Université Paris-Saclay

Institut de Mathématique d'Orsay

INTERNSHIP REPORT

Conducted at ENSTA Paris

02 April 2024 - 30 August 2024

Algorithms for constrained optimization problems

Nail Baloul

Master 2, Optimization, Université Paris-Saclay

ENSTA Paris
828 Bd des Maréchaux
91120 Palaiseau, France

Internship Tutor (ENSTA)
Professor Sorin-Mihai Grad
Referent Teacher (IMO)
Professor Quentin Mérigot

Version of September 4, 2024

Abstract

This report provides a brief overview of the optimization techniques applied to facility location and portfolio management problems during my internship at ENSTA Paris from April 2 to August 30, under the supervision of Professor Sorin-Mihai Grad. The focus is on tackling these complex optimization challenges through the use of splitting proximal point methods and other approaches that broadly rely on the proximal operator, which offer a robust framework for efficiently solving large-scale optimization problems. The report highlights the methodology, implementation, and outcomes of applying these techniques in practical scenarios, demonstrating their effectiveness and potential applications in real-world contexts.

Contents

List of Figures

List of Tables

Introduction

0.1 Background

In this internship report, I will be describing and reflecting on my five month long internship at ENSTA Paris. This research work which was conducted under the supervision of Sorin-Mihai Grad, Professor at ENSTA Paris, is part of my second-semester curriculum for the Master 2 program in Optimization at Université Paris-Saclay. During the internship, we studied multiple algorithms to solve constrained optimization problems. We insisted on the implementation of methods which enable us to transform our initial constrained problem into an unconstrained optimization problem. Thus, we had to deal with the introduction in the objective function of indicator functions of the feasible regions.
A very important concept on which relies our work is duality theory. Indeed, conjugate duality provides useful results that give valuable insights into the solutions to our considered problems. A crucial tool used exhaustively during our proceedings is the proximal operator which delivers properties and simplifications underpinning our eventual numerical experiments.
Several numerical methods were considered in the context of both facility location problems and entropy optimization. The algorithms presented in this report showcased various degrees of efficiency depending on the scale and the complexity of the problem and the constraints undermining it.

0.2 Objectives

The primary objectives of my work encompass both theoretical insights and a strong focus on numerical applications.

- The first goal was to conduct a in-depth review of optimization methods for solving unconstrained problems, with a particular focus on proximal splitting methods. This included assessing the performance of parallel primal and dual splitting methods specifically in the context of location problems. The analysis involved testing these algorithms across various examples and exploring the potential implementation of inertial methods to enhance performance.

- The second objective was to identify and evaluate alternative methods that not only solve the problem but also potentially offer better results. This involved an explicit comparison of different methods, highlighting their respective advantages and inconveniences. The focus was on two particular methods, the Chambolle-Pock algorithm and the mirror descent method.

- The third objective aimed at generalizing certain theoretical results, particularly those related to the explicit formulation of the proximal operator for some functions involved in the objective. If formulating new results proved challenging, the goal was to clearly explain the difficulties encountered.

- Finally, the fourth aim was to assess the performance of parallel splitting methods in the context of portfolio optimization. This included a comparison of their efficiency with a primal-dual interior point method that has been proposed in the literature.

0.3 Structure of the report

- The first chapter of this report covers all preliminary concepts related to optimization emphasizing the key tools for constrained optimization that will be employed in the subsequent chapters. It provides essential foundational elements, including well-known results and the necessary framework for our analysis.
- The second chapter is dedicated to solving facility location problems and introduces as well as implements numerical methods specifically designed for this purpose. We will also explore certain theoretical results within the same context.
- The third and final chapter presents our work on the application of splitting methods based on the proximal operator in the context of portfolio optimization, along with their numerical implications.

Chapter 1

Preliminaries

While many of the following definitions and results can be applied in broader contexts, such as Banach spaces or even Hausdorff locally convex spaces, we will concentrate on Hilbert spaces to ensure a unified presentation and because our numerical experiments were conducted within this framework. We will identify the dual spaces of the Hilbert spaces with the spaces themselves, in accordance with the Riesz–Fréchet representation theorem.
Let X be a nonempty Hilbert space and set $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$.

1.1 Basic notions related to constrained optimization

Let $f : X \to \bar{\mathbb{R}}$ a be an extended real-valued function.

Definition 1.1.1. Let A be a subset of X.

- The indicator function of A denoted $\delta_A : X \to \bar{\mathbb{R}}$ is defined by

$$\delta_A(x) := \begin{cases} 0, & \text{if } x \in A, \\ +\infty, & \text{otherwise.} \end{cases}$$

- The support function of A denoted $\sigma_A : X \to \bar{\mathbb{R}}$ is defined by

$$\sigma_A(x) := \sup_{x^* \in A} \langle x^*, x \rangle.$$

Definition 1.1.2.
- The domain of f is $\mathrm{dom} f := \{x \in X : f(x) < +\infty\}$.
- The graph of f is $\mathrm{gra} f := \{(x, t) \in X \times \mathbb{R} : f(x) = t\}$.
- The epigraph of f is $\mathrm{epi} f := \{(x, t) \in X \times \mathbb{R} : f(x) \leq t\}$.

Definition 1.1.3.
- The function f is proper if for all $x \in X$, $f(x) > -\infty$ and $\mathrm{dom} f \neq \emptyset$.
- f is convex if its epigraph is a convex subset of $X \times \mathbb{R}$.
  Equivalently, f is convex if

$$\forall x, y \in X, \quad \forall t \in [0, 1], \quad f((1-t)x + ty) \leq (1-t)f(x) + tf(y).$$

- f is lower semicontinuous if its epigraph is a closed subset of $X \times \mathbb{R}$.
  Equivalently, f is lower semicontinuous if

$$\forall x \in X, \quad \liminf_{y \to x} f(y) \geq f(x).$$

Remark 1. For $\sigma > 0$, we call the function f σ-strongly convex if it satisfies

$$\forall x, y \in X, \quad \forall t \in [0,1], \quad f\left((1-t)x + ty\right) + \frac{\sigma}{2}(1-t)t\|x-y\|^2 \leq (1-t)f(x) + tf(y).$$

Remark 2. We will denote by $\Gamma_0(X)$ the set of all proper, convex and lower semicontinuous functions defined on X.

Definition 1.1.4. Let $A : X \to 2^X$ be a multivalued operator.

- The set of all fixed points of A is $\mathrm{Fix}A = \{x \in X : x \in Ax\}$.
- The set of all zeros of A is $\mathrm{zer}A = \{x \in X : 0_X \in Ax\}$.
- The set of all minimizers of f is $\mathrm{Argmin}f = \{x \in X : \forall y \in X, f(x) \leq f(y)\}$.

Definition 1.1.5. Let $f : X \to \bar{\mathbb{R}}$ be proper. The subdifferential of f is the set-valued operator defined by

$$\begin{aligned} \partial f : \quad & X \to 2^X \\ & x \mapsto \left\{x^* \in X : \forall y \in X, f(y) \geq f(x) + \langle x^*, y - x\rangle\right\}. \end{aligned}$$

f is subdifferentiable at x if $\partial f(x) \neq \emptyset$. In that case, x^* is a subgradient of f at x.

Theorem 1.1.6. *(Fermat's rule) Let $f : X \to \bar{\mathbb{R}}$ be proper. Then*

$$\mathrm{Argmin}f = \mathrm{zer}\partial f = \left\{x \in X : 0_X \in \partial f(x)\right\}. \tag{1.1}$$

Let $f : X \to \bar{\mathbb{R}}$. We introduce the notion of conjugate function of f in order to deal with the general duality theory for convex optimization problems [8, Chapter 3].

Definition 1.1.7. The Fenchel conjugate function of f is defined by

$$\begin{aligned} f^* : \quad & X \to \bar{\mathbb{R}} \\ & x^* \mapsto \sup_{x \in X} \left\{\langle x^*, x\rangle - f(x)\right\}. \end{aligned}$$

Remark 3. It is clear that

$$f^*(x^*) = \sup_{x \in \mathrm{dom}f} \left\{\langle x^*, x\rangle - f(x)\right\}.$$

We can similarly introduce the Fenchel conjugate function of f with respect to the nonempty set $A \subseteq X$ by

$$\begin{aligned} f^*_A : \quad & X \to \bar{\mathbb{R}} \\ & x^* \mapsto (f + \delta_A)^*(x) = \sup_{x \in A} \left\{\langle x^*, x\rangle - f(x)\right\}. \end{aligned}$$

Proposition 1.1.8. *[8, Proposition 2.3.2] Let $f : X \to \bar{\mathbb{R}}$ be a proper function. Then we have the following Young-Fenchel inequality*

$$\forall\, (x, x^*) \in X^2, \quad f(x) + f^*(x^*) \geq \langle x^*, x\rangle. \tag{1.2}$$

Proposition 1.1.9. *[8, Proposition 2.3.2] Let $f : X_1 \times \cdots \times X_m \to \bar{\mathbb{R}}$ be a function defined by $f(x_1, \cdots, x_m) = \sum_{i=1}^m f_i(x_i)$, where for $i = 1, \cdots, m$, X_i is a Hilbert space, in which case f is separable. Then, the conjugate function of f is given by*

$$\forall (x_1^*, \cdots, x_m^*) \in X_1 \times \cdots \times X_m, \quad f^*(x_1^*, \cdots, x_m^*) = \sum_{i=1}^m f_i^*(x_i^*). \tag{1.3}$$

Definition 1.1.10. The biconjugate function of f is defined by

$$\begin{aligned} f^{**} : \quad & X \to \bar{\mathbb{R}} \\ & x \mapsto \sup_{x \in X} \{\langle x^*, x\rangle - f^*(x^*)\}. \end{aligned}$$

Remark 4. A direct consequence of the Young-Fenchel inequality is that

$$\forall x \in X, \quad f^{**}(x) \leq f(x).$$

Theorem 1.1.11. *[8, Theorem 2.3.5] If $f \in \Gamma_0(X)$ then f^* is proper and $f = f^{**}$.*

Remark 5. [8, Theorem 2.3.6] A stronger statement of the previous theorem is given by the equivalence

$$f = f^{**} \iff f \in \Gamma(X).$$

Proposition 1.1.12. *[8, Theorem 2.3.17] Let $f : X \to \bar{\mathbb{R}}$ an extended real-valued function and $x \in X$.*

1. *$x^* \in \partial f(x) \implies x \in \partial f^*(x^*)$,*
2. *If $f = f^{**}$ then $x^* \in \partial f(x) \iff x \in \partial f^*(x^*)$,*
3. *If $f \in \Gamma_0(X)$ then $x^* \in \partial f(x) \iff x \in \partial f^*(x^*)$.*

Theorem 1.1.13. *[5, Theorem 16.23] Let $f : X \to \bar{\mathbb{R}} \in \Gamma_0(X)$. Let $x \in X$ and $x^* \in X^*$. Then*

$$x^* \in \partial f(x) \iff f(x) + f^*(x^*) = \langle x^*, x\rangle \iff x \in \partial f^*(x^*). \tag{1.4}$$

Corollary 1.1.14. *[5, Corollary 16.24] Let $f \in \Gamma_0(X)$. Then $(\partial f)^{-1} = \partial f^*$.*

We introduce next the concept of infimal convolution, which will prove useful in addressing facility location problems in the upcoming chapter.

Definition 1.1.15. [5, Definition 12.1] Let $f, g : X \to \bar{\mathbb{R}}$ be two functions. The infimal convolution of f and g is defined by

$$\begin{aligned} f \ \square \ g : \quad & X \to \bar{\mathbb{R}} \\ & x \mapsto \inf_{y \in X} \{f(y) + g(x - y)\}. \end{aligned}$$

It is exact at a point $x \in X$ if

$$f \ \square \ g(x) = \min_{y \in X} \{f(y) + g(x - y)\}.$$

$f \ \square \ g$ is exact if it is exact at every point of its domain in which case we denote it $f \ \boxdot \ g$.

Remark 6. More generally, considering $f_i : X \to \bar{\mathbb{R}}, i = 1, \cdots, m$, we define the infimal convolution

$$\begin{aligned} f_1 \,\Box\, \cdots \,\Box\, f_m : \quad & X \to \bar{\mathbb{R}} \\ & x \mapsto \inf_{\substack{x_i \in X, \quad i=1,\cdots,m, \\ \sum_{i=1}^m x_i = x}} \left\{ \sum_{i=1}^m f_i(x_i) \right\}. \end{aligned}$$

Proposition 1.1.16. *[5, Proposition 12.11] Let $f, g : X \to \bar{\mathbb{R}}$ be convex functions. Then the infimal convolution $f \,\Box\, g$ is convex.*

Corollary 1.1.17. *[5, Corollary 12.12] If C is a convex subset of X, then the distance function d_C is convex.*

Definition 1.1.18. [5, Definition 12.33] Let Y be a Hilbert space and $L \in \mathcal{L}(X, Y)$ is a continuous linear operator. The infimal postconvolution of f by L is defined by

$$\begin{aligned} L \rhd f : \quad & Y \to \bar{\mathbb{R}} \\ & y \mapsto \inf_{x \in X, Lx = y} f(x). \end{aligned}$$

It is exact at a point $y \in Y$ if

$$L \rhd f(y) = \min_{x \in X, Lx = y} f(x).$$

$L \rhd f$ is exact if it is exact at every point of its domain in which case we denote $f \mathbin{\boxed{\rhd}} g$.

Proposition 1.1.19. *[8, Proposition 2.3.8]*

1. *Let Y be a Hilbert space, $L : X \to Y$ a continuous linear operator and $f : X \to \bar{\mathbb{R}}$ an extended real-valued function. Then*
$$(Af)^* = f^* \circ A^*. \tag{1.5}$$

2. *Let $f_i : X \to \bar{\mathbb{R}}, i = 1, \ldots, m$, given functions. Then*
$$(f_1 \,\Box\, \cdots \,\Box\, f_m)^* = \sum_{i=1}^m f_i^*. \tag{1.6}$$

Proposition 1.1.20. *[5, Proposition 16.32] Let Y be a Hilbert space and $f \in \Gamma_0(X)$ and $g \in \Gamma_0(Y)$ be two functions. Let $L \in \mathcal{B}(X, Y)$ be a bounded linear operator such that $L(\mathrm{dom} f) \cap \mathrm{dom} g \neq \emptyset$. Suppose that $(f + g \circ L)^* = f^* \,\boxdot\, (L^* \mathbin{\boxed{\rhd}} g^*)$. Then*

$$\partial(f + g \circ L) = \partial f + L^* \circ (\partial g) \circ L. \tag{1.7}$$

Remark 7. In particular, letting $f, g \in \Gamma_0(X)$ such that $\mathrm{dom} f \cap \mathrm{dom} g \neq \emptyset$ and assuming that $(f + g)^* = f^* \,\boxdot\, g^*$, we deduce that $\partial(f + g) = \partial f + \partial g$.

A similar result follows using the notion of strong quasi-relative interior.

Definition 1.1.21. Let $A \subseteq X$ be a subset of X. We define the following notions of generalized interior.

$$\begin{aligned} \mathrm{ri} A &= \{x \in A : \exists \epsilon > 0 : B(x, \epsilon) \cap \mathrm{aff}(A) \subset A\}, \\ \mathrm{sqri} A &= \{x \in A : \mathrm{cone}(A - x) \text{ is a closed linear subspace of } X\} \end{aligned}$$

are respectively the relative interior and the strong quasi-relative interior of the set A.

Remark 8. If $X = \mathbb{R}^n$ and $A \subseteq \mathbb{R}^n$, then $\text{sqri}A = \text{ri}A$.

Theorem 1.1.22. *[5, Theorem 16.37] Let Y be a Hilbert space, $f \in \Gamma_0(X)$ and $g \in \Gamma_0(Y)$ be two fonctions and $L \in \mathcal{B}(X, Y)$ be a bounded linear operator. Suppose that $0 \in \text{sqri}(\text{dom}g - L(\text{domf}))$. Then $\partial(f + g \circ L) = \partial f + L^* \circ (\partial g) \circ L$.*

Remark 9. [5, Remark 16.39] In particular, let $f, g \in \Gamma_0(X)$. Suppose that $0 \in \text{sqri}(\text{dom}g - \text{dom}f)$. Then $\partial(f + g) = \partial f + \partial g$.

Definition 1.1.23. [5, Definition 12.20] Let $f : X \to \bar{\mathbb{R}}$ be a proper, convex and lower semicontinuous function and let $\gamma > 0$. The Moreau envelope of f of parameter γ is defined by

$$\begin{aligned} {}^{\gamma}f : \quad & X \to \bar{\mathbb{R}} \\ & x \to \left(f \Box \frac{1}{2\gamma}\|.\|^2\right)(x). \end{aligned} \tag{1.8}$$

Definition 1.1.24. Let $f \in \Gamma_0(X)$ and let $\gamma > 0$. The proximal (or proximity) operator of f of parameter γ is defined by

$$\begin{aligned} \text{prox}_{\gamma f} : \quad & X \to X \\ & x \mapsto \text{argmin}_{y \in X} \left\{ f(y) + \frac{1}{2\gamma}\|x - y\|^2 \right\}. \end{aligned} \tag{1.9}$$

Remark 10. [5, Remark 12.24] For $f \in \Gamma_0(X)$ and $\gamma > 0$, it yields that

$${}^{\gamma}f(x) = f(\text{prox}_{\gamma f}(x)) + \frac{1}{2\gamma}\|x - \text{prox}_{\gamma f}(x)\|^2.$$

Example 1.1.25. [5, Example 12.25] Let C be a nonempty closed and convex subset of X. Then

$$\text{prox}_{i_C} = \text{P}_C. \tag{1.10}$$

Proposition 1.1.26. *[5, Proposition 16.34] Let $f \in \Gamma_0(X)$ and $(x, p) \in X^2$. Then*

$$p = \text{prox}_f(x) \iff x - p \in \partial f(p), \tag{1.11}$$

or equivalently,

$$\text{prox}_f = (\text{Id} + \partial f)^{-1}.$$

Definition 1.1.27. Let $A : X \to 2^X$ be a multivalued operator and let $\gamma > 0$. The resolvant of the operator A and its Yoshida approximation of index γ are respectively given by

$$J_A = (\text{Id} + A)^{-1} \text{ and } {}^{\gamma}A = \gamma^{-1}(\text{Id} - J_{\gamma A}).$$

Remark 11. [5, Example 23.3] Let $f \in \Gamma_0(X)$ and let $\gamma > 0$. Then

$$J_{\gamma \partial f} = \text{prox}_{\gamma f} \text{ and } {}^{\gamma}(\partial f) = \nabla({}^{\gamma}f).$$

Proposition 1.1.28. *Let $f : X_1 \times \cdots \times X_m \to \bar{\mathbb{R}}$ be a function defined by $f(x_1, \cdots, x_m) = \sum_{i=1}^m f_i(x_i)$ where, for $i = 1, \cdots, m$, X_i is a Hilbert space and $f_i : X_i \to \bar{\mathbb{R}}$ is a proper, convex and lower semicontinuous function. Then*

$$\forall x = (x_1, \cdots, x_m) \in X_1 \times \cdots \times X_m, \quad \text{prox}_f(x) = \left(\text{prox}_{f_1}(x_1), \cdots, \text{prox}_{f_m}(x_m)\right). \tag{1.12}$$

Proposition 1.1.29. *[5, Proposition 12.26] Let $f \in \Gamma_0(X)$ and let $(x, p) \in X^2$. Then*

$$p = \mathrm{prox}_f(x) \iff \forall y \in H, \langle y - p, x - p \rangle + f(p) \le f(y).$$

Proposition 1.1.30. *[5, Proposition 14.3] Let $f \in \Gamma_0(X)$ and $\gamma > 0$. Then, the extended Moreau decomposition formula is given by*

$$\mathrm{Id} = \mathrm{prox}_{\gamma f} + \gamma \mathrm{prox}_{\gamma^{-1} f^*} \circ \gamma^{-1}\mathrm{Id}. \tag{1.13}$$

Moreover, one has

$$\forall x \in X, \quad f(\mathrm{prox}_{\gamma f}(x)) + f^*(\mathrm{prox}_{\gamma^{-1} f^*}(\gamma^{-1}x)) = \left\langle \mathrm{prox}_{\gamma f}(x), \mathrm{prox}_{\gamma^{-1} f^*}(\gamma^{-1}x) \right\rangle. \tag{1.14}$$

Proposition 1.1.31. *[5, Proposition 12.28] Let $f \in \Gamma_0(X)$. Then*

$$\mathrm{Fixprox}_f = \mathrm{Argmin} f. \tag{1.15}$$

Proposition 1.1.32. *[5, Proposition 12.29] Let $f \in \Gamma_0(X)$ and let $\gamma > 0$. Then ${}^\gamma f : X \to \mathbb{R}$ is Fréchet differentiable on X and its gradient*

$$\nabla({}^\gamma f) = \gamma^{-1}(\mathrm{Id} - \mathrm{prox}_{\gamma f}) \tag{1.16}$$

is γ^{-1}-Lipschitz continuous.

