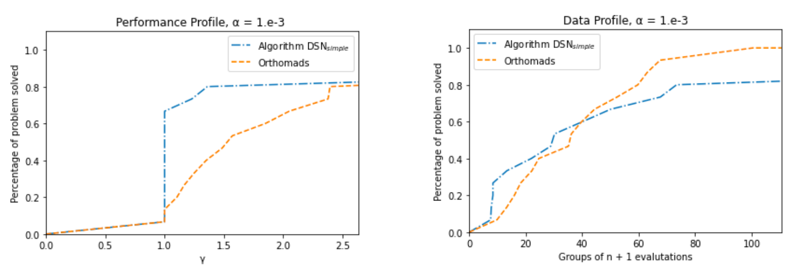

Figure 5.4: Performance and Data Profiles when Algorithm DSN$_{simple}$ and Orthomads are compared on the new collection of problems.

The line search-based approach in Algorithm DSN$_{simple}$ is able to guarantee convergence toward stationary points of the nonsmooth problem, provided that

suitable sequences of search directions $\{d_k\}$ are dense in the unit sphere.
In the second algorithm, a Sobol sequence was considered, opportunely scaled, and translated to ensure the explorable directions lie on the unit sphere.

Algorithm CS-DSN works like Algorithm DSN$_{simple}$, but before exploring a direction of the dense prescribed sequence, it performs a line search on the coordinate direction.
The search is performed on the positive spanning set $\pm e_1, ..., \pm e_n$ and for a general predefined set of directions, we should expect the algorithm to leave less feasible convex cones unexplored, and hence, to perform better.
However, here we are considering a set of explorable directions that allows us to cover uniformly the space, so a better performance from Algorithm CS-DSN is not expected at all.

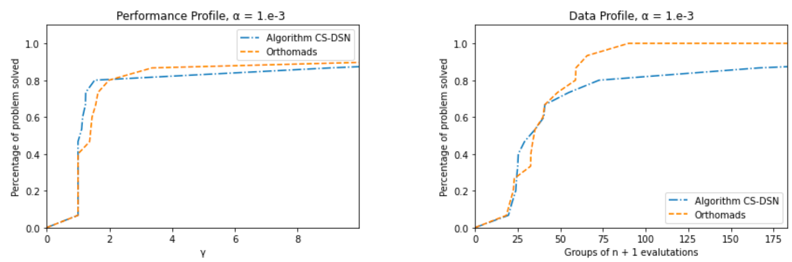

Figure 5.5: Performance and Data Profiles when Algorithm CS-DSN and Orthomads are compared on the new collection of problems.

The plots confirm our intuition, as it is shown Algorithm CS-DSN performs well but does not achieve better results than Orthomads on our 15 problems.

Chapter 6

Conclusions

In this work, we proposed a direct-search approach for bilevel optimization, under the assumption that we have access to the lower-level minimizer up to a bounded numerical error. In particular, we adopted a line search approach based on a sufficient decrease condition.
In the first part of the thesis, we considered problems with only bound constraints on the upper-level variables and we proposed three different algorithms for their solution. Both smooth and nonsmooth frameworks were considered when no first-order information was available.
The main effort was proving that every accumulation point of the sequence of iterates produced by the algorithms is stationary according either to a gradient-related condition or the Clarke–Jahn one.
In the second part of the thesis, we also allowed the presence of nonlinear inequality constraints. We introduced, motivated by the previous case treated, the use of an exact penalty function to transform the given problem into a bound-constrained one, which is solved by adapting the method proposed for the bound-constrained case. Similarly to the bound-constrained case, we were able to prove again that every accumulation point of the generated sequence of iterates is Clarke stationary for the original constrained problem.
Finally, we compared the proposed methods with MADS on two test sets of bound-constrained and nonlinearly constrained nonsmooth problems. The proposed algorithms are parameter-free, hence there is no need to properly set them to ensure convergence of the methods. However, a different choice of these parameters might boost the efficiency of our methods. During the thesis, the main effort was to bring some algorithms that fit in different frameworks, together with a convergence guarantee, so we did not focus on the optimal choice of the predefined parameters that appear in the proposed methods. Future developments will target the choice of the parameters. Also numerical comparisons with recent zeroth order smoothing-based approaches will be considered.

Bibliography

[1] Fasano, Giovanni & Liuzzi, Giampaolo & Lucidi, Stefano & Rinaldi, Francesco. (2014). *A Linesearch-based Derivative-free Approach for Nonsmooth Constrained Optimization*. SIAM Journal on Optimization. 24. 959-992. 10.1137/130940037.

[2] A. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, MPS/SIAM Ser. Optim., SIAM, Philadelphia, 2009.

[3] C. J. Lin, S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone, *Decomposition algorithm model for singly linearly-constrained problems subject to lower and upper bounds*, J. Optim.Theory Appl., 141 (2009), pp. 107-126.

[4] J. B. Hiriart-Urruty, *On optimality conditions in nondifferentiable programming*, Math. Program., 14 (1978), pp. 73–86.

[5] O. L. Mangasarian, *Nonlinear Programming, Classics in Applied Mathematics*, SIAM, Philadelphia, 1994.

[6] Michael I Jordan, Guy Kornowski, Tianyi Lin, Ohad Shamir, and Manolis Zampetakis. *Deterministic nonsmooth nonconvex optimization. arXiv preprint arXiv:2302.08300*, 2023.

[7] F. H. Clarke, *Optimization and Nonsmooth Analysis, John Wiley and Sons*, New York, 1983.

[8] J. J. Moré and S. M. Wild. *Benchmarking derivative-free optimization algorithms*. 20:172–191, 2009.

[9] K. Shimizu, Y. Yshizuka, and J. F. Bard, *Nondifferentiable and Two-Level Mathematical Programming*, Kluwer Academic Publishers, Norwell, MA, 1997.

[10] Y. Yshizuka and K. Shimizu, *Necessary and sufficient conditions for the efficient solutions of nondifferentiable multi-objective problems*, IEEE Trans. Systems Man Cybernet., 14 (1984), pp. 624–629.

[11] O. L. Mangasarian, *Nonlinear Programming, Classics in Applied Mathematics*, SIAM, Philadelphia, 1994.

[12] G. Di Pillo and F. Facchinei, *Exact barrier function methods for Lipschitz programs*, Appl. Math. Optim., 32 (1995), pp. 1–31.

[13] Abramson, Mark & Audet, Charles & Dennis, J. & Le Digabel, Sébastien. (2009). *OrthoMADS: a deterministic MADS instance with orthogonal directions*. SIAM Journal on Optimization.

[14] Lucidi, S., Sciandrone, M. *A Derivative-Free Algorithm for Bound Constrained Optimization*. Computational Optimization and Applications 21, 119–142 (2002).

[15] Charles Audet. *A survey on direct-search methods for blackbox optimization and their applications*. Springer, 2014.

[16] Kolda, Tamara & Lewis, Robert & Torczon, Virginia. (2003). T.G. Kolda, R.M. Lewis, V. Torczon: *Optimization by direct-search: New perspectives on some classical and modern methods*. SIAM Review 45, 385-482. SIAM Review. 45. 385-482. 10.1137/S003614450242889.

[17] Lin, Tianyi & Zheng, Zeyu & Jordan, Michael. (2022). *Gradient-Free Methods for Deterministic and Stochastic Nonsmooth Nonconvex Optimization*. 10.48550/arXiv.2209.05045.