Corollary 1.1.33. *[5, Corollary 12.30] Let C be a nonempty closed and convex subset of X. Then d_C^2 is Fréchet differentiable on X and*

$$\nabla({}^\gamma f) = 2(\mathrm{Id} - \mathrm{P}_C). \tag{1.17}$$

1.2 Duality

Let Y be a Hilbert space and suppose that $f : X \to \bar{\mathbb{R}}$ and $g : Y \to \bar{\mathbb{R}}$ are proper functions. Let $A \in \mathcal{L}(X, Y)$ be a continuous linear operator such that

$$\mathrm{dom} f \cap A^{-1}(\mathrm{dom} g) \neq \emptyset.$$

Let us consider the following optimization problem

$$\inf_{x \in X} \{f(x) + g(Ax)\}. \tag{P^A}$$

Then, by virtue of perturbation theory [8, Section 3.1], a dual formulation of (P^A) is given by

$$\sup_{y^* \in Y} \{-f^*(-A^*y^*) - g^*(y^*)\}. \tag{D^A}$$

Example 1.2.1. 1. If $X = Y$ and $A = \mathrm{Id}$, the primal problem writes

$$\inf_{x \in X} \{f(x) + g(x)\} \tag{P^{Id}}$$

and its dual formulation is given by

$$\sup_{y^* \in X} \{-f^*(-y^*) - g^*(y^*)\}. \tag{D^{Id}}$$

2. If $f = 0$, the primal problem writes
$$\inf_{x \in X} \{g(x)\} \tag{P^{A_g}}$$
and duality yields
$$\sup_{\substack{y^* \in X, \\ A^* y^* = 0}} \{-g^*(y^*)\}. \tag{D^{A_g}}$$

3. If $g(x) = \sum_{i=1}^m f_i(x_i)$ and $Ax = (x, \cdots, x)$, the primal problem writes
$$\inf_{x \in X} \left\{ \sum_{i=1}^m f_i(x_i) \right\} \tag{P^{Σ}}$$
and its dual counterpart is given by
$$\sup_{\substack{x_{i*} \in X, i=1,\cdots,m, \\ \sum_{i=1}^m x_{i*} = 0}} \left\{ -\sum_{i=1}^m f_i^*(x_{i*}) \right\}. \tag{D^{Σ}}$$

We recall that a subset $K \subseteq X$ is a cone if it satisfies $\forall x \in K, \forall \alpha > 0$, $\alpha x \in K$ and a convex cone if $\forall x, y \in K, \forall \alpha, \beta > 0$, $\alpha x + \beta y \in K$.

Definition 1.2.2. Let $K \subseteq X$ be a convex cone.
K induces on X a partial ordering relation $\preceq_K$ if for $x, y \in X$ it holds $x \preceq_K y \Leftrightarrow y - x \in K$.
In this case, we denote $\preceq_K := \{(x, y) \in X \times X : y - x \in K\}$.

We denote $\bar{X} = X \cup \{+\infty_K\}$ where $+\infty_K$ is the largest element attached to X with respect to $\preceq_K$. Then it holds $x \preceq_K +\infty_K$ for all $x \in X$.

Remark 12. We write $x \prec_K y$ when $x \preceq_K y$ and $x \neq y$.
In particular when $X = \mathbb{R}$ and $K = \mathbb{R}_+$, we have $\leq := \preceq_{\mathbb{R}_+}$ and $< := \prec_{\mathbb{R}_+}$.
We use the convention $0.(+\infty) := +\infty$ and $0.(-\infty) := 0$. Similarly, on $\bar{X}$, we will employ the following convention
$$0.(+\infty_K) := +\infty_K \text{ and } 0.(-\infty_K) := 0$$
We refer to [5] and [8] for all operations and conventions in that regard.

Definition 1.2.3. The dual cone of K is defined as
$$K^* := \{x^* \in X : \langle x^*, x \rangle \geq 0, \forall x \in K\}$$
By convention, we set for all $x^* \in K^*$, $\langle x^*, +\infty_K \rangle := +\infty$.

Definition 1.2.4. Let A be a subset of X. The normal cone of A at $x \in X$ is given as
$$N_A(x) := \begin{cases} \{x^* \in X : \langle x^*, y - x \rangle \leq 0, \quad \forall y \in A\} & \text{if } x \in A, \\ N_A(x) = \emptyset, & \text{otherwise.} \end{cases}$$

Altough our focus will be on scalar optimization, it is necessary to extend the classical concepts of convexity and closedness from scalar functions to vector functions, as the objective function's composition may involve vector-valued components.

Definition 1.2.5. Let $W \subseteq X$ be a nonempty set. A function $f : X \to \mathbb{R}$ is K-increasing on W if

$$\forall x, y \in W, \quad x \preceq_K y \implies f(x) \leq f(y).$$

When $W = X$, the function f is simply K-increasing.

Definition 1.2.6. Let Z a Hilbert space. Assume that Z is partially ordered by the convex cone $Q \subseteq Z$. Set $\bar{Z} := Z \cup \{\pm\infty_Q\}$ and consider a vector function $F : X \to \bar{Z}$.

- The domain of F is defined as dom $F := \{x \in X : F(x) \neq +\infty_Q\}$.
- F is proper if dom $F \neq \emptyset$.
- F is Q-convex if

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

- The Q-epigraph of F is defined as

$$\text{epi}_Q F = \{(x, z) \in X \times Z : F(x) \preceq_Q z\}.$$

- Assume Q is closed. F is Q-epi-closed if $\text{epi}_Q F$ is a closed set.
- For $z^* \in Q^*$, the function $(z^*F) : X \to \mathbb{R}$ is defined by $(z^*F)(x) := \langle z^*, F(x)\rangle$.
  In particular, $\text{dom}(z^*F) = \text{dom}F$.
- F is called positively Q-lower semicontinuous at $x \in X$ if (z^*F) is lower semicontinuous at x for all $z^* \in Q^*$.
  F is called positively Q-lower semicontinuous if it is positively Q-lower semicontinuous at every $x \in X$.
- F is (K, Q)-increasing on W if

$$\forall x, y \in W, x \preceq_K y \implies F(x) \preceq_Q F(y).$$

  When $W = X$, the function F is simply (K, Q)-increasing.

Remark 13. When $Z = \mathbb{R}$ and $Q = \mathbb{R}_+$, the notion of Q-epi-closedness corresponds to that of lower semicontinuity.
If F is positively Q-lower semicontinuous, then it is also Q-epi-closed, while the inverse statement is not generally true.
If F is Q-convex, then (z^*F) is convex for all $z^* \in Q^*$.
By the aforementioned convention, we have

$$\forall f : X \to \mathbb{R}, \quad 0.f = \delta_{\text{dom}f} \text{ and } \forall F : X \to Z, \quad 0_Z.F = \delta_{\text{dom}F}. \tag{1.18}$$

To conclude this preliminary chapter, let us introduce our general optimization problem with geometric and cone constraint. We make the following assumptions.

- $S \subseteq X$ is a given nonempty set.
- Z is a Hilbert space partially ordered by the convex cone $C \subseteq Z$.

- $f : X \to \bar{\mathbb{R}}$ is a proper function.
- $g : X \to \bar{Z} = Z \cup \{\pm\infty_C\}$ is a proper vector function such that

$$\mathrm{dom} f \cap S \cap g^{-1}(-C) \neq \emptyset.$$

We set $\mathcal{A} = \{x \in S : g(x) \in -C\}$ and consider the following constrained optimization problem

$$\inf_{x \in \mathcal{A}} f(x). \tag{$P_{\mathcal{A}}$}$$

Using perturbation theory and depending on the perturbation function considered (and thus on the duality chosen) we obtain the following dual formulations.

1. By Lagrange duality, the Lagrange dual of $(P_{\mathcal{A}})$ writes

$$\sup_{z^* \in Z} \left\{ -\left(f + (z^* g)\right)^*_S(0) - \delta_{-C^*}(z^*) \right\}. \tag{D_L}$$

2. By Fenchel duality, the Fenchel dual of $(P_{\mathcal{A}})$ is given

$$\sup_{y^* \in X} \left\{ -f^*(y^*) - \sigma_{\mathcal{A}}(x^* - y^*) \right\}. \tag{D_F}$$

3. By Fenchel-Lagrange duality, the Fenchel-Lagrange dual of $(P_{\mathcal{A}})$ formulates

$$\sup_{y^* \in X, z^* \in Z} \left\{ -f^*(y^*) - (z^* g)^*_S(-y^*) - \delta_{-C^*}(z^*) \right\}. \tag{D_{FL}}$$

These formulations demonstrate that a convex optimization problem can be paired, through Fenchel-Rockafellar duality, to a dual problem involving the conjugates of the functions present in the primal formulation. The derivation of these formulations will not be detailed here, for a comprehensive explanation we refer to [8, Section 3.1].

Chapter 2

Facility location optimization

A facility location problem is the classical optimization problem for determining the sites for given public facilities such as factories and warehouses based on geographical demands, facility costs, and transportation distances. It consists of choosing the best among potential sites to minimize costs or maximize service to customers or demand points.
The minsum and minimax location problems are two different types of facility location problems that focus on optimizing different objective functions to determine the best locations for facilities. The first problem aims to minimize the total (or average) distance between demand points and their assigned facility (or the nearest facility if multiple are implemented). This minsum problem is particularly relevant when the goal is to minimize the overall transportation cost (or time). Practically, it consits of determining (the coordinates of) a new point such that the sum of all distances between this point and the given points is minimal.
The second problem focuses on minimizing the maximum distance between any demand point and its assigned facility (or the nearest facility if multiple are implemented). This minimax problem is relevant when the goal is to ensure that the worst-case service level (longest distance) is as good as possible, which is important for emergency services (and other critical applications). Practically, it involves finding (the coordinates of) a new point such that the maximum of the aforementioned distances is minimized.
The latter problem tends to be more complex because it involves minimizing the worst-case scenario, which is inherently more difficult. Furthermore, compared to minsum problems, solving minimax problems often requires more computational effort and advanced optimization techniques. Namely, these formulations can become computationally very demanding as the problem's dimensionality increases.
During our internship, we focused on the study of minimax location problems on Hilbert spaces with geometric constraints. Let X be a Hilbert space.

2.1 Basic concepts related to location problems

In facility location problems, the distances considered depend on the context and the objective function. Typically, these distances can be Euclidean distances, gauges or minimal time distances. The gauge distance refers to the straight-line distance between two points in a geometric space, whereas the minimal time distance denotes the shortest travel time between these points.
While gauge distances are often used for theoretical or simplified models, minimal time distances are used for more realistic and application-driven models.

Definition 2.1.1. Let A be a subset of X. The gauge function $\gamma_A : X \to \bar{\mathbb{R}}$ is defined by

$$\gamma_A(x) := \inf \{t > 0 : x \in tA\}.$$

Definition 2.1.2. Let C be a nonempty closed and convex subset of X and $\Omega \subseteq X$ be a nonempty set. The minimal time function associated with the sets C and Ω is defined by

$$\mathcal{T}_{\Omega}^{C}(x) := \inf \{t \geq 0 : x + tC \cap \Omega \neq \emptyset\}.$$

Remark 14. $\mathcal{T}_{\Omega}^{C}(x)$ stands for the minimal time needed to the point x to reach the target set Ω along the constant dynamics C.

- If $C = \bar{B}(0_X, 1)$ is the closed unit ball, then for all $x \in X$, $\mathcal{T}_{\Omega}^{C}(x) = d(x, \Omega) := \inf_{y\in\Omega}\|x - y\|$.
- If $C = \{0_X\}$, then for all $x \in X$, $\mathcal{T}_{\Omega}^{C}(x) = \delta_{\Omega}(x)$.
- If $\Omega = \{0_X\}$ then for all $x \in X$, $\mathcal{T}_{\Omega}^{C}(x) = \gamma(-x)$.

We present the following notion which was introduced in [11] and that generalizes the classical minimal time.

Definition 2.1.3. Let $C \subseteq X$ and $\Omega \subset X$ be nonempty sets and $f : X \to \bar{\mathbb{R}}$ a proper function. The extended perturbed minimal time function $\mathcal{T}_{\Omega,f}^{C} : X \to \bar{\mathbb{R}}$ is defined, for all $x \in X$, as the infimal convolution of γ_C, f and δ_{Ω} by

$$\mathcal{T}_{\Omega,f}^{C}(x) := \inf_{y\in X, z\in X} \{\gamma_C(x - y - z) + f(y) + \delta_{\Omega}(z)\}$$

or equivalently

$$\mathcal{T}_{\Omega,f}^{C}(x) = \inf_{y\in X, z\in\Omega} \{\gamma_C(x - y - z) + f(y)\}. \tag{2.1}$$

Remark 15. If $f = \delta_{0_X}$ and C is closed and convex then $\mathcal{T}_{\Omega,f}^{C}$ reduces to $\mathcal{T}_{\Omega}^{C}$, hence the notion of generalized minimal time function.

In order to simplify the forthcoming notations, we introduce the notion of polar set as provided in [11, Remark 2.1].

Definition 2.1.4. Let C be a nonempty subset of X.

- The polar set of C is defined as

$$C^0 := \{x^* \in X : \sigma_C(x^*) \leq 1\},$$

  where σ_C is the support function of C defined in Definition 1.1.1.

- The dual gauge $\gamma_{C^0} : X \to \bar{\mathbb{R}}$ of the set C is defined for all $x^* \in X$, by

$$\gamma_{C^0}(x^*) := \sup_{x\in C}\langle x^*, x\rangle = \sigma_C(x^*).$$

Remark 16. [11, Remark 2.1] proves that $\gamma_C^* = \delta_{C^0}$ which yields in particular that $\operatorname{dom}\gamma_C^* = \operatorname{dom}\delta_{C^0} = C^0$.

Remark 17. The optimization problems where the objective function involves the extended minimal time function are convex optimization problems. This property justified by Theorem 3.6.1 enables us to refer to the relevant theory dedicated to the convex setting.

Remark 18. We provide two geometric interpretations related to our facility location model, focusing on two specific cases. These instances will be the focus along our upcoming numerical experiments.

1. Assume that in expression (2.1) $f = \gamma_G$ where G is a closed and convex set such that $0_X \in C \cap G$, then the generalized minimal time function rewrites

$$\begin{aligned}\mathcal{T}^{-C}_{\Omega,\gamma_G}(x) &= \inf_{y\in X, z\in\Omega} \{\gamma_{-C}(x-y-z) + \gamma_G(y)\} \\ &= \inf_{\substack{\alpha,\beta>0, y\in X, z\in\Omega,\\ x-y-z\in-\alpha C, y\in\beta G}} \{\alpha+\beta\} \\ &= \inf_{\substack{\alpha,\beta>0, k\in X, z\in\Omega,\\ x-k\in-\alpha C, k-z\in\beta G}} \{\alpha+\beta\} \\ \mathcal{T}^{-C}_{\Omega,\gamma_G}(x) &= \inf_{\alpha,\beta>0, (x+\alpha C)\cap(\Omega+\beta G)\neq\emptyset} \{\alpha+\beta\}, \end{aligned} \tag{2.2}$$

where (2.2) interprets α as the minimal time needed for the given point x to reach the set Ω along the constant dynamics C, while Ω is moving in direction of x with respect to the constant dynamics characterized by the set G. β gives the minimal time needed for Ω to reach x.

2. Assume that $f = \delta_L$ where L is a nonempty closed and convex set, then the generalized minimal time function rewrites

$$\begin{aligned}\mathcal{T}^{-C}_{\Omega,\delta_L}(x) &= \inf_{y\in X, z\in\Omega} \{\gamma_{-C}(x-y-z) + \delta_L(y)\} \\ &= \inf_{y\in L, z\in\Omega} \{\gamma_{-C}(x-y-z)\} \\ &= \inf_{\substack{\lambda>0, y\in L, z\in\Omega,\\ x-y-z\in-\lambda C}} \lambda \\ \mathcal{T}^{-C}_{\Omega,\delta_L}(x) &= \inf_{\lambda>0, (x+\lambda C)\cap(\Omega+L)\neq\emptyset} \lambda, \end{aligned} \tag{2.3}$$

where (2.3) interprets λ as the minimal time needed for the given point x to reach the set $\Omega + L$ along the constant dynamics C.

We can now present the framework of the general minimax location problem studied.

2.2 Position of the problem

Before stating our minimax location problem which involves a generalized minimal time function, we begin by making the following assumptions.

- $S \subseteq X$ is nonempty closed and convex.
- $C_i \subseteq X$ is closed and convex with $0_X \in \text{int}C_i$, $i = 1, \cdots, n$.
- $\Omega_i \subseteq X$ is nonempty convex and compact, $i = 1, \cdots, n$.
- $f_i : X \to \bar{\mathbb{R}}$ is proper, convex and lower semicontinuous, $i = 1, \cdots, n$.

- $h_i : \mathbb{R} \to \bar{\mathbb{R}}$ defined by $h_i(x) := \begin{cases} x, \text{ if } x \in \mathbb{R}_+, \\ +\infty, \text{ otherwise} \end{cases}$ is proper, convex, lower semicontinuous and increasing on $\mathbb{R}_+$, $i = 1, \cdots, n$.
- $a_i \in \mathbb{R}_+$ are nonnegative set-up costs, $i = 1, \cdots, n$.

Consider the following generalized location problem

$$\inf_{x \in S} \max_{1 \le i \le n} \left\{ h_i \left(\mathcal{T}^{C_i}_{\Omega_i, f_i}(x) \right) + a_i \right\}. \tag{$P^S_{h,\mathcal{T}}$}$$

Remark 19. By the above assumptions, it yields that

$$0_{X^*} \in C_i^0 \cap \operatorname{dom} f_i^* \cap \operatorname{dom} \sigma_{\Omega_i}, \quad i = 1, \cdots, n.$$

Moreover, since $\gamma^*_{C_i} = \delta_{C_i^0}$ and σ_{Ω_i} are continuous functions, we get by Theorem 3.6.1 that for every $i = 1, \cdots, n$, $\mathcal{T}^{C_i}_{\Omega_i, f_i}$ is a proper, convex and lower semicontinuous function with full domain and thus continuous.
In addition, since h_i is a proper, convex, lower semicontinuous and increasing function on $\mathbb{R}_+$, we deduce that the objective function of $(P^S_{h,\mathcal{T}})$ is proper, convex and lower semicontinuous which ensures that $(P^S_{h,\mathcal{T}})$ is indeed a convex optimization problem.

We will study $(P^S_{h,\mathcal{T}})$ using conjugate duality which was notably introduced in this context in [13].

2.3 Duality results

In order to adress $(P^S_{h,\mathcal{T}})$ by means of conjugate duality duality, we reformulate it as a multi-composed optimization problem for which a suitable approach is available [10],[13],[15]. Consider the equivalent formulation of problem $(P^S_{h,\mathcal{T}})$ given by

$$\inf_{x \in S} (f \circ F \circ G)(x), \tag{$P'^S_{h,\mathcal{T}}$}$$

where we define the functions

- $f : \mathbb{R}^n \to \mathbb{R}, \quad f(z) := \begin{cases} \max_{1 \le i \le n} \{h_i(z_i) + a_i\} & \text{if } z = (z_1, \cdots, z_n)^T \in \mathbb{R}^n_+, \quad i = 1, \cdots, n, \\ +\infty & \text{otherwise,} \end{cases}$
- $F : X^n \to \mathbb{R}^n, \quad F(y) := \left(\mathcal{T}^{C_1}_{\Omega_1, f_1}(y_1), \cdots, \mathcal{T}^{C_n}_{\Omega_n, f_n}(y_n) \right)^T, \quad y = (y_1, \cdots, y_n)^T,$
- $G : X \to X^n, \quad G(x) := (x, \cdots, x).$

Based on [15, Section 3] where the authors formulated a conjugate dual problem to an optimization problem with geometric and cone constraints whose objective function is a composition of $n+1$ functions, we take here $n = 2$, which leads to the following observations.

- $X_0 := \mathbb{R}^n$ partially ordered by the convex cone $K_0 = \mathbb{R}^n_+$.
- $X_1 := X^n$ partially ordered by the nonempty convex cone $K_1 = 0_{X^n}$.
- $X_2 := X, S \subseteq X_2$.

- f is proper, convex and K_0-increasing on $F(\text{dom}F) + K_0 \subseteq \text{dom } f$.
- $F : X_1 \to X_0 = X_0 \cup \{+\infty_{K_0}\}$ is proper and K_1-K_0-increasing on $G(\text{dom } G \cap S) + K_1 \subseteq \text{dom } F$.
- $G : X_2 \to X_1 = X_1 \cup \{+\infty_{K_1}\}$ is a proper and K_1-convex function.

We use the convention $f(+\infty_{K_0}) = +\infty$ and $F(+\infty_{K_1}) = +\infty_{K_1}$, so that $f : \bar{X}_0 \to \bar{\mathbb{R}}$ and $F : \bar{X}_1 \to \bar{X}_0$ are now considered. Employing the same duality concept as in [15], we get the following conjugate dual

$$\sup_{\substack{z_i^* \in \mathbb{R}_+, w_i^* \in X, \\ i=1,\cdots,n}} \left\{ \inf_{x \in S} \left\{ \sum_{i=1}^{n} \langle w_i^*, x \rangle \right\} - f^*(z^*) - (z^* F)^*(w^*) \right\},$$

where $x \in S$ is the primal variable and $z^* \in \mathbb{R}_+^n$ and $w^* \in X^n$ are the dual variables.
In [13], the authors proved that

$$f^*(z^*) = \min_{\substack{\sum_{i=1}^n \lambda_i \leq 1, \lambda_i \geq 0, \\ i=1,\cdots,n}} \left\{ \sum_{i=1}^{n} \left[(\lambda_i h_i)^* (z_i^*) - \lambda_i a_i \right] \right\},$$

and

$$(z^* F)^*(w^*) = \sum_{i=1}^{n} \left(z_i^* \mathcal{T}_{\Omega_i, f_i}^{C_i} \right)^* (w_i^*),$$

so that the conjugate dual problem to $(P_{h,\mathcal{T}}^S)$ is given by

$$\sup_{\substack{\sum_{i=1}^n \lambda_i \leq 1, \lambda_i \geq 0, \\ z_i^* \in \mathbb{R}_+, w_i^* \in X, \\ i=1,\cdots,n}} \left\{ -\sigma_S \left(\sum_{i=1}^{n} w_i^*, x \right) - \sum_{i=1}^{n} \left[(\lambda_i h_i)^* (z_i^*) - \lambda_i a_i \right] - \sum_{i=1}^{n} \left(z_i^* \mathcal{T}_{\Omega_i, f_i}^{C_i} \right)^* (w_i^*) \right\}, \qquad (D_{h,\mathcal{T}}^S)$$

where $\lambda \in \mathbb{R}_+^n$ is an additionnal dual variable.

Remark 20. If the set-up costs $a_i, i = 1, \cdots, n$, are arbitrary, then the conjugate of f is given by

$$f^*(z^*) = \min_{\substack{\sum_{i=1}^n \lambda_i = 1, \lambda_i \geq 0, \\ i=1,\cdots,n}} \left\{ \sum_{i=1}^{n} \left[(\lambda_i h_i)^* (z_i^*) - \lambda_i a_i \right] \right\},$$

and the dual problem $(D_{h,\mathcal{T}}^S)$ writes

$$\sup_{\substack{\sum_{i=1}^n \lambda_i = 1, \lambda_i \geq 0, \\ z_i^* \in \mathbb{R}_+, w_i^* \in X, \\ i=1,\cdots,n}} \left\{ -\sigma_S \left(\sum_{i=1}^{n} w_i^*, x \right) - \sum_{i=1}^{n} \left[(\lambda_i h_i)^* (z_i^*) - \lambda_i a_i \right] - \sum_{i=1}^{n} \left(z_i^* \mathcal{T}_{\Omega_i, f_i}^{C_i} \right)^* (w_i^*) \right\}.$$

For simplicity, we will continue focusing on the case where non-negative setup costs are considered.

After some computations -which are not detailed here- and introducing, like in [11], the sets defined by $I = \{i \in \{1, \cdots, n\} : z_i^* > 0\}$ and $R = \{r \in \{1, \cdots, n\} : \lambda_r > 0\}$, we rewrite the dual problem $(D^S_{h,\mathcal{T}})$ as

$$\sup_{\substack{\lambda_i, z_i^* \geq 0, i=1,\cdots,n, \\ I \subseteq R \\ w_i^* \in X, \gamma_{C_i^0}(w_i^*) \leq z_i^*, i \in I, \\ \sum_{r\in R} \lambda_r \leq 1}} \left\{ -\sigma_S\left(-\sum_{i\in I} w_i^*\right) - \sum_{r\in R} \lambda_r \left[h_r^*\left(\frac{z_r^*}{\lambda_r}\right) - a_r\right] - \sum_{i\in I}\left[z_i^* f_i^*\left(\frac{1}{z_i^*} w_i^*\right) + \sigma_{\Omega_i}(w_i^*)\right]\right\},$$

which can be equivalently written thanks to [11, Proposition 3.2] as

$$\sup_{\substack{\lambda_i, z_i^* \geq 0, \\ w_i^* \in X, \gamma_{C_i^0}(w_i^*) \leq z_i^*, \\ \sum_i^n \lambda_i \leq 1, i=1,\cdots,n}} \left\{ -\sigma_S\left(-\sum_i^n w_i^*\right) - \sum_{i=1}^n \left[(\lambda_i h_i)^*(z_i^*) - \lambda_i a_i + (z_i^* f_i)^*(w_i^*) + \sigma_{\Omega_i}(w_i^*)\right]\right\}. \quad (\hat{D}^S_{h,\mathcal{T}})$$

Remark 21. In our numerical experiments, we will manipulate dual formulation $(\hat{D}^S_{h,\mathcal{T}})$ as handling with the set I and R introduces the inconvenience of transforming dual problem $(D^S_{h,\mathcal{T}})$ into a discrete optimization problem, making it more challenging to solve.

The weak duality between the primal-dual pair $(P^S_{h,\mathcal{T}})$-$(D^S_{h,\mathcal{T}})$ always holds, $v(P^S_{h,\mathcal{T}}) \geq v(D^S_{h,\mathcal{T}})$. For details regarding this assertion, we refer to [15, Theorem 3.1 and Theorem 3.2] and [8, Theorem 3.1.1]. The proof involves formulating an equivalent primal problem and computing its dual, which will be equivalent to $(D^S_{h,\mathcal{T}})$. We then utilize the Young-Fenchel inequality (1.2) to establish the desired result.
Our aim is now to verify whether strong duality holds.

Theorem 2.3.1 (Strong duality)**.** *Under the assumptions of section 2.2, strong duality holds between the primal problem $(P^S_{h,\mathcal{T}})$ and its dual $(D^S_{h,\mathcal{T}})$, $v(P^S_{h,\mathcal{T}}) = v(D^S_{h,\mathcal{T}})$, and the dual problem has an optimal solution $(\bar{\lambda}, \bar{z}^*, \bar{w}^*) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times X^{|\bar{I}|}$, with the corresponding sets $\bar{I} \subseteq \bar{R} \subseteq \{1, \cdots, n\}$.*

Proof. It suffices to show that regularity conditions [15, (RC_2^C), page 12] are satisfied in which case [15, Theorem 4.1] applies. Indeed, we make the following observations.

- X_0, X_1, X_2 and $Z := X_2$ which is ordered by the trivial cone $Q := X_2 = X$ are Hilbert spaces, f is lower semicontinuous, S is closed.
- Letting $g : X_2 \to \bar{Z} : Z \cup \{+\infty_Q\}$ such that $g(x) := x$, g is X-epi-closed.
- $K_0 = \mathbb{R}^n_+$ is closed with $\operatorname{int} K_0 \neq \emptyset$, and F is K_0-epi-closed.
- $K_1 = 0_{X^n}$ is closed and G is K_1-epi-closed. Although $\operatorname{int} K_1 = \emptyset$, the continuity of G voids the necessity of this condition.

Furthermore, as

$$\begin{aligned}
&\operatorname{sqri}(F(\operatorname{dom} F) - \operatorname{dom} f + K_0) = \operatorname{sqri}(F(\operatorname{dom} F) - \mathbb{R}^n_+ + \mathbb{R}^n_+) = \mathbb{R}^n, \\
&\operatorname{sqri}(G(\operatorname{dom} G \cap \operatorname{dom} g \cap S) - \operatorname{dom} F + K_1) = \operatorname{sqri}(G(X \cap X \cap S) - X^n + 0_{X^n}) = X^n, \\
&\operatorname{sqri}(g(\operatorname{dom} G \cap \operatorname{dom} g \cap S) + Q) = \operatorname{sqri}(g(X \cap X \cap S) + X) = X,
\end{aligned}$$

we have the following requirements fulfilled

$$\begin{aligned}
&0_{\mathbb{R}^n} \in \operatorname{sqri}(F(\operatorname{dom} F) - \operatorname{dom} f + K_0),\\
&0_{X^n} \in \operatorname{sqri}(G(\operatorname{dom} G \cap \operatorname{dom} g \cap S) - \operatorname{dom} F + K_1),\\
&0_X \in \operatorname{sqri}(g(\operatorname{dom} G \cap \operatorname{dom} g \cap S) + Q).
\end{aligned}$$

By virtue of the above mentioned result, we deduce strong duality. In particular, the dual problem $(D^S_{h,\mathcal{T}})$ has a solution and the equality $v(P^S_{h,\mathcal{T}})= v(D^S_{h,\mathcal{T}})$ holds. □

The following result -which is a consequence of Theorem 2.3.1- states the necessary and sufficient optimality conditions in the context of our minimax location problem. We refer to [11] for the proof which relies on standard arguments.

Theorem 2.3.2 (Optimality conditions). *1. Suppose that the assumptions of section 2.2 are fulfilled and let $\bar{x} \in S$ be an optimal solution to the problem $(P^S_{h,\mathcal{T}})$. Then there exists an optimal solution $(\bar{\lambda}, \bar{z}^*, \bar{w}^*) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times X^{|\bar{I}|}$ to the dual problem $(D^S_{h,\mathcal{T}})$ with the corresponding sets $\bar{I} \subseteq \bar{R} \subseteq \{1, \cdots, n\}$ such that*

$$\begin{aligned}
&\max_{1\le j\le n} \left\{ h_j\left(\mathcal{T}^{C_j}_{\Omega_j,f_j}(\bar{x})\right) + a_j \right\}\\
=&\sum_{i\in\bar{I}} \bar{z}^*_i \mathcal{T}^{C_i}_{\Omega_i,f_i}(\bar{x}) - \sum_{r\in\bar{R}} \bar{\lambda}_r \left[h^*_r\left(\frac{\bar{z}^*_r}{\bar{\lambda}_r}\right) - a_r \right] = \sum_{r\in\bar{R}} \bar{\lambda}_r \left[h_r\left(\mathcal{T}^{C_r}_{\Omega_r,f_r}(\bar{x})\right) + a_r \right], && (i)\\
&\bar{\lambda}_r h^*_r\left(\frac{\bar{z}^*_r}{\bar{\lambda}_r}\right) + \bar{\lambda}_r h_r\left(\mathcal{T}^{C_r}_{\Omega_r,f_r}(\bar{x})\right) = \bar{z}^*_r \mathcal{T}^{C_r}_{\Omega_r,f_r}(\bar{x}), \quad \forall r \in \bar{R}, && (ii)\\
&\bar{z}^*_i \mathcal{T}^{C_i}_{\Omega_i,f_i}(\bar{x}) + \bar{z}^*_i f^*_i\left(\frac{1}{\bar{z}^*_i}\bar{w}^*_i\right) + \sigma_{\Omega_i}(\bar{w}^*_i) = \langle \bar{w}^*_i, \bar{x}\rangle, \quad \forall i \in \bar{I}, && (iii)\\
&\sum_{i\in\bar{I}} \langle \bar{w}^*_i, \bar{x}\rangle = -\sigma_S\left(-\sum_{i\in\bar{I}} \bar{w}^*_i\right), && (iv)\\
&\max_{1\le j\le n} \left\{ h_j\left(\mathcal{T}^{C_j}_{\Omega_j,f_j}(\bar{x})\right) + a_j \right\} = h_r\left(\mathcal{T}^{C_r}_{\Omega_r,f_r}(\bar{x})\right) + a_r, \quad \forall r \in \bar{R}, && (v)\\
&\sum_{r\in\bar{R}} \bar{\lambda}_r = 1, \bar{\lambda}_k > 0, \quad k \in \bar{R}, \quad \bar{\lambda}_l = 0, \quad l \notin \bar{R}, \quad \bar{z}^*_i > 0, \quad i \in \bar{I}, \quad \bar{z}^*_j = 0, \quad j \notin \bar{I}, && (vi)\\
&\gamma_{C^0_i}(\bar{w}^*_i) = \bar{z}^*_i, \bar{w}^*_i \in X \setminus \{0_X\}, \quad i \in \bar{I}. && (vii)
\end{aligned}$$

2. Conversely, if there exists $\bar{x} \in S$ such that for some $(\bar{\lambda}, \bar{z}^*, \bar{w}^*) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times (X)^{|\bar{I}|}$ with the corresponding index sets $I \subseteq R \subseteq \{1, \dots, n\}$ the conditions (i)-(vii) are fulfilled, then $\bar{x}$ is an optimal solution to $(P^S_{h,\mathcal{T}})$, $(\bar{\lambda}, \bar{z}^*, \bar{w}^*) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times (X)^{|\bar{I}|}$ is an optimal solution to $(D^S_{h,\mathcal{T}})$ and $v(P^S_{h,\mathcal{T}})= v(D^S_{h,\mathcal{T}})$.

2.4 Special cases

In this section, we will focus on two specific instances of our general minimax location problem. We recall that in such problem, the objective function involves the so-called extended minimal time function is $\mathcal{T}^{C_i}_{\Omega_i,f_i}$, where, for $i = 1, \cdots, n$, C_i is closed and convex with $0_X \in \operatorname{int} C_i$, Ω_i is nonempty convex and compact and f_i is a proper, convex and lower semicontinuous extended real-valued

function.
Before introducing our methods, algorithms, and subsequent numerical experiments, we will first formulate the optimization problem and outline the optimality conditions for these two particular cases.

2.4.1 Special case one

Take, in the primal formulation $(P^S_{h,\mathcal{T}})$, $S = X, h_i = . + \delta_{\mathbb{R}_+}, a_i = 0$ and $f_i = \gamma_{G_i}$, where G_i is a closed and convex set such that $0_X \in C_i \cap G_i$. Using Remark 16, it yields $f_i^* = \delta_{G_i^0}$.

The primal problem $(P^S_{h,\mathcal{T}})$ becomes in this specific case

$$\inf_{x\in S}\max_{1\leq i\leq n}\left\{\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(x)\right\}, \tag{$P_{\gamma_G,\mathcal{T}}$}$$

while its dual formulation rewrites

$$\sup_{\substack{z_i^*\geq 0, i=1,\cdots,n,\\ I=\left\{i\in\{1,\cdots,n\}: z_i^*>0\right\}\\ w_i^*\in X, \gamma_{C_i^0}(w_i^*)\leq z_i^*, i\in I,\\ \gamma_{G_i^0}(w_i^*)\leq z_i^*, i\in I,\\ \sum_{i\in I} z_i^*\leq 1, \sum_{i\in I} w_i^*=0_X}}\left\{-\sum_{i\in I}\sigma_{\Omega_i}(w_i^*)\right\}. \tag{$D_{\gamma_G,\mathcal{T}}$}$$

The following result reformulates the optimality conditions (i)-(vii) in the context of Special case one 2.4.1.

Theorem 2.4.1. *1. Suppose that the assumptions of section 2.2 are fulfilled and let $\bar{x} \in S$ be an optimal solution to the problem $(P_{\gamma_G,\mathcal{T}})$. Then there exists an optimal solution $(\bar{z}^*, \bar{w}^*) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times X^{|\bar{I}|}$to the dual problem $(D_{\gamma_G,\mathcal{T}})$ with the corresponding sets $\bar{I} \subseteq \{1, \cdots, n\}$ such that*

$$\max_{1\leq j\leq n}\left\{\mathcal{T}^{C_j}_{\Omega_j,\gamma_{G_j}}(\bar{x})\right\} = \sum_{i\in\bar{I}}\bar{z}_i^*\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(\bar{x}), \tag{i}$$

$$\bar{z}_i^*\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(\bar{x}) + \sigma_{\Omega_i}(\bar{w}_i^*) = \langle\bar{w}_i^*, \bar{x}\rangle, \quad \forall i\in\bar{I}, \tag{ii}$$

$$\sum_{i\in\bar{I}}\bar{w}_i^* = 0_X, \tag{iii}$$

$$\max_{1\leq j\leq n}\left\{\mathcal{T}^{C_j}_{\Omega_j,\gamma_{G_j}}(\bar{x})\right\} = \mathcal{T}^{C_r}_{\Omega_i,\gamma_{G_i}}(\bar{x}), \quad \forall i\in\bar{I}, \tag{iv}$$

$$\sum_{i\in\bar{I}}\bar{z}_i^* = 1, \quad \bar{z}_i^* > 0, \quad i\in\bar{I}, \quad \bar{z}_j^* = 0, \quad j\notin\bar{I}, \tag{v}$$

$$\gamma_{C_i^0}(\bar{w}_i^*) = \bar{z}_i^*, \bar{w}_i^* \in X\setminus\{0_X\}, \gamma_{G_i^0}(\bar{w}_i^*) \leq \gamma_{C_i^0}(\bar{w}_i^*), \quad i\in\bar{I}. \tag{vi}$$

2. Conversely there exists $\bar{x} \in S$ such that for some $(\bar{z}^*, \bar{w}^*) \in \mathbb{R}^n_+ \times X^{|\bar{I}|}$ with the corresponding index sets $I \subseteq R \subseteq \{1, \ldots, n\}$ the conditions (i)-(vi) are fulfilled, then $\bar{x}$ is an optimal solution to $(P_{\gamma_G,\mathcal{T}})$, $(\bar{z}^*, \bar{w}^*) \in \mathbb{R}^n_+ \times X^{|\bar{I}|}$ is an optimal solution to $(D_{\gamma_G,\mathcal{T}})$ and $v(D_{\gamma_G,\mathcal{T}})$= $v(D_{\gamma_G,\mathcal{T}})$.

2.4.2 Special case two

We set on this instance $S = X, h_i = . + \delta_{\mathbb{R}_+}, a_i = 0$ and $f_i = \delta_{L_i}$, where L_i is a nonempty closed and convex set. It is clear that $f_i^* = \sigma_{L_i}$.

The primal problem $(P^S_{h,\mathcal{T}})$ rewrites

$$\inf_{x\in S} \max_{1\le i\le n} \left\{ \mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}(x) \right\} \tag{$P_\mathcal{T}$}$$

while its dual counterpart $(D^S_{h,\mathcal{T}})$ is given by

$$\sup_{\substack{z_i^*\ge 0, i=1,\cdots,n,\\ I=\left\{i\in\{1,\cdots,n\}: z_i^*>0\right\}\\ w_i^*\in X, \gamma_{C_i^0}(w_i^*)\le z_i^*, i\in I,\\ \sum_{i\in I} z_i^*\le 1, \sum_{i\in I} w_i^* = 0_X}} \left\{ -\sum_{i\in I} \left[\sigma_{L_i}(w_i^*) + \sigma_{\Omega_i}(w_i^*)\right] \right\}. \tag{$D_\mathcal{T}$}$$

The following result, which is analogous to Theorem 2.4.1, states the optimality conditions in this second case.

Theorem 2.4.2. *1. Suppose that the assumptions of section 2.2 are fulfilled and let $\bar{x} \in S$ be an optimal solution to the problem ($P_\mathcal{T}$). Then there exists an optimal solution $\bar{w}^* \in X^n$ to the dual problem ($D_\mathcal{T}$) with the corresponding sets $\bar{I} \subseteq \{1,\cdots,n\}$ such that*

$$\max_{1\le j\le n} \left\{ \mathcal{T}^{C_j}_{\Omega_j,\delta_{L_j}}(\bar{x}) \right\} = \sum_{i\in \bar{I}} \gamma_{C_i^0}(w_i^*) \mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}(\bar{x}), \tag{i}$$

$$\gamma_{C_i^0}(w_i^*) \mathcal{T}^{C_i}_{\Omega_i,+\delta_{L_i}}(w_i^*) + \sigma_{\Omega_i}(\bar{w}_i^*) = \langle \bar{w}_i^*, \bar{x}\rangle, \quad \forall i\in \bar{I}, \tag{ii}$$

$$\sum_{i\in\bar{I}} \bar{w}_i^* = 0_X, \tag{iii}$$

$$\max_{1\le j\le n} \left\{ \mathcal{T}^{C_j}_{\Omega_j,\delta_{L_j}}(\bar{x}) \right\} = \mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}(\bar{x}), \quad \forall i\in\bar{I}, \tag{iv}$$

$$\sum_{i\in\bar{I}} \gamma_{C_i^0}(w_i^*) = 1, \tag{v}$$

$$\bar{w}_i^* \in X\setminus\{0_X\}, \quad i\in\bar{I}, \quad \bar{w}_i^* = 0_X, \quad i\notin\bar{I}. \tag{vi}$$

2. Conversely there exists $\bar{x}\in S$ such that for some $(\bar{w}^*)\in X^n$ with the corresponding index sets $I\subseteq\{1,\ldots,n\}$ the conditions (i)-(vi) are fulfilled, then $\bar{x}$ is an optimal solution to ($P_\mathcal{T}$), $(\bar{w}^*)\in X^n$ is an optimal solution to ($D_\mathcal{T}$) and $v(P_\mathcal{T})$= $v(D_\mathcal{T})$.

Our intention in illustrating the same two cases as given in [11] is to compare the efficiency of their proposed method against two other algorithms.

2.5 Algorithms

In this section, we apply two existing algorithms in the context of our location problem, the Chambolle-Pock and the mirror descent algorithms. To compare our results with those obtained using the parallel splitting introduced in our location context in [11], we will consider the two instances provided in section 2.4.

We begin our proceedings with a brief overview of the method employed in [11]. The authors implemented a parallel version of the Douglas-Rachford splitting algorithm. This method is designed to decompose the (complex) optimization problem into simpler subproblems that can be solved independently and in parallel, improving computational efficiency. For more details about these splitting algorithms, we refer to the existing literature.
To do so, they expressed both the primal $P^S_{h,\mathcal{T}}$ and dual $D^S_{h,\mathcal{T}}$ problems as unconstrained optimization problems while setting $X = \mathbb{R}^d$.

1. Regarding Special case one 2.4.1, the primal $(P_{\gamma_G,\mathcal{T}})$ and dual $(D_{\gamma_G,\mathcal{T}})$ problems rewrite respectively

$$\min_{\substack{t>0, x\in\mathbb{R}^d, z\in(\mathbb{R}^d)^n,\\ \alpha\in(\mathbb{R}_+)^n, \beta\in(\mathbb{R}_+)^n}} \left\{ t + \sum_{i=1}^n \delta_{\mathrm{epi}\gamma_{C_i}}(x - p_i - z_i, \alpha_i) + \sum_{i=1}^n \delta_{\mathrm{epi}\gamma_{G_i}}(z_i, \beta_i) + \delta_H(\alpha, \beta, t) \right\}, \quad (P_{\gamma_G,\mathcal{T}})$$

where $H = \left\{(\alpha, \beta, t)^T : \alpha_i + \beta_i = t, \quad i = 1, \cdots, n\right\}$ and

$$-\min_{\substack{w^*\in(\mathbb{R}^d)^n,\\ z^*\in(\mathbb{R}_+)^n}} \left\{ \sum_{i=1}^n p_i^T w_i^* + \sum_{i=1}^n \delta_{\mathrm{epi}\gamma_{C_i^0}}(w_i^*, z_i^*) + \sum_{i=1}^n \delta_{\mathrm{epi}\gamma_{G_i^0}}(w_i^*, z_i^*) + \delta_D(z^*) + \delta_E(w^*) \right\}, \quad (D_{\gamma_G,\mathcal{T}})$$

where $D = \{z^* \in (\mathbb{R}_+)^n : \sum_{i=1}^n z_i^* \leq 1\}$ and $E = \left\{w^* \in (\mathbb{R}^d)^n : \sum_{i=1}^n w_i^* = 0_{\mathbb{R}^d}\right\}$.

2. Regarding Special case two 2.4.2, the primal $(P_\mathcal{T})$ and dual $(D_\mathcal{T})$ problems become

$$\min_{\substack{t>0, x\in\mathbb{R}^d,\\ y\in(\mathbb{R}^d)^n, z\in(\mathbb{R}^d)^n}} \left\{ t + \sum_{i=1}^n \delta_{\mathrm{epi}\gamma_{C_i}}(x - y_i - z_i, t) + \sum_{i=1}^n \delta_{\Omega_i}(y_i) + \sum_{i=1}^n \delta_{L_i}(z_i) \right\}, \quad (P_\mathcal{T})$$

and

$$-\min_{w^*\in(\mathbb{R}^d)^n,} \left\{ \sum_{i=1}^n \sigma_{\Omega_i}(w_i^*) + \sum_{i=1}^n \sigma_{L_i}(w_i^*) + \delta_F(w^*) + \delta_E(w^*) \right\}, \quad (D_\mathcal{T})$$

where $E = \left\{w^* \in (\mathbb{R}^d)^n : \sum_{i=1}^n w_i^* = 0_{\mathbb{R}^d}\right\}$ and $F = \left\{w^* \in (\mathbb{R}_+)^n : \sum_{i=1}^n \gamma_{C_i^0}(w_i^*) \leq 1\right\}$.

The parallel splitting proximal point method used to solve the minimax location problem relies heavily on the proximal operator (1.9) and is based on Theorem 3.6.2.

We can now present and detail our investigations. We choose to adapt two existings numerical methods in the context of our location problem. The first one is the Chambolle-Pock algorithm which was introduced by Antonin Chambolle and Thomas Pock [9] and the second one is the mirror descent that was originally proposed by Nemirovsky and Yudin in 1983.

2.5.1 Chambolle-Pock algorithm

The Chambolle-Pock algorithm is a primal-dual method designed for solving optimization problems where the objective function can be decomposed into a sum of two convex functions, one of which involves a linear operator.
We shall first remind briefly the context of application of the Chambolle-Pock algorithm.
Let X and Y be two Hilbert spaces and $K : X \to Y$ a continuous linear operator. Let $F : Y \to \bar{\mathbb{R}}$ and $G : X \to \bar{\mathbb{R}}$ be two convex (not necessarily continuous or differentiable) functions such that

their respective proximity operators are inexpensive.
Consider the problem

$$\min_{x\in X} F(Kx)+G(x) \tag{P}$$

which we will refer to as the primal formulation. The corresponding dual formulation is given by

$$\max_{y\in Y} -\left(F^*(y)+G^*(-K^*y)\right). \tag{D}$$

Using conjugate theory, we write the so-called primal-dual formulation

$$\min_{x\in X}\max_{y\in Y} -F^*(y)+\langle Kx,y\rangle+G(x). \tag{PD}$$

Let us call the objective function of (PD) the min-max function. We recall that $(\bar{x},\bar{y})$ is a saddle point of the min-max function if $\bar{x}$ solves the primal problem (P) and $\bar{y}$ solves the dual problem (D).
A saddle point $(\bar{x},\bar{y})\in X\times Y$ of this min-max function should satisfy the optimality conditions

$$\begin{cases} 0_Y \in K\bar{x}-\partial F^*(\bar{y}),\\ 0_X\in K^*\bar{y}+\partial G(\bar{x}).\end{cases} \tag{2.4}$$

The Chambolle-Plock algorithm alternates between updates of the primal and dual variables. It relies on the using of the proximal operator to handle non-smooth terms. In [9], the authors proposed Algorithm 5 as well as the following result.

Theorem 2.5.1 (Convergence)**.** *Let $L=\|K\|$ ans suppose that (PD) has a saddle point. Choose $\theta=1,\tau\sigma L^2<1$ and let $(x^k,\bar{x}^k,y^k)_{k\in\mathbb{N}}$ defined as in Algorithm 5.*
Then if the dimensions of the spaces X and Y are finite then there exists a saddle point (x^*,y^*) such that $x^k\to x^*$ and $y^k\to y^*$.

Remark 22. It is worth noting that we chose to present in Theorem 2.5.1 a result concerning finite-dimensional spaces, as this is directly relevant to our numerical experiments. However, the original theorem by Chambolle and Pock is stated for possibly infinite-dimensional spaces and involves the concept of a partial primal-dual gap. For a more detailed explanation and the full theorem, we refer [9].

Remark 23. In principle, one could make other choices for θ under the condition that $\theta\in[0,1]$. However, the authors observed that getting estimations of the convergence was more straightforward while imposing that $\theta=1$.

In order to apply the Chambolle-Pock method to our problem, we first reformulate the primal problem in both special cases. For Special case one 2.4.1, we rewrite $(P_{\gamma_{G,\mathcal{T}}})$ as

$$\inf_{x\in X}\max_{1\le i\le n}\left\{\mathcal{T}^{C_i}_{\Omega_i,\gamma G_i}(x)\right\} \iff \inf_{(x,t)\in X\times\mathbb{R}}\left\{t+\sum_{i=1}^n \delta_{\mathrm{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(x,t)\right\} \tag{2.5}$$

and for Special case two 2.4.2, $(P_{\mathcal{T}})$ rewrites

$$\inf_{x\in X}\max_{1\le i\le n}\left\{\mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}(x)\right\} \iff \inf_{(x,t)\in X\times\mathbb{R}}\left\{t+\sum_{i=1}^n \delta_{\mathrm{epi}\mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}}(x,t)\right\}. \tag{2.6}$$

Then, we compute the Fenchel dual of both right hand side formulations of (2.5)-(2.6) which are obtained via perturbation theory. We will detail our computations for the analysis of the first case, noting that similar steps apply for the second case.

Proposition 2.5.2. *Under the assumptions of section 2.2, the Fenchel dual problem associated to $(P_{\gamma_G,\mathcal{T}})$ is given by*

$$\sup_{\substack{x_i^*\in X,\lambda_i^*\in\mathbb{R},\\ \gamma_{C_i^0}(x_i^*)\le\lambda_i,\gamma_{G_i^0}(x_i^*)\le\lambda_i,\\ i=1,\cdots,n,\\ \sum_{i=1}^n x_i^*=0_X,\sum_{i=1}^n\lambda_i^*=1}} \left\{-\sum_{i=1}^n \sigma_{\Omega_i}(x_i^*)\right\}. \qquad (D^F_{\gamma_G,\mathcal{T}})$$

Proof. Let us first introduce the perturbation function in order to formulate the Fenchel conjugate dual associate to $(P_{\gamma_G,\mathcal{T}})$.
For $((x,t),(y,s))\in(X\times\mathbb{R})^2$, define $\Phi:(X\times\mathbb{R})^2\to\bar{\mathbb{R}}$ by

$$\Phi((x,t),(y,s)) := t+s+\sum_{i=1}^n \delta_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(x+y,t+s)$$

Let us first compute its conjugate.

$$\Phi^*((x^*,t^*),(y^*,s^*)) = \sup_{\substack{x\in X,t\in\mathbb{R}\\ y\in X,s\in\mathbb{R}}}\Big\{\langle((x^*,t^*),(y^*,s^*)),((x,t),(y,s))\rangle-\Phi((x,t),(y,s))\Big\}$$

$$= \sup_{\substack{x\in X,t\in\mathbb{R}\\ y\in X,s\in\mathbb{R}}}\left\{\langle((x^*,t^*),(y^*,s^*)),((x,t),(y,s))\rangle-\left(t+s+\sum_{i=1}^n \delta_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(x+y,t+s)\right)\right\}$$

$$= \sup_{x\in X}\left\{\sum_{i=1}^n\langle x_i^*,x\rangle+\sup_{t\in\mathbb{R}}\left[\sum_{i=1}^n\langle t_i^*,t\rangle+\sup_{y\in X}\left\{\sum_{i=1}^n\langle y_i^*,y\rangle\right.\right.\right.$$

$$\left.\left.\left.+\sup_{s\in\mathbb{R}}\left(\sum_{i=1}^n\langle s_i^*,s\rangle-\left(t+s+\sum_{i=1}^n\delta_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(x+y,t+s)\right)\right)\right\}\right]\right\}$$

$$= \sup_{x\in X}\left\{\sum_{i=1}^n\langle x_i^*,x\rangle+\sup_{t\in\mathbb{R}}\left[\sum_{i=1}^n\langle t_i^*,t\rangle+\sup_{y\in X}\left\{\sum_{i=1}^n\langle y_i^*,z-x\rangle\right.\right.\right.$$

$$\left.\left.\left.+\sup_{s\in\mathbb{R}}\left(\sum_{i=1}^n\langle s_i^*,\tau-t\rangle-\left(\tau+\sum_{i=1}^n\delta_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(z,\tau)\right)\right)\right\}\right]\right\}$$

$$= \sup_{x\in X}\left\{\sum_{i=1}^n\langle x_i^*-y_i^*,x\rangle+\sup_{t\in\mathbb{R}}\left[\sum_{i=1}^n\langle t_i^*-s_i^*,t\rangle+\sup_{z\in X}\left\{\sum_{i=1}^n\langle y_i^*,z\rangle\right.\right.\right.$$

$$\left.\left.\left.+\sup_{\tau\in\mathbb{R}}\left(\sum_{i=1}^n\langle s_i^*-\frac{1}{n},\tau\rangle-\left(\sum_{i=1}^n\delta_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(z,\tau)\right)\right)\right\}\right]\right\}$$

$$= \sum_{i=1}^n\sup_{x\in X}\langle(x_i^*-y_i^*),x\rangle+\sum_{i=1}^n\sup_{t\in\mathbb{R}}\langle(t_i^*-s_i^*),t\rangle+$$

$$\sup_{z\in X}\left\{\sum_{i=1}^n\langle y_i^*,z\rangle+\sup_{\tau\in\mathbb{R}}\left(\sum_{i=1}^n\langle s_i^*-\frac{1}{n},\tau\rangle-\left(\sum_{i=1}^n\delta_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(z,\tau)\right)\right)\right\}$$

$$= \sup_{x\in X}\langle\sum_{i=1}^n(x_i^*-y_i^*),x\rangle+\sup_{t\in\mathbb{R}}\langle\sum_{i=1}^n\left(t_i^*-s_i^*-\frac{1}{n}\right),t\rangle$$

$$+\sup_{z\in X}\left\{\sum_{i=1}^{n}\langle y_i^*,z\rangle+\sup_{\tau\in\mathbb{R}}\left(\sum_{i=1}^{n}\langle s_i^*,\tau\rangle-\left(\sum_{i=1}^{n}\delta_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(z,\tau)\right)\right)\right\},$$

where the last equality was obtained after a change of variable involving the variable s^*. In particular, it yields that

$$\begin{aligned}\Phi^*((0_X,0),(y^*,s^*)) &= \sup_{x\in X}\langle -\sum_{i=1}^{n} y_i^*,x\rangle+\sup_{t\in\mathbb{R}}\langle -\sum_{i=1}^{n}s_i^*-1,t\rangle\\ &\quad+\sum_{i=1}^{n}\sup_{\substack{z\in X,\\ \tau\in\mathbb{R}}}\left\{\langle y_i^*,z\rangle+\langle s_i^*,\tau\rangle-\delta_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(z,\tau)\right\}.\end{aligned}\tag{2.7}$$

From the first two terms of the right hand side of (2.7), we deduce that

$$\begin{aligned}\sum_{i=1}^{n} y_i^* &= 0_X,\\ \sum_{i=1}^{n} s_i^* &= -1.\end{aligned}\tag{2.8}$$

Moreover, regarding the third term of the same equality (2.7), we have

$$\sum_{i=1}^{n}\sup_{\substack{z\in X,\\ \tau\in\mathbb{R}}}\left\{\langle y_i^*,z\rangle+\langle s_i^*,\tau\rangle-\delta_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(z,\tau)\right\}=\sum_{i=1}^{n}\delta^*_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(y_i^*,s_i^*)=\sum_{i=1}^{n}\sigma_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(y_i^*,s_i^*).\tag{2.9}$$

By virtue of the perturbation theory and notably due to Young-Fenchel inequality, the dual problem of $(P_{\gamma_G,\mathcal{T}})$ is given by

$$\sup_{x_i^*\in X,t_i^*\in\mathbb{R},i=1,\cdots,n} -\Phi^*((0_X,0),(x^*,t^*)),$$

which, thanks to (2.7)-(2.9), leads us to

$$\sup_{\substack{x_i^*\in X,t_i^*\in\mathbb{R},\\ i=1,\cdots,n,\\ \sum_{i=1}^n x_i^*=0_X,\sum_{i=1}^n t_i^*=-1}}\left\{-\sum_{i=1}^{n}\sigma_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(x_i^*,t_i^*)\right\}.$$

Finally, using the following result given in [11, Remark 3.8]

$$\sigma_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(x_i^*,t_i^*)=\min_{\substack{\lambda_i>0,t_i^*=-\lambda_i,\\ \gamma_{C_i^0}(x_i^*)\le\lambda_i,\gamma_{G_i^0}(x_i^*)\le\lambda_i}}-\sigma_{\Omega_i}(x_i^*),$$

we deduce that the dual problem $(D_{\mathcal{T}})$ reformulates

$$\sup_{\substack{x_i^*\in X,\lambda_i^*\in\mathbb{R},\\ \gamma_{C_i^0}(x_i^*)\le\lambda_i,\gamma_{G_i^0}(x_i^*)\le\lambda_i,\\ i=1,\cdots,n,\\ \sum_{i=1}^n x_i^*=0_X,\sum_{i=1}^n\lambda_i^*=1}}\left\{-\sum_{i=1}^{n}\sigma_{\Omega_i}(x_i^*)\right\}.$$

□

A similar result is provided when addressing Special case two 2.4.2.

Proposition 2.5.3. *Under the assumptions of section 2.2, the Fenchel dual problem associated to $(P_{\mathcal{T}})$ is given by*

$$\sup_{\substack{x_i^* \in X, \lambda_i^* \in \mathbb{R}, \\ \gamma_{C_i^0}(x_i^*) \leq \lambda_i, \\ i=1,\cdots,n, \\ \sum_{i=1}^n x_i^* = 0_X, \sum_{i=1}^n \lambda_i^* = 1}} \left\{ -\sum_{i=1}^n \left[\sigma_{\Omega_i}(x_i^*) + \sigma_{L_i}(x_i^*) \right] \right\}. \tag{$D_{\mathcal{T}}^F$}$$

Let us now check the assumptions and the corresponding framework of the Chambolle-Pock algorithm. As previously, we will provide a detailed analysis for Special case one 2.4.1 and the methodology for Special case one 2.4.1 will follow analogous steps.
We adopte the notation $((x,t)_{i=1}^n) = ((x,t), \cdots, (x,t))$ and we set, for all $x \in X$, $x_i \in X, i = 1, \cdots, n$ and for all $t \in \mathbb{R}$, $t_i \in \mathbb{R}, i = 1, \cdots, n$,

$$K(x,t) = ((x,t)_{i=1}^n), \quad F((x_i,t_i)_{i=1}^n) = \sum_{i=1}^n \delta_{\text{epi} T_{\Omega_i, \gamma_{G_i}}^{C_i}}(x_i,t_i), \quad G((x,t)) = t.$$

Introducing corresponding dual variables $x^* \in X$, $x_i^* \in X, i = 1, \cdots, n$ and $t^* \in \mathbb{R}$, $t_i^* \in \mathbb{R}, i = 1, \cdots, n$, we can easily make the following observations

$$\begin{aligned}
&K^*\left((x_i^*, t_i^*)_{i=1}^n\right) = (\sum_{i=1}^n x_i^*, \sum_{i=1}^n t_i^*), \\
&G^*((x^*,t^*)) = \sigma_{X \times \mathbb{R}}\left(x^*, t^* - 1\right), \\
&G^*(-K^*\left((x_i^*, t_i^*)_{i=1}^n\right)) = \sigma_{X \times \mathbb{R}}\left(-\sum_{i=1}^n x_i^*, -\sum_{i=1}^n t_i^* - 1\right), \\
&F(K(x,t)) = \sum_{i=1}^n \delta_{\text{epi} T_{\Omega_i, \gamma_{G_i}}^{C_i}}(x,t), \\
&F^*((x_i^*, t_i^*)_{i=1}^n) = \sum_{i=1}^n \sigma_{\text{epi} T_{\Omega_i, \gamma_{G_i}}^{C_i}}(x_i^*, t_i^*).
\end{aligned}$$

The optimality conditions (2.4) rewrites

$$\left\{ \begin{array}{l} 0_X \in ((x,t)_{i=1}^n) - \partial\left(\sum_{i=1}^n \sigma_{\text{epi} T_{\Omega_i, \gamma_{G_i}}^{C_i}}\right)(x_i^*, t_i^*), \\ 0 \in (\sum_{i=1}^n x_i^*, \sum_{i=1}^n t_i^*) + \partial(t), \end{array} \right. ,$$

which yields by the subdifferential rule given in Theorem 1.1.22

$$\left\{ \begin{array}{l} ((x,t)_{i=1}^n) \in \sum_{i=1}^n \partial \sigma_{\text{epi} T_{\Omega_i, \gamma_{G_i}}^{C_i}}(x_i^*, t_i^*), \\ (\sum_{i=1}^n x_i^*, \sum_{i=1}^n t_i^*) = (0_X, -1). \end{array} \right.$$

Given the initial data $((x,t)^0, (y,\lambda)^0) \in (X \times \mathbb{R}) \times ((X)^n \times \mathbb{R}^n)$, and constants $\sigma, \tau > 0$, $\theta \in [0,1]$, setting $(\bar{x}, \bar{t})^0 = (x,t)^0$, Algorithm 5 rewrites in our case

$$\left\{ \begin{array}{l} (y^{k+1}, \lambda^{k+1}) = \text{prox}_{\sigma \sum_{i=1}^n \sigma_{\text{epi} T_{\Omega_i, \gamma_{G_i}}^{C_i}}}\left((y^k, \lambda^k) + \sigma K(\bar{x}^k, \bar{t}^k)\right), \\ (x^{k+1}, t^{k+1}) = \text{prox}_{\tau G}\left((x^k, t^k) - \tau K^*(y^{k+1}, \lambda^{k+1})\right), \\ (\bar{x}^{k+1}, \bar{t}^{k+1}) = (x^{k+1}, t^{k+1}) + \theta\left((x^{k+1}, t^{k+1}) - (x^k, t^k)\right). \end{array} \right. \tag{2.10}$$

Using Proposition 1.1.28, identity 1.10 and the extended Moreau decomposition formula 1.13 we can deduce that

$$\begin{aligned} &\operatorname{prox}_{\sigma \sum_{i=1}^{n} \sigma_{\operatorname{epi} T^{C_i}_{\Omega_i,\gamma_{G_i}}}}\Big((y^k,\lambda^k)+\sigma K(\bar{x}^k,\bar{t}^k)\Big) \\ &= \Big(\operatorname{prox}_{\sigma \operatorname{epi} T^{C_i}_{\Omega_i,\gamma_{G_i}}}\Big((y_i^k,\lambda_i^k)+\sigma(\bar{x}^k,\bar{t}^k)\Big)\Big)_{i=1}^{n} \\ &= \Big(\Big(I-\sigma \operatorname{prox}_{\sigma^{-1}\operatorname{epi} T^{C_i}_{\Omega_i,\gamma_{G_i}}}\Big)\Big(\sigma^{-1}(y_i^k,\lambda_i^k)+(\bar{x}^k,\bar{t}^k)\Big)\Big)_{i=1}^{n} \\ &= \Big(\Big(I-\sigma \mathrm{P}_{\operatorname{epi} T^{C_i}_{\Omega_i,\gamma_{G_i}}}\Big)\Big(\sigma^{-1}(y_i^k,\lambda_i^k)+(\bar{x}^k,\bar{t}^k)\Big)\Big)_{i=1}^{n}. \end{aligned} \tag{2.11}$$

Then, (2.10) reformulates

$$\begin{cases} (y_i^{k+1},\lambda_i^{k+1}) = (y_i^k,\lambda_i^k)+\sigma(\bar{x}^k,\bar{t}^k)-\sigma \mathrm{P}_{\operatorname{epi} T^{C_i}_{\Omega_i,\gamma_{G_i}}}\left(\sigma^{-1}(y_i^k,\lambda_i^k)+(\bar{x}^k,\bar{t}^k)\right), \quad i=1,\cdots,n, \\ (x^{k+1},t^{k+1}) = (x^k,t^k)-\tau\Big(\sum_{i=1}^{n} y_i^{k+1}, \sum_{i=1}^{n}\lambda_i^{k+1}+1\Big), \\ (\bar{x}^{k+1},\bar{t}^{k+1}) = (x^{k+1},t^{k+1})+\theta\left((x^{k+1},t^{k+1})-(x^k,t^k)\right). \end{cases} \tag{2.12}$$

Finally, we can now present the algorithm when applied for our Special case one 2.4.1 and Special case two 2.4.2.

Algorithm 1 Chambolle-Pock Algorithm - Special Case One

```
1: Input: Choose and constants $\sigma,\tau > 0$, $\theta \in [0,1]$. Set $(\bar{x},\bar{t})^0 = (x,t)^0$.
2: For $k \geq 0$ do
3: for $i = 1,\cdots,n$ do
4:     $y_i^{k+1} = y_i^k + \sigma\bar{x}^k - \sigma \mathrm{P}_i \mathrm{P}_{\operatorname{epi} T^{C_i}_{\Omega_i,\gamma_{G_i}}}\left(\sigma^{-1}(y_i^k,\lambda_i^k)+(\bar{x}^k,\bar{t}^k)\right)$
5: end for
6: for $i = 1,\cdots,n$ do
7:     $\lambda_i^{k+1} = \lambda_i^k + \sigma\bar{t}^k - \sigma \mathrm{P}_{n+i} \mathrm{P}_{\operatorname{epi} T^{C_i}_{\Omega_i,\gamma_{G_i}}}\left(\sigma^{-1}(y_i^k,\lambda_i^k)+(\bar{x}^k,\bar{t}^k)\right)$
8: end for
9: $x^{k+1} = x^k - \tau\sum_{i=1}^{n} y_i^{k+1}$
10: $t^{k+1} = t^k - \tau\left(\sum_{i=1}^{n}\lambda_i^{k+1}+1\right)$
11: $\bar{x}^{k+1} = x^{k+1} + \theta\left(x^{k+1}-x^k\right)$
12: $\bar{t}^{k+1} = t^{k+1} + \theta\left(t^{k+1}-t^k\right)$
13: End for
```

where P_i is the projection onto the i-th component.

Similarly, regarding the second case, we have the following.

We would like to explicit the formula of the projection onto the epigraph of the two perturbed minimal time functions $T^{C_i}_{\Omega_i,\gamma_{G_i}}$ and $T^{C_i}_{\Omega_i,\delta_{L_i}}, i=1,\cdots,n$.

Inspired by [14] where the authors stated and proved the result with respect to the gauge function γ_C, we give the following theorem, considering this time the extended perturbed minimal time function $\mathcal{T}^{C}_{\Omega,\gamma_G} = \gamma_C \;\Box\; \delta_\Omega \;\Box\; \gamma_G$.

Algorithm 2 Chambolle-Pock - Special Case Two

```
1: Input: Choose $((x,t)^0,(y,\lambda)^0) \in (X \times \mathbb{R}) \times (X^n \times \mathbb{R}^n)$, and constants $\sigma, \tau > 0$, $\theta \in [0,1]$. Set $(\bar{x},\bar{t})^0 = (x,t)^0$.
2: For $k \geq 0$ do
3: for $i = 1, \cdots, n$ do
4:     $y_i^{k+1} = y_i^k + \sigma\bar{x}^k - \sigma \mathrm{P}_i \mathrm{P}_{\operatorname{epi} T^{C_i}_{\Omega_i,\delta_{L_i}}}\left(\sigma^{-1}(y_i^k,\lambda_i^k) + (\bar{x}^k,\bar{t}^k)\right)$
5: end for
6: for $i = 1, \cdots, n$ do
7:     $\lambda_i^{k+1} = \lambda_i^k + \sigma\bar{t}^k - \sigma \mathrm{P}_{n+i} \mathrm{P}_{\operatorname{epi} T^{C_i}_{\Omega_i,\delta_{L_i}}}\left(\sigma^{-1}(y_i^k,\lambda_i^k) + (\bar{x}^k,\bar{t}^k)\right)$
8: end for
9: $x^{k+1} = x^k - \tau\sum_{i=1}^n y_i^{k+1}$
10: $t^{k+1} = t^k - \tau\left(\sum_{i=1}^n \lambda_i^{k+1} + 1\right)$
11: $\bar{x}^{k+1} = x^{k+1} + \theta\left(x^{k+1} - x^k\right)$
12: $\bar{t}^{k+1} = t^{k+1} + \theta\left(t^{k+1} - t^k\right)$
13: End for
```
where P_i is the projection onto the i-th component.

Theorem 2.5.4. *Let C be a closed convex subset of X such that $0_X \in C$. Then it holds for every $(x,\xi) \in X \times \mathbb{R}$ that*

$$\mathrm{P}_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}}(x,\xi) = \begin{cases} (x,\xi), & \text{if } \mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(x) \leq \xi, \\ \left(\mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right)}(x), \xi\right), & \text{if } x \notin \operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}} \\ & \text{and } \mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\left(\mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right)}(x)\right) \leq \xi < \mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(x), \\ (\bar{y},\bar{\theta}), & \text{otherwise}, \end{cases}$$

where

$$\bar{y} = x - \operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_\Omega\right)}\left(\bar{\lambda}^{-1}x\right) \quad \text{and} \quad \bar{\theta} = \bar{\lambda} + \xi,$$

and $\bar{\lambda} > 0$ is a solution of an equation of the form

$$\begin{aligned} \lambda + \xi =& \langle x, \operatorname{prox}_{\lambda^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}\left(\lambda^{-1}x\right)\rangle_X - \lambda\left\|\operatorname{prox}_{\lambda^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}\left(\lambda^{-1}x\right)\right\|_X^2 \\ &- \sigma_{\Omega_i}\left(\operatorname{prox}_{\lambda^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}\left(\lambda^{-1}x\right)\right). \end{aligned}$$

Proof. Let us consider for fixed $(x,\xi) \in X \times \mathbb{R}$ the following optimization problem

$$\min_{\substack{(y,\theta)\in X\times\mathbb{R}, \\ \mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(y)\leq\theta}} \left\{\frac{1}{2}(\theta-\xi)^2 + \frac{1}{2}\left\|y - x\right\|_X^2\right\}. \tag{2.13}$$

If $\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(x) \le \xi$ -in which case $(x,\xi) \in \operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}$-, then it is obvious that $(\bar{y},\bar{\theta}) = (x,\xi)$.
In the following, we consider the non-trivial case where $\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(x) > \xi$.
We define the functions $f : X \times \mathbb{R} \to \mathbb{R}$ and $g : X \times \mathbb{R} \to \mathbb{R}$ by

$$f(y,\theta) := \frac{1}{2}(\theta-\xi)^2 + \frac{1}{2}\left\|y-x\right\|^2_X, \quad g(y,\theta) = \mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(y) - \theta.$$

It is easy to see that f is continuous and strongly convex, and g is proper, lower semicontinuous, and convex by [13, Theorem 2.1]. As $0_X \in C_i \cap G_i$, we have $\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(0) < 1$, it follows by [8, Theorem 3.3.16] that

$$0 \in \partial(f + \bar{\lambda} g)(\bar{y},\bar{\theta}) \tag{2.14}$$

and

$$\begin{cases} \bar{\lambda} g(\bar{y},\bar{\theta}) = 0, \\ g(\bar{y},\bar{\theta}) \le 0, \end{cases}$$

or equivalently

$$\begin{cases} \bar{\lambda}(\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(\bar{y}) - \bar{\theta}) = 0, \\ \mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(\bar{y}) \le \bar{\theta}, \end{cases} \tag{2.15}$$

where $(\bar{y},\bar{\theta})$ is the unique solution of (2.13) and $\bar{\lambda} \ge 0$ is the associated Lagrange multiplier. Furthermore, from [8, Theorem 3.5.13] one gets that

$$0 \in \partial(f + \bar{\lambda} g)(\bar{y},\bar{\theta}) \iff 0 \in \partial f(\bar{y},\bar{\theta}) + \partial(\bar{\lambda} g)(\bar{y},\bar{\theta}). \tag{2.16}$$

1. If $\bar{\lambda} = 0$, then it follows by identity (1.10) and equivalence (1.11) that

$$\begin{aligned} 0 \in \partial f(\bar{y},\bar{\theta}) + \partial\delta_{\operatorname{dom} g}(\bar{y},\bar{\theta}) &\iff 0 \in (\bar{y}-x, \bar{\theta}-\xi) + \partial\delta_{\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}} \times \mathbb{R}}(\bar{y},\bar{\theta}) \\ &\iff 0 \in (\bar{y}-x, \bar{\theta}-\xi) + \partial\delta_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right) \times \mathbb{R}}(\bar{y},\bar{\theta}) \\ &\iff (x-\bar{y}, \xi-\bar{\theta}) \in \partial\delta_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right) \times \mathbb{R}}(\bar{y},\bar{\theta}) \\ &\iff (\bar{y},\bar{\theta}) = \mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right) \times \mathbb{R}}(x,\xi) \\ &\iff \bar{y} = \mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right)}(x) \text{ and } \bar{\theta} = \xi, \end{aligned}$$

   and thus, it holds by the feasibility condition (2.15) that $\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\left(\mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right)}(x)\right) \le \xi$, from which follows that $\mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right)}(x) \in \operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}$.

   (a) If $x \in \operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}$, then $\mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right)}(x) = x$ which would imply by the feasibility condition (2.15) that $\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(x) \le \xi$, which contradicts our assumption.

   (b) If $x \notin \operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}$ and the inequalities $\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\left(\mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right)}(x)\right) \le \xi < \mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}(x)$ hold, then $(\bar{y},\bar{\theta}) = \left(\mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right)}(x), \xi\right)$.

2. If $\bar{\lambda} > 0$, then we deduce from (2.16) and equivalence (1.11) that

$$\begin{aligned} 0 \in \partial(f + \bar{\lambda} g)(\bar{y}, \bar{\theta}) &\iff 0 \in \partial f(\bar{y}, \bar{\theta}) + \bar{\lambda}\partial g(\bar{y}, \bar{\theta}) \\ &\iff \nabla f(\bar{y}, \bar{\theta}) \in -\bar{\lambda}\partial g(\bar{y}, \bar{\theta}) \\ &\iff (\bar{y} - x, \bar{\theta} - \xi) \in -\bar{\lambda}\partial(\mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}(\bar{y}) - \bar{\theta}), \\ &\iff \left\{ \begin{array}{l} \bar{y} - x \in -\bar{\lambda}\partial(\mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}})(\bar{y}), \\ \bar{\theta} - \xi = \bar{\lambda}, \end{array} \right. \end{aligned}$$

so that

$$\left\{ \begin{array}{l} \bar{y} = \operatorname{prox}_{\bar{\lambda}\mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}}(x), \\ \bar{\theta} = \xi + \bar{\lambda}. \end{array} \right. \tag{2.17}$$

By combining (2.17) and (2.15) we derive that $\mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}(\bar{y}) = \xi + \bar{\lambda}$. Finally, by [13, Lemma 2.1 and Remark 2.2] it holds that $\left(\mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}\right)^* = \delta_{C_i^0} + \delta_{G_i^0} + \sigma_\Omega$. Then, by the extended Moreau decomposition formula (1.13), one has

$$\begin{aligned} \bar{y} = \operatorname{prox}_{\bar{\lambda}\mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}}(x) =& x - \bar{\lambda}\operatorname{prox}_{\bar{\lambda}^{-1}\left(\mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}\right)^*}\left(\bar{\lambda}^{-1}x\right) \\ =& x - \bar{\lambda}\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right) \\ =& x - \bar{\lambda}\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0 \cap G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right). \end{aligned} \tag{2.18}$$

Furthermore, it yields by identity (1.14) that

$$\begin{aligned} &\bar{\lambda} + \xi = \mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}(\bar{y}) \\ =& \mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}\left(\operatorname{prox}_{\bar{\lambda}\mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}}(x)\right) \pm \left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)\left(\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right) \\ =& \langle \operatorname{prox}_{\bar{\lambda}\mathcal{T}^{C_i}_{\Omega_i, \gamma_{G_i}}}(x), \operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\rangle_X \\ &- \left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)\left(\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right) \\ =& \langle x - \bar{\lambda}\left(\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right), \operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\rangle_X \\ &- \left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)\left(\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right) \\ =& \langle x, \operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\rangle_X - \bar{\lambda}\left\|\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right\|_X^2 \\ &- \left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)\left(\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0} + \delta_{G_i^0} + \sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right) \end{aligned} \tag{2.19}$$

$$
\begin{aligned}
&=\langle x, \operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\rangle_X - \bar{\lambda}\left\|\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right\|_X^2 \\
&\quad - \left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)\left(\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right) \\
&=\langle x, \operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\rangle_X - \bar{\lambda}\left\|\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right\|_X^2 \\
&\quad - \sigma_{\Omega_i}\left(\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right)\right),
\end{aligned}
\tag{2.20}
$$

where the last equality is obtained from [1] where the authors showed that under assumptions on the functions f and g, one has $\operatorname{prox}_{f+g} = \operatorname{prox}_f \circ \operatorname{prox}_g^f$.

□

Naturally, when considering the second minimal time function, we get the following result, which can be proven analogously to the previous one.

Theorem 2.5.5. *Let C be a closed convex subset of X such that $0_X \in C$. Then it holds for every $(x,\xi) \in X \times \mathbb{R}$ that*

$$
\mathrm{P}_{\operatorname{epi}\mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}}(x,\xi) = \begin{cases}
(x,\xi), & \text{if } \mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}(x) \leq \xi, \\
\left(\mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}\right)}(x), \xi\right), & \text{if} x \notin \operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}} \\
& \text{and } \mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}\left(\mathrm{P}_{\operatorname{cl}\left(\operatorname{dom}\mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}\right)}(x)\right) \leq \xi < \mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}(x), \\
(\bar{y},\bar{\theta}), & \text{otherwise},
\end{cases}
$$

where

$$
\bar{y} = x - \operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0}+\sigma_{L_i}+\sigma_{\Omega_i}\right)}\left(\bar{\lambda}^{-1}x\right) \quad \text{and} \quad \bar{\theta} = \bar{\lambda} + \xi,
$$

and $\bar{\lambda} > 0$ is a solution of an equation of the form

$$
\begin{aligned}
\lambda + \xi =& \langle x, \operatorname{prox}_{\lambda^{-1}\left(\delta_{C_i^0}+\sigma_{L_i}+\sigma_{\Omega_i}\right)}\left(\lambda^{-1}x\right)\rangle_X - \lambda\left\|\operatorname{prox}_{\lambda^{-1}\left(\delta_{C_i^0}+\sigma_{L_i}+\sigma_{\Omega_i}\right)}\left(\lambda^{-1}x\right)\right\|_X^2 \\
& - \left(\delta_{C_i^0}+\sigma_{L_i}+\sigma_{\Omega_i}\right)\left(\operatorname{prox}_{\lambda^{-1}\left(\delta_{C_i^0}+\sigma_{L_i}+\sigma_{\Omega_i}\right)}\left(\lambda^{-1}x\right)\right).
\end{aligned}
$$

Remark 24. It is important to note that in both cases, it is not trivial (if not impossible) to determine a closed form representation of $\operatorname{prox}_{\lambda\left(\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}\right)^*}$. Indeed, in the first instance, we have to compute $\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0\cap G_i^0}+\sigma_{\Omega_i}\right)}$ which stands as the proximal operator of the sum of two functions while in the second one, we need to compute $\operatorname{prox}_{\bar{\lambda}^{-1}\left(\delta_{C_i^0}+\sigma_{L_i}+\sigma_{\Omega_i}\right)}$ which corresponds to the proximal operator of the sum of three functions. The issue, which exceeds the scope of our present work, has been partially addressed in [1] where the authors proposed the notion of f-proximal operator of g denoted by prox_g^f and that generalizes the classical notion of the proximal operator of g.

Remark 25. In [14], the autors considered the particular case where $\mathcal{T}_{\Omega_i,\gamma_{G_i}}^{C_i} = \gamma_{C_i}$ in which case they dealt with $\operatorname{prox}_{\bar{\lambda}^{-1}\delta_{C_i^0}} = \mathrm{P}_{C_i^0}$ which can be computed with relative ease. Indeed, in that instance $(\bar{y}, \bar{\theta})$ are given by

$$\bar{y} = x - \mathrm{P}_{C_i^0}\left(\bar{\lambda}^{-1}x\right) \quad \text{and} \quad \theta = \bar{\lambda} + \xi,$$

where $\bar{\lambda} > 0$ is a solution of an equation of the form

$$\lambda + \xi = \langle x, \mathrm{P}_{C_i^0}\left(\lambda^{-1}x\right)\rangle_X - \lambda \left\| \mathrm{P}_{C_i^0}\left(\lambda^{-1}x\right)\right\|_X^2.$$

While we have proposed and proven the previous results, our upcoming numerical tests will focus on examples where a closed-form representation of the proximal operators is available.
Let us next deploy the second method that we examined in our study.

2.5.2 Mirror descent algorithm

Mirror descent algorithm is a first-order optimization method generalizing gradient descent which involves minimizing a regularized objective in each iteration.
Let us begin by presenting the framework on which relies this method. We suppose, for simplicity, that the optimization takes place in the Euclidean space $\mathbb{R}^n$.
The main idea of mirror descent is to explicitely distinguish between the primal and dual spaces and to specify a useful bijection (called mirror map) between this two spaces. For that purpose, we introduce a function $\Phi : \mathbb{R}^n \to \mathbb{R}$.
Instead of taking gradient steps in the primal space, the mirror descent method takes gradient steps in the dual space. The bijection $\nabla\Phi$ and its inverse $(\nabla\Phi)^*$ are used to map back and forth between primal points and dual points. While our focus is on constrained optimization over a convex set, say C, the gradient step may have produced a point outside of C. To address this issue, we project that point onto C under the so-called Bregman divergence D_Φ. Let us first introduce this notion.

Definition 2.5.6. Let C be a closed and convex subset of $\mathbb{R}^n$. Let $\Phi : C \to \mathbb{R}$ be a continuously-differentiable and convex function. The first-order approximation of Φ at $x \in C$ is

$$\Phi(x) = \Phi(y) + \langle \nabla\Phi(y), x - y\rangle, \quad \forall y \in C.$$

The Bregman divergence is defined to be

$$D_\Phi(x, y) = \Phi(x) - \Phi(y) - \langle \nabla\Phi(y), x - y\rangle, \quad \forall x \in C, \quad \forall y \in C.$$

The projection of y onto C under the Bregman divergence is

$$\mathrm{P}_C^\Phi(y) = \arg\min_{x \in X} D_\Phi(x, y).$$

Remark 26. The Bregman divergence possesses several properties that we choose to omit. For details about this notion, we refer to the existing literature.
We claim that the projection under the Bregman divergence satisfies, like the usual projection, the property

$$\langle \nabla\Phi(y) - \nabla\Phi\left(\mathrm{P}_C^\Phi(y)\right), x - \mathrm{P}_C^\Phi(y)\rangle \leq 0, \quad \forall x \in X.$$

Remark 27. It is worth-mentioning that if $\Phi(x) = \frac{1}{2}\|x\|^2$, then $D_\Phi(x,y) = \frac{1}{2}\|x-y\|^2$. We will restrict ourselves to this case in our experiments.

Definition 2.5.7. Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper, convex and lower semicontinuous function and $H : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper, convex, lower semicontinuous and σ-strongly convex function where $\sigma > 0$. The Bregman-proximal operator of f with respect to H is defined as

$$\begin{aligned}\operatorname{prox}^H_{\gamma f}: \quad & \operatorname{dom}\nabla H \to \mathbb{R}^n \\ & x \mapsto \operatorname{argmin}_{y\in X}\left\{f(y) + D_H(y,x)\right\}.\end{aligned}$$

Remark 28. When $H = \frac{1}{2}\|\cdot\|^2$, the Bregman-proximal operator reduces to the classical proximal operator.

Our task is to implement in Algorithm 6 a deterministic version of the stochastic incremental mirror descent algorithm with Nesterov smoothing .
Let us make the following assumptions.

- $C \subseteq \mathbb{R}^n$ is a nonempty convex and closed set.
- $f_i : \mathbb{R}^p \to \mathbb{R}$ is a proper convex and semicontinuous function for $i = 1,\ldots,m$.
- $A_i : \mathbb{R}^n \to \mathbb{R}^p$ is a linear operator such that

$$f \circ A_i(x) = \sup_{u\in\operatorname{dom} f^*}\left\{\langle A_i x, u\rangle - f^*(u)\right\}, \quad \forall x \in \mathbb{R}^p, \qquad \text{for } i,\cdots,m.$$

- $g : \mathbb{R}^n \to \mathbb{R}$ is a proper, convex and lower semicontinuous function satisfying

$$C \cap \left(\bigcap_{i=1}^m \operatorname{dom}(f_i \circ A_i)\right) \cap \operatorname{dom} g \neq \emptyset.$$

Consider the problem

$$\inf_{x\in C}\left\{\sum_{i=1}^n f_i(A_i x) + g(x)\right\}. \tag{P_A}$$

Employing the smoothing of the functions f_i for $i = 1,\ldots,m$ via the Moreau-envelope (1.8), which was suggested in [6, Section 4], we obtain

$$\left(f_i^\gamma \circ A_i\right)(x) = \left(f_i \Box \frac{1}{2\gamma}\|\cdot\|^2\right)(A_i x), \quad \forall x \in \mathbb{R}^n$$

while the gradients are given by virtue of Proposition 1.1.32 as

$$\nabla\left(f_i^\gamma \circ A_i\right)(x) = \gamma^{-1}A_i^*\left(A_i x - \operatorname{prox}_{\gamma f_i}(A_i x)\right), \quad \forall x \in \mathbb{R}^n.$$

We adapt Algorithm 6 to our minimax location problem, considering it within a deterministic framework with a specific function H.
Indeed, taking $H(x) = \frac{1}{2}\|x\|^2$, one deduces immediately that

$$H^*(x) = \frac{1}{2}\|x\|^2, \quad \nabla H(x) = \nabla H^*(x) = x, \quad \operatorname{prox}^H_{\nu g} = \operatorname{prox}_{\nu g}, \quad \forall x \in \mathbb{R}^n, \quad \forall \nu > 0.$$

In particular, the function H verifies the assumptions given in [6, Problem 4.1]. Thus, considering parameters $\gamma^k > 0$, stepsizes $\nu^k > 0$, for $k \geq 0$, and a deterministic setting we obtain the following instance of mirror descent proximal point algorithm.

Algorithm 3 Mirror Descent Algorithm

```
1: **Input:** Choose $x^0 \in C$, the smoothing parameters $\gamma^k > 0$ and the step sizes $\nu^k > 0$, for $k \geq 0$.
2: **For** $k \geq 0$ **do**
3: $\psi_0^k := x^k$
4: **for** $i := 1, \dots, m$ **do**
5:     $\psi_i^k := \psi_{i-1}^k - \frac{\nu^k}{\gamma^k} A_i^* \left( A_i \psi_{i-1}^k - \operatorname{prox}_{\gamma^k f_i} \left( A_i \psi_{i-1}^k \right) \right)$
6: **end for**
7: $x^{k+1} := \operatorname{prox}_{\nu^k g}(\psi_m^k)$
8: **End for**
```

Remark 29. It is important to note that our choice to focus on a deterministic framework is driven by the fact that a stochastic approach is typically advantageous for problems with very large dimensions. Since this is not the case here, a deterministic approach is more suitable for our purposes.

The convergence of Algorithm 3 is guaranteed by the following two results, which are based on [6, Theorem 4.2 and Corollary 4.2]

Theorem 2.5.8. *Let the sequence $(x_k)_k$ be generated by Algorithm 3. Let $\delta > 0$ and let $\gamma^k := \nu^k \frac{\delta}{\sigma}$. Then for all $N \geq 1$ and all $y \in \mathbb{R}^n$, one has*

$$\min_{0 \leq k \leq N-1} \left(\sum_{i=1}^m f_i \circ A_i + g \right)(x_{k+1}) - \left(\sum_{i=1}^m f_i \circ A_i + g \right)(y)$$
$$\leq \frac{\frac{1}{2}\|y - x^0\|^2 + \frac{1}{\sigma}\left(\delta \sum_{i=1}^m D_{\operatorname{dom} f_i^*} + 4 \left(\sum_{i=1}^m \|A_i\| \sqrt{D_{\operatorname{dom} f_i^*}} \right)^2 \left(\sqrt{m} + \frac{3}{2} + m \right) \right) \sum_{k=0}^{N-1} \nu_k^2}{\sum_{k=0}^{N-1} \nu_k}.$$

Corollary 2.5.9. *Let $x^* \in$ dom H be an optimal solution to (P_A). Let $\delta > 0$ and let $\gamma^k := \nu^k \frac{\delta}{\sigma}$. Then the optimal stepsize for Algorithm 3 is given by*

$$\nu^k := \sqrt{\frac{\frac{\sigma}{2}\|y - x^0\|^2}{\delta \sum_{i=1}^m D_{\operatorname{dom} f_i^*} + 4 \left(\sum_{i=1}^m \|A_i\| \sqrt{D_{\operatorname{dom} f_i^*}} \right)^2 \left(\sqrt{m} + \frac{3}{2} + m \right)}} \frac{1}{\sqrt{k}}, \quad \forall k \geq 0,$$

which yields for every $N \geq 1$

$$\min_{0 \leq k \leq N-1} \left(\sum_{i=1}^m f_i \circ A_i + g \right)(x_k) - \left(\sum_{i=1}^m f_i \circ A_i + g \right)(x^*)$$
$$\leq 2 \sqrt{\frac{\frac{1}{2}\|y - x^0\|^2 \left(\delta \sum_{i=1}^m D_{\operatorname{dom} f_i^*} + 4 \left(\sum_{i=1}^m \|A_i\| \sqrt{D_{\operatorname{dom} f_i^*}} \right)^2 \left(\sqrt{m} + \frac{3}{2} + m \right) \right)}{\sigma}} \frac{1}{\sqrt{N}}.$$

Remark 30. In our numerical tests, detailed in the next section, we opted for a constant stepsize ν, as it yielded more effective results.

2.6 Numerical experiments

The experiments were conducted on a DELL Latitude 7420 PC equipped with an 11th Gen Intel(R) Core(TM) i7-1185G7 processor running at 3.00 GHz with a base frequency of 1.80 GHz.
Before presenting our numerical results, it is important to clarify that, for all subsequent tests, the origin was consistently used as the starting point. Additionally, to ensure a fair comparison between the algorithms, we employed a uniform stopping criterion across all tests, defined by the accuracy levels $\epsilon_1 = 10^{-3}$, $\epsilon_2 = 10^{-4}$, $\epsilon_3 = 10^{-6}$ and $\epsilon_4 = 10^{-8}$ where for $i = 1, 2, 3, 4$, ϵ_i represents the maximum difference between the solutions obtained from two consecutive iterations.
The results presented in the tables reflect the best performance for each method, determined by the optimal choice of parameters specific to each algorithm. In this section, we will deploy our numerical tests while considering a special instance Special case two 2.4.2. Indeed, let us take

$$X = \mathbb{R}^d, \quad \gamma_{C_i} = \|.\|, \quad \Omega_i = \{p_i\}, \quad L_i = \{0_{\mathbb{R}^d}\}, \quad i = 1, \cdots, n,$$

where for $i = 1, \cdots, n$, p_i is a given point in $\mathbb{R}^d$.
Hence, we have, for all $i = 1, \cdots, n$ and for all $x \in \mathbb{R}^d, T^{C_i}_{\Omega_i,\delta_{L_i}}(x) = \|x - p_i\|$, and $P_{\mathcal{T}}$ rewrites

$$\inf_{(x,t)\in X\times\mathbb{R}} \left\{ t + \sum_{i=1}^{n} \delta_{\text{epi}\|.-p_i\|}(x,t) \right\}. \tag{2.21}$$

For the implementation of Algorithm 2, we will require Lemma 3.6.5 and corollary 3.6.4 which were given and proved [14].

Example 2.6.1. We first consider a basic example where $d = 2$ and $n = 3$.
Take $p_1 = (2, -1)^T, p_2 = (-3, 2)^T, p_3 = (4, 5)^T$.

Remark 31. We can make the following interpretation. The three points represent the location of three given facilities and the goal is to determine the coordinates in the plane (we treat a simplified version of the real-world problem, which would typically lie in a higher-dimensional space) for the location of a new facility to be constructed. The problem involves minimizing the maximum distance between the existing facilities and the new facility to be implemented.

We compare our fundings with the ones obtained using the parallel splitting method proposed by [11]. We recall that they implemented in Algorithm 3.6.2 a version of the Douglas-Rachford method for both the primal and dual problems.

MATLAB computed via the Chambolle-Pock algorithm (as well as the primal parallel splitting), an optimal solution illustrated in Figure 2.1a given by

$$\bar{x} = (0.8333, 2.7222)^\top, \quad \bar{t} = 3.9008,$$

while the optimal primal objective value is $v(P) = 3.9008$.

The results presented in Table 2.1 - Table 2.4 show that the Chambolle-Pock method consistently outscores both its primal and dual splitting counterparts across all error tolerances, excelling in both the number of iterations required for convergence and computational time. We will provide a more detailed comparison when we will examine the next example that involves a greater number of points.

When implementing the mirror descent, we run our MATLAB programs for various stepsizes $\nu > 0$ as well as multiple parameters $\delta > 0$.
We observed that the results were significantly influenced by the choice of the parameters, particularly the value of ν. To illustrate this, we present in Table 2.5 - Table 2.7 the results obtained by

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Chambolle-Pock** ($\sigma = \tau = 0.83, \theta = 1$)
Number of iterations	174	60	23
CPU time (seconds)	0.0310	0.0224	0.0136
Objective value	3.9012	3.8997	3.9008

Table 2.1: Comparison of splitting and Chambolle-Pock methods ($d = 2, n = 3$) for $\epsilon_1 = 10^{-3}$

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Chambolle-Pock** ($\sigma = \tau = 0.83, \theta = 1$)
Number of iterations	223	106	31
CPU time (seconds)	0.0400	0.0240	0.0116
Objective value	3.9008	3.9008	3.9008

Table 2.2: Comparison of splitting and Chambolle-Pock methods ($d = 2, n = 3$) for $\epsilon_2 = 10^{-4}$

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Chambolle-Pock** ($\sigma = \tau = 0.83, \theta = 1$)
Number of iterations	344	197	47
CPU time (seconds)	0.0442	0.0290	0.0120
Objective value	3.9008	3.9008	3.9008

Table 2.3: Comparison of splitting and Chambolle-Pock methods ($d = 2, n = 3$) for $\epsilon_3 = 10^{-6}$

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Chambolle-Pock** ($\sigma = \tau = 0.83, \theta = 1$)
Number of iterations	469	288	63
CPU time (seconds)	0.0521	0.0298	0.0132
Objective value	3.9008	3.9008	3.9008

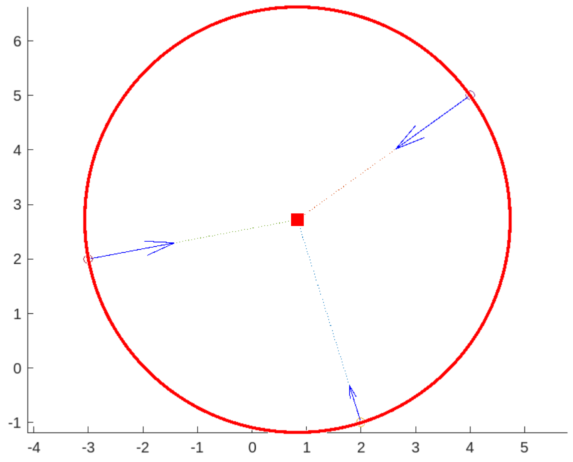
Table 2.4: Comparison of splitting and Chambolle-Pock methods ($d = 2, n = 3$) for $\epsilon_4 = 10^{-8}$

Mirror descent ($\delta = 1$)	($\nu = 0.1$)	($\nu = 0.01$)	($\nu = 0.001$)	($\nu = 0.0001$)
Number of iterations	86	306	2564	25241
CPU time (seconds)	0.0273	0.0460	0.2297	1.9225
Optimal primal solution	$(0.8667, 2.6303)^T$	$(0.8371, 2.7122)^T$	$(0.8337, 2.7212)^T$	$(0.8334, 2.7221)^T$
Objective value	3.7050	3.8810	3.8988	3.9006

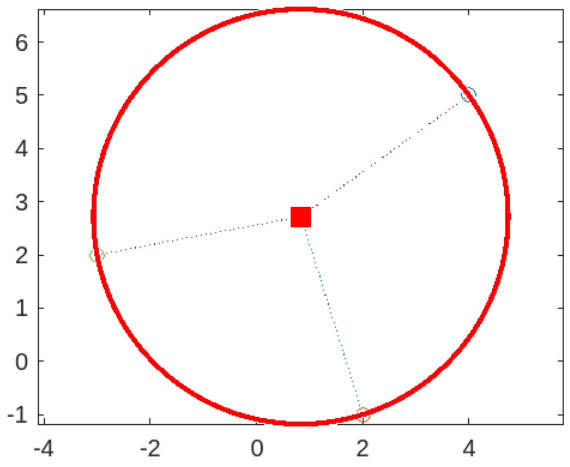
Table 2.5: Performance of the mirror descent method with ($d = 2, n = 3$) for $\epsilon_4 = 10^{-8}$

varying the values of the parameters δ and ν while considering the best accuracy related to error tolerance $\epsilon_4 = 10^{-8}$

As ν decreases from 0.1 to 0.0001, the number of iterations required for convergence increases dramatically. The increase in computational time is consistent with the growing number of itera-



(a) Optimal solution via Chambolle-Pock



(b) Optimal solution via mirror descent

Figure 2.1: Visualization of an optimal solution to problem $(d = 2, n = 3)$ via Chambolle-Pock and mirror descent methods

Mirror descent $(\delta = 0.6127)$	$(\nu = 0.1)$	$(\nu = 0.01)$	$(\nu = 0.001)$	$(\nu = 0.0001)$
Number of iterations	56	331	3115	30990
CPU time (seconds)	0.0327	0.0510	0.3263	2.3271
Optimal primal solution	$(0.8545, 2.6654)^T$	$(0.8356, 2.7163)^T$	$(0.8336, 2.72196)^T$	$(0.8334, 2.7222)^T$
Objective value	3.7892	3.8896	3.8997	3.9007

Table 2.6: Performance of the mirror descent method with $(d = 2, n = 3)$ for $\epsilon_4 = 10^{-8}$

Mirror descent $(\delta = 0.334)$	$(\nu = 0.1)$	$(\nu = 0.01)$	$(\nu = 0.001)$	$(\nu = 0.0001)$
Number of iterations	198	670	5471	53708
CPU time (seconds)	0.0439	0.0770	0.4426	3.8048
Optimal primal solution	$(0.8447, 2.6926)^T$	$(0.8345, 2.7192)^T$	$(0.8335, 2.7219)^T$	$(0.8333, 2.7222)^T$
Objective value	3.8505	3.8958	3.9003	3.9007

Table 2.7: Performance of the mirror descent method with $(d = 2, n = 3)$ for $\epsilon_4 = 10^{-8}$

tions, indicating a trade-off between the choice of ν and the the desired efficiency.

MATLAB computed, via the mirror descent algorithm, an optimal solution illustrated in Figure 2.1b given by

$$\bar{x} = (0.8334, 2.7219)^\top, \quad \bar{t} = 3.9002,$$

while the optimal primal objective value is $v(P) = 3.9002$. In the following we compare the performance of the mirror descent with the splitting methods for two values of the parameter ν and a constant value $\delta = 0.6127$.

The first observation is that the mirror descent algorithm requires for both choices of ν a higher number of iterations and more computational time compared to both splitting methods.
With $(\delta = 0.6127, \nu = 0.0005)$, the mirror descent algorithm requires significantly more iterations and computational time. Despite this, it produces an objective value very close to that of the primal and dual splitting methods, suggesting that while the solution quality is comparable, the

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Mirror descent** ($\delta = 0.6127$) ($\nu = 0.005$)	($\nu = 0.0005$)
Number of iterations	174	60	618	11
CPU time (seconds)	0.0310	0.0224	0.0705	0.0236
Objective value	3.9012	3.8997	3.8953	4.1969

Table 2.8: Comparison of splitting and mirror descent methods ($d = 2, n = 3$) for $\epsilon_1 = 10^{-3}$

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Mirror descent** ($\delta = 0.6127$) ($\nu = 0.005$)	($\nu = 0.0005$)
Number of iterations	223	106	622	6194
CPU time (seconds)	0.0400	0.0240	0.0747	0.5993
Objective value	3.9008	3.9008	3.8952	3.9002

Table 2.9: Comparison of splitting and mirror descent methods ($d = 2, n = 3$) for $\epsilon_2 = 10^{-4}$

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Mirror descent** ($\delta = 0.6127$) ($\nu = 0.005$)	($\nu = 0.0005$)
Number of iterations	344	197	631	6202
CPU time (seconds)	0.0442	0.0290	0.0736	0.5188
Objective value	3.9008	3.9008	3.8952	3.9002

Table 2.10: Comparison of splitting and mirror descent methods ($d = 2, n = 3$) for $\epsilon_3 = 10^{-6}$

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Mirror descent** ($\delta = 0.6127$) ($\nu = 0.005$)	($\nu = 0.0005$)
Number of iterations	469	288	640	6211
CPU time (seconds)	0.0521	0.0298	0.0752	0.5755
Objective value	3.9008	3.9008	3.8952	3.9002

Table 2.11: Comparison of splitting and mirror descent methods ($d = 2, n = 3$) for $\epsilon_4 = 10^{-8}$

increased computational effort does not yield a substantially better outcome.
While the mirror descent method with ($\delta = 0.6127, \nu = 0.0005$) offers marginal improvements in objective value, it requires significantly more computational time and iterations compared to using ($\delta = 0.6127, \nu = 0.0005$).

In the following example which involves larger data ($n = 10$), we will examine whether the mirror descent method can outperform its splitting counterparts by appropriately selecting the parameters. We will also emphasize the efficiency of the Chambolle-Pock method in the context of our minimax location problem.

Example 2.6.2. Set $n = 10$ and consider the points $p_1 = (2,5)^T, p_2 = (4,-3)^T, p_3 = (1,-5)^T, p_4 = (7,-6)^T, p_5 = (6,1)^T, p_6 = (3,-5)^T, p_7 = (6,-3)^T, p_8(-2,3)^T, p_9 = (4,3)^T, p_{10} = (2,-7)^T$.
We run our MATLAB programs for various stepsizes $\sigma, \tau > 0$ for the Chambolle-Pock algorithm

and ν for the mirror descent as well as multiple parameters $0 \leq \theta \leq 1$ and δ respectively.

MATLAB computed, via both the Chambolle-Pock and the mirror descent algorithms (as well as the primal parallel splitting), an optimal solution illustrated in Figure 2.2a and Figure 2.2b respectively given by

$$\bar{x} = (2.6667, -1.3333)^\top, \quad \bar{t} = 6.3683,$$

while the optimal primal objective value is $v(P) = 6.3683$.

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Chambolle-Pock** ($\sigma = \tau = 0.83, \theta = 1$)	**Mirror descent** ($\delta = 0.6127, \nu = 0.00005$)
Number of iterations	610	369	75	11
CPU time (seconds)	0.1371	0.0438	0.0140	0.0280
Objective value	6.3691	6.3695	6.3685	6.3687

Table 2.12: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 2, n = 10$) for $\epsilon_1 = 10^{-3}$

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Chambolle-Pock** ($\sigma = \tau = 0.83, \theta = 1$)	**Mirror descent** ($\delta = 0.6127, \nu = 0.00005$)
Number of iterations	817	643	102	13
CPU time (seconds)	0.2161	0.0568	0.0145	0.0340
Objective value	6.3683	6.3683	6.3683	6.3692

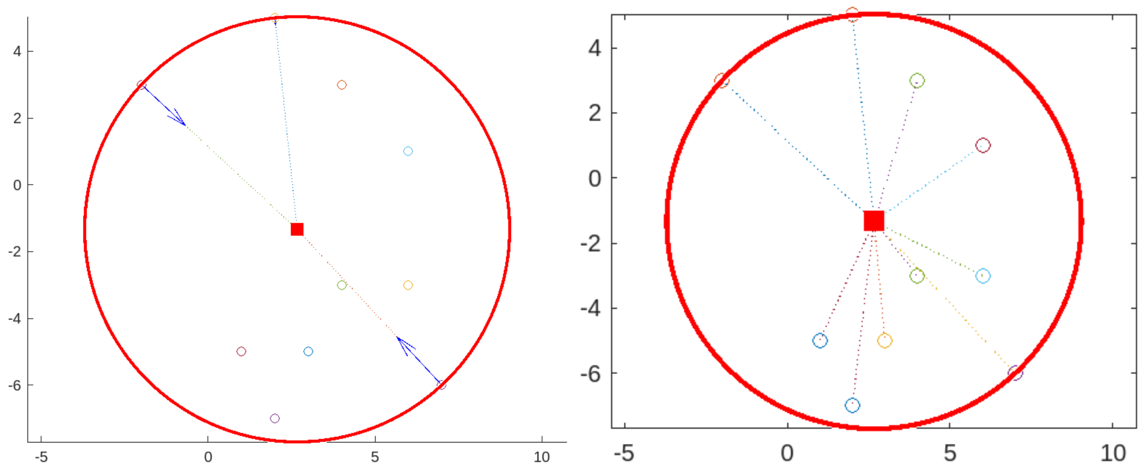
Table 2.13: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 2, n = 10$) for $\epsilon_2 = 10^{-4}$

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Chambolle-Pock** ($\sigma = \tau = 0.83, \theta = 1$)	**Mirror descent** ($\delta = 0.6127, \nu = 0.00005$)
Number of iterations	1232	1204	181	62
CPU time (seconds)	0.3118	0.0965	0.0180	0.0411
Objective value	6.3683	6.3683	6.3683	6.3683

Table 2.14: Comparison of splitting and Chambolle-Pock methods ($d = 2, n = 10$) for $\epsilon_3 = 10^{-6}$

Algorithm	**Primal** ($\nu = 24$)	**Dual** ($\nu = 0.076$)	**Chambolle-Pock** ($\sigma = \tau = 0.83, \theta = 1$)	**Mirror descent** ($\delta = 0.6127, \nu = 0.00005$)
Number of iterations	1647	1789	263	95
CPU time (seconds)	0.3995	0.1579	0.0202	0.0559
Objective value	6.3683	6.3683	6.3683	6.3683

Table 2.15: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 2, n = 10$) for $\epsilon_4 = 10^{-8}$



(a) Optimal solution via Chambolle-Pock (b) Optimal solution via mirror descent

Figure 2.2: Visualization of an optimal solution to problem $(d = 2, n = 10)$ via Chambolle-Pock and mirror descent methods

The first observation of the results presented in Table 2.12 - Table 2.15 is that -in contrast with our previous example- the mirror descent algorithm achieves convergence remarkably quickly, requiring only 13 iterations for a tolerance of $\epsilon_2 = 10^{-4}$ and 95 iterations for $\epsilon_4 = 10^{-8}$. In contrast, the two splitting methods are significantly slower, needing at least 600 iterations and up to 1800 iterations to converge for these same error tolerances. Furthermore, this method exhibits faster computational times than both the primal and dual methods across different error tolerances. For example, with $\epsilon_4 = 10^{-8}$, it converges eight times faster than the primal method and three times faster than the dual, all while achieving the same objective value.
[11], the authors have already demonstrated that the dual splitting algorithm requires fewer iterations than the primal version. However, the Chambolle-Pock algorithm significantly outperforms both, requiring 8 times fewer iterations than the primal algorithm and 6 times fewer than the dual algorithm to reach convergence. Additionally, the Chambolle-Pock method exhibits faster computational times than both the primal and dual methods across different error tolerances. Precisely, with an error tolerance of $\epsilon_1 = 10^{-6}$, it converges 17 times faster than the primal method and 5 times faster than the latter's dual counterpart. At $\epsilon_4 = 10^{-8}$ accuracy, the Chambolle-Pock algorithm requires 20 times less computational time than the primal splitting and almost 7 times less than its dual counterpart, all while achieving the same objective value.
While comparing between the Chambolle-Pock and the mirror descent methods, the latter demonstrates in this instance superior efficiency in terms of the number of iterations required for convergence at both tolerance levels. In terms of computational time, while both methods outperform the parallel splitting algorithms, the Chambolle-Pock method is faster than the mirror descent for all error tolerances (for example, at ϵ_4, 0.0202 compared to 0.0559 second). The objective values obtained by both methods are very similar, indicating that both algorithms are effective in the context of this facility location problem.

Example 2.6.3. Let us now consider an instance where $n = 100$. The points were generated using MATLAB's *rand* function, which produces uniformly distributed random numbers.

MATLAB computed

- via the Chambolle-Pock algorithm, an optimal solution illustrated in Figure 2.3a given by

$$\bar{x} = (3.7488, 7.3316)^\top, \quad \bar{t} = 33.4969,$$

while the optimal primal objective value is $v(P) = 33.4969$,

- via the mirror descent, an optimal solution illustrated in Figure 2.3a given by

$$\bar{x} = (3.7506, 7.32153)^\top, \quad \bar{t} = 33.4860,$$

while the optimal primal objective value is $v(P) = 33.4860$.

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent**
	$(\nu = 27)$	$(\nu = 0.0046)$	$(\sigma = \tau = 0.69, \theta = 1)$	$(\delta = 1, \nu = 0.005)$
Number of iterations	6549	2070	628	30
CPU time (seconds)	16.9174	2.3513	0.3182	0.1708
Objective value	33.4964	33.5087	33.4972	33.9451

Table 2.16: Comparison of splitting, Chambolle-Pock and mirror descent methods $(d = 2, n = 100)$ for $\epsilon_1 = 10^{-3}$

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent**
	$(\nu = 27)$	$(\nu = 0.0046)$	$(\sigma = \tau = 0.69, \theta = 1)$	$(\delta = 1, \nu = 0.005)$
Number of iterations	8627	6057	670	5695
CPU time (seconds)	24.0446	3.9127	0.3763	18.3372
Objective value	33.4968	33.5003	33.4969	33.4859

Table 2.17: Comparison of splitting, Chambolle-Pock and mirror descent methods $(d = 2, n = 100)$ for $\epsilon_2 = 10^{-4}$

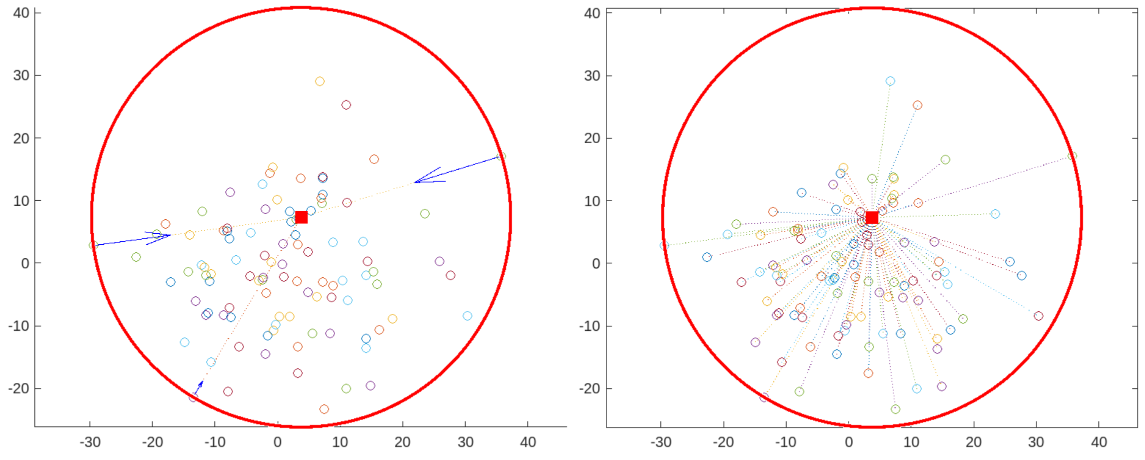
Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent**
	$(\nu = 27)$	$(\nu = 0.0046)$	$(\sigma = \tau = 0.69, \theta = 1)$	$(\delta = 1, \nu = 0.005)$
Number of iterations	12780	15914	755	5749
CPU time (seconds)	40.0198	10.0492	0.3789	17.4388
Objective value	33.4969	33.4970	33.4969	33.4860

Table 2.18: Comparison of splitting, Chambolle-Pock and mirror descent methods $(d = 2, n = 100)$ for $\epsilon_3 = 10^{-6}$

Table 2.16 - Table 2.19 highlight that mirror descent shows, compared to the splitting methods and the Chambolle-Pock algorithm, some difficulty in converging to the exact objective value in this case. While the other three methods successfully reached the optimal value of $v(P) = 33.4969$, the mirror descent converged to a slightly lower value of $v(P) = 33.4860$. Additionally, as demonstrated in the previous example, the method's high computational cost underscores its sensitivity to parameter choices, particularly the value of ν.

Algorithm	**Primal** ($\nu = 27$)	**Dual** ($\nu = 0.0046$)	**Chambolle-Pock** ($\sigma = \tau = 0.69, \theta = 1$)	**Mirror descent** ($\delta = 1, \nu = 0.005$)
Number of iterations	16936	25670	840	5803
CPU time (seconds)	59.6376	20.8613	0.4375	22.3911
Objective value	33.4969	33.4969	33.4969	33.4860

Table 2.19: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 2, n = 100$) for $\epsilon_4 = 10^{-8}$



(a) Optimal solution via Chambolle-Pock (b) Optimal solution via mirror descent

Figure 2.3: Visualization of an optimal solution to problem ($d = 2, n = 100$) via Chambolle-Pock and mirror descent methods

When considering a large number of points, the advantages of the Chambolle-Pock algorithm over the splitting methods become even more outstanding. For instance, choosing an error tolerance of $\epsilon_2 = 10^{-4}$, it showcases convergence 65 times faster than the primal splitting while necessitating 13 times fewer iterations (respectively 9 times faster and 10 times fewer iterations than the dual). With the highest accuracy of $\epsilon_4 = 10^{-8}$, the Chambolle-Pock method converges to the solution with 20 times fewer iterations compared to the primal splitting method, while the computational cost is reduced by a factor of 150 ! (0.4 second for the Chambolle-Pock method versus 59.6 seconds for the primal splitting method). The ratio becomes 1/50 when comparing it to the dual method which is remarkable taking into account that the latter doesn't explicitely yields the optimal solution.

In the next examples, we will implement our method in $\mathbb{R}^3$ which is arguably a more suitable space for studying location problems.

Example 2.6.4. Let us assert the previous tendancies by considering the example given in $\mathbb{R}^3$ by [11] where $d = 3, n = 7$, with $p_1 = (-8, 8, 8)^T, p_2 = (-7, 0, 0)^T, p_3 = (-4, -1, 1)^T, p_4 = (2, 0, 2)^T, p_5 = (2, -6, 2)^T, p_6 = (7, 1, 1)^T, p_7 = (6, 5, 4)^T$, we have the following results.

MATLAB computed

- via the Chambolle-Pock algorithm -as well as the primal parallel splitting-, an optimal solution illustrated in Figure 2.4a given by

$$\bar{x} = (-1.5166, 2.2381, 4.5835)^\top, \quad \bar{t} = 9.3224,$$

while the optimal primal objective value is $v(P) = 9.3224$,

- via the mirror descent algorithm with $\nu = 0.005$, an optimal solution illustrated in Figure 2.4b given by
$$\bar{x} = (-1.5181, 2.2364, 4.5839)^\top, \quad \bar{t} = 9.3199,$$
while the optimal primal objective value is $v(P) = 9.3199$.

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent** ($\delta = 0.33$)	
	($\nu = 10$)	($\nu = 0.055$)	($\sigma = \tau = 0.83, \theta = 1$)	($\nu = 0.05$)	($\nu = 0.005$)
Number of iterations	473	142	70	531	3727
CPU time (seconds)	0.1026	0.0293	0.0217	0.1225	0.7506
Objective value	9.3228	9.3358	9.3224	9.2995	9.3298

Table 2.20: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 3, n = 7$) for $\epsilon_1 = 10^{-3}$

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent** ($\delta = 0.33$)	
	($\nu = 10$)	($\nu = 0.055$)	($\sigma = \tau = 0.83, \theta = 1$)	($\nu = 0.05$)	($\nu = 0.005$)
Number of iterations	615	260	97	958	5248
CPU time (seconds)	0.1333	0.0319	0.0228	0.2480	1.0133
Objective value	9.3223	9.3229	9.3224	9.2977	9.3218

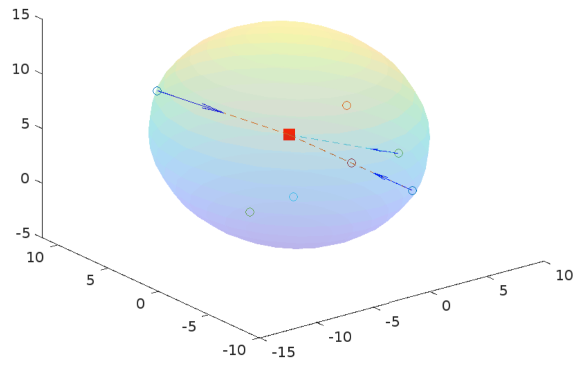
Table 2.21: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 3, n = 7$) for $\epsilon_2 = 10^{-4}$

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent** ($\delta = 0.33$)	
	($\nu = 10$)	($\nu = 0.055$)	($\sigma = \tau = 0.83, \theta = 1$)	($\nu = 0.05$)	($\nu = 0.005$)
Number of iterations	897	503	151	1817	13836
CPU time (seconds)	0.1694	0.0423	0.0262	0.3719	3.0440
Objective value	9.3224	9.3224	9.3224	9.2977	9.3199

Table 2.22: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 3, n = 7$) for $\epsilon_3 = 10^{-6}$

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent** ($\delta = 0.33$)	
	($\nu = 10$)	($\nu = 0.055$)	($\sigma = \tau = 0.83, \theta = 1$)	($\nu = 0.05$)	($\nu = 0.005$)
Number of iterations	1180	743	205	2675	22422
CPU time (seconds)	0.2078	0.0514	0.0278	0.5360	5.1840
Objective value	9.3224	9.3224	9.3224	9.2977	9.3199

Table 2.23: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 3, n = 7$) for $\epsilon_4 = 10^{-8}$


(a) Optimal solution via Chambolle-Pock

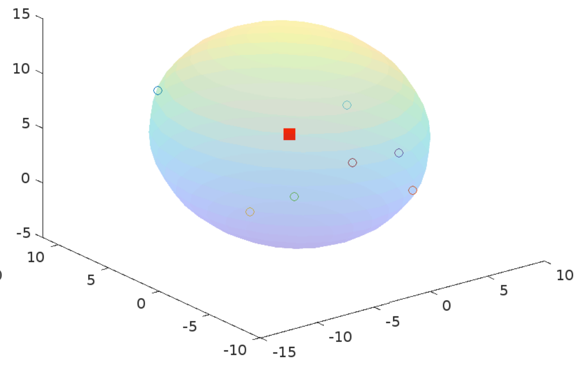

(b) Optimal solution via mirror descent

Figure 2.4: Visualization of an optimal solution to problem ($d = 3, n = 7$) via Chambolle-Pock and mirror descent methods

Table 2.20 - Table 2.23 indicate Chambolle-Pock method once again significantly surpasses the two parallel splitting algorithms, both in the number of iterations required for convergence and in computational efficiency. For instance, when considering the tolerance $\epsilon_1 = 10^{-3}$, the Chambolle-Pock algorithm converges faster than the dual (twice as fast as the primal) while requiring half as many iterations than the dual parallel splitting method (and almost 7 times less comparing to the primal). At the highest accuracy considered, $\epsilon_4 = 10^{-8}$, the Chambolle-Pock algorithm converges twice as fast as the dual (and ten times faster than the primal) while requiring almost 4 times fewer iterations than the dual splitting method (and 5 times less comparing to the primal).

The final two examples of this section will enable us to assess the performance of the Chambolle-Pock and the mirror descent algorithms in $\mathbb{R}^3$ while considering a large number of points.

Example 2.6.5. Let us deploy a particular case in $\mathbb{R}^3$ with 50 points. The points were generated using MATLAB's *rand* function, which produces uniformly distributed random numbers.
MATLAB computed,

- via the Chambolle-Pock algorithm -as well as the primal parallel splitting-, an optimal solution illustrated in Figure 2.5a given by
$$\bar{x} = (-1.8804, -2.3775, -0.4515)^\top, \quad \bar{t} = 15.4233,$$
while the optimal primal objective value is $v(P) = 15.4233$.
- via the mirror descent algorithm with $\nu = 0.01$, an optimal solution illustrated in Figure 2.5b given by
$$\bar{x} = (-1.8768, -2.3748, -0.4570)^\top, \quad \bar{t} = 15.4025,$$
while the optimal primal objective value is $v(P) = 15.4025$.

Per Table 2.24- Table 2.27, our first observation-unrelated to both Chambolle-Pock and mirror descent algorithms-is that, unlike in all previous cases, the dual parallel method requires more iterations than its primal counterpart (up to twice as many fo for $\epsilon_4 = 10^{-8}$). It still reaches an optimal solution faster than the primal method, completing the task in half the time for the same accuracy.
We could argue that, with $\nu = 0.05$, the mirror descent consistently requires fewer iterations and

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent** ($\delta = 1$)	
	($\nu = 10$)	($\nu = 0.055$)	($\sigma = \tau = 0.57, \theta = 1$)	($\nu = 0.05$)	($\nu = 0.01$)
Number of iterations	2790	2888	194	504	634
CPU time (seconds)	3.0294	0.8104	0.0696	0.8049	1.1607
Objective value	15.4239	15.4222	15.4233	15.3226	15.4357

Table 2.24: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 3, n = 50$) for $\epsilon_1 = 10^{-3}$

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent** ($\delta = 1$)	
	($\nu = 10$)	($\nu = 0.055$)	($\sigma = \tau = 0.57, \theta = 1$)	($\nu = 0.05$)	($\nu = 0.01$)
Number of iterations	3789	5300	257	1215	3568
CPU time (seconds)	4.2070	1.3515	0.0906	1.9338	5.9118
Objective value	15.4234	15.4232	15.4233	15.3195	15.4033

Table 2.25: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 3, n = 50$) for $\epsilon_2 = 10^{-4}$

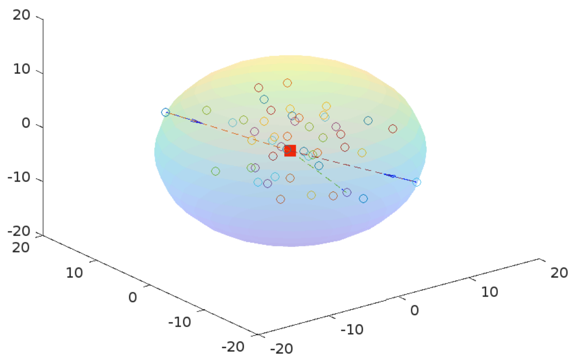
Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent** ($\delta = 1$)	
	($\nu = 10$)	($\nu = 0.055$)	($\sigma = \tau = 0.57, \theta = 1$)	($\nu = 0.05$)	($\nu = 0.01$)
Number of iterations	5786	10124	384	2635	10671
CPU time (seconds)	6.2922	2.4914	0.1353	4.2169	29.5845
Objective value	15.4233	15.4233	15.4233	15.3195	15.4025

Table 2.26: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 3, n = 50$) for $\epsilon_3 = 10^{-6}$

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**	**Mirror descent** ($\delta = 1$)	
	($\nu = 10$)	($\nu = 0.055$)	($\sigma = \tau = 0.57, \theta = 1$)	($\nu = 0.05$)	($\nu = 0.01$)
Number of iterations	7784	14947	511	4056	17773
CPU time (seconds)	8.3364	3.3432	0.20269	6.2743	44.1431
Objective value	15.4233	15.4233	15.4233	15.3195	15.4025

Table 2.27: Comparison of splitting, Chambolle-Pock and mirror descent methods ($d = 3, n = 50$) for $\epsilon_4 = 10^{-8}$

is faster in computational time compared to the primal method, regardless of the error tolerance. However, the reduced accuracy in the objective value (15.4233 for the primal and 15.3195 for mirror descent) undermines this benefit. Using $\nu = 0.05$, the computational cost, both in terms of the number of iterations and CPU time, increases significantly. This raises questions about the benefit of choosing this parameter, even though it yields an objective value that is closer to the exact one. As for the Chambolle-Pock method, it consistently delivers the best performance across all metrics and error tolerances. For instance, it achieves convergence at $\epsilon_3 = 10^{-6}$, with 15 times fewer iterations than the primal (and over 25 times fewer than the dual) while also being 48 times faster in computation speed (and nearly 20 times faster than the dual).


(a) Optimal solution via Chambolle-Pock

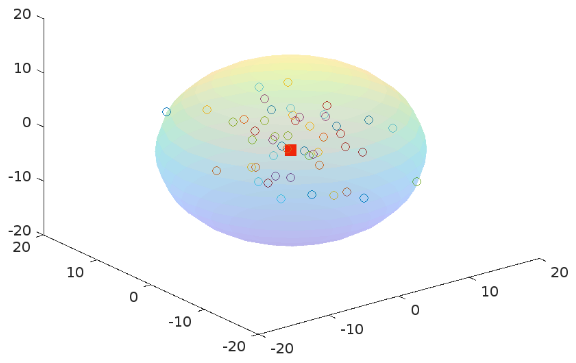

(b) Optimal solution via mirror descent

Figure 2.5: Visualization of an optimal solution to problem ($d = 3, n = 50$) via Chambolle-Pock and mirror descent methods

Example 2.6.6. Finally, let us mention an example in $\mathbb{R}^3$ with 100 points. The points were generated using MATLAB's *rand* function, which produces uniformly distributed random numbers. We will in this instance apply the Chambolle-Pock method while testing different values of θ.

MATLAB computed, via the Chambolle-Pock algorithm (as well as the parallel splitting) an optimal solution illustrated in Figure 2.6 given by

$$\bar{x} = (-4.2662, -0.0343, 0.6603)^\top, \quad \bar{t} = 28.6181,$$

while the optimal primal objective value is $v(P) = 28.6181$.

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**		
	($\nu = 10$)	($\nu = 0.055$)	($\theta = 0.33$)	($\theta = 0.5$)	($\theta = 1$)
Number of iterations	5010	4389	554	545	552
CPU time (seconds)	12.0047	3.2195	0.5792	0.4194	0.3263
Objective value	28.6184	28.5771	28.6184	28.6182	28.6182

Table 2.28: Comparison of splitting and Chambolle-Pock methods ($d = 3, n = 100$) for $\epsilon_1 = 10^{-3}$

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**		
	($\nu = 10$)	($\nu = 0.055$)	($\theta = 0.33$)	($\theta = 0.5$)	($\theta = 1$)
Number of iterations	6725	9530	658	660	669
CPU time (seconds)	16.1302	7.4406	0.6951	0.6638	0.3461
Objective value	28.6182	28.6129	28.6182	28.6182	28.6181

Table 2.29: Comparison of splitting and Chambolle-Pock methods ($d = 3, n = 100$) for $\epsilon_2 = 10^{-4}$

Table 2.28 - Table 2.31 confirm Chambolle-Pock algorithm is, across multiple values of its parameters, remarkably efficient in both computational speed and the number of iterations required for convergence. Indeed, regardless of the choice of θ, it outperforms in both metrics the primal and dual splitting algorithms. It's also worth mentioning that in their article, [11] already demonstrated that their parallel splitting method outperforms a subgradient-based algorithm.
Based on the results, the choice $\theta = 1$ appears to be the best for the Chambolle-Pock algorithm

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**		
	$(\nu = 10)$	$(\nu = 0.055)$	$(\theta = 0.33)$	$(\theta = 0.5)$	$(\theta = 1)$
Number of iterations	10639	19672	888	891	903
CPU time (seconds)	24.8141	13.3049	1.0415	0.8813	0.4963
Objective value	28.6182	28.6182	.28.6182	28.6182	28.6181

Table 2.30: Comparison of splitting and Chambolle-Pock methods $(d = 3, n = 100)$ for $\epsilon_3 = 10^{-6}$

Algorithm	**Primal**	**Dual**	**Chambolle-Pock**		
	$(\nu = 10)$	$(\nu = 0.055)$	$(\theta = 0.33)$	$(\theta = 0.5)$	$(\theta = 1)$
Number of iterations	14613	29919	1118	1123	1136
CPU time (seconds)	37.9604	20.4486	1.1534	0.9432	0.6074
Objective value	28.6181	28.6181	28.6182	28.6182	28.6181

Table 2.31: Comparison of splitting and Chambolle-Pock methods $(d = 3, n = 100)$ for $\epsilon_4 = 10^{-8}$

when considering the CPU time. Although the choice $\theta = 1$ requires slightly more iterations - at $\epsilon_4 = 10^{-8}$ accuracy, it requires 1136 iterations compared to 1123 with $\theta = 0.33$ and $\theta = 0.5$ respectively- the difference is minimal.
However, regarding computational time, the Chambolle-Pock algorithm with $\theta = 1$ achieves the fastest performance in all error tolerances. For instance, or $\epsilon_4 = 10^{-8}$, it delivers the optimal solution within 0.6074 second, significantly outperforming both $\theta = 0.5$ (0.9432 second) and $\theta = 0.33$ (1.1534 second).
The objective value is consistent across all values, confirming that the performance differences are not due to variations in solution accuracy.

Remark 32. While alternative values for θ within the range $[0, 1]$ were considered, extensive numerical testing regarding the Chambolle-Pock method showed that $\theta = 1$ consistently yielded the best performance for the Chambolle-Pock algorithm. This choice corroborated the theoretical results mentioned in Remark 23.

2.7 Discussion

- Witin our context, the Chambolle-Pock algorithm consistently excels, irrespective of the error tolerance level. It significantly outperforms the splitting methods and the mirror descent algorithm, both in terms of iteration efficiency and computational speed. Its advantages become even more pronounced when dealing with a large number of points which asserts its superiority over competing methods.

- It is important to insist on the fact that, as already showed by [11], the dual parallel splitting method is more efficient than the primal version but does not explicitly yield an optimal solution to the initial optimization problem. While an optimal solution can sometimes be recovered -via optimality conditions-, the remarkable advantage of the Chambolle-Pock method is that it not only performs outstandingly better that the dual splitting algorithm but also explicitly delivers an optimal solution to the problem at hand.

- The mirror descent algorithm tends to be less efficient than other methods, both in terms of

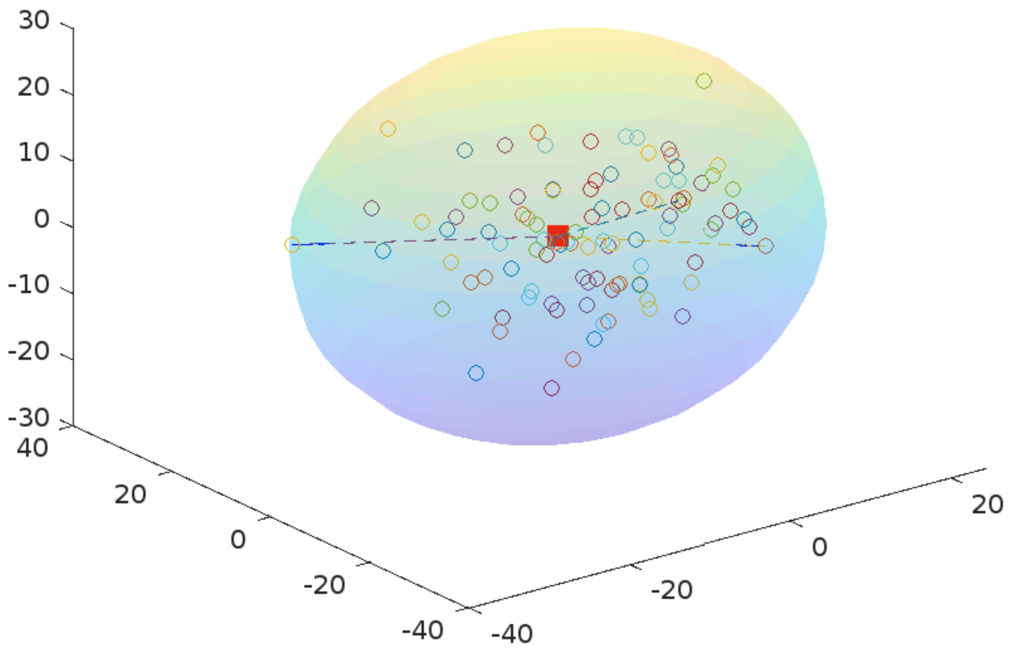


Figure 2.6: Visualization of an optimal solution to of problem ($d = 3, n = 100$ via Chambolle-Pock method)

iterations and computational time, especially when the parameter ν is small. The challenge is that while a smaller ν is required for achieving more precise convergence, it also results in increased computational demands and more iterations, thus affecting overall efficiency.
The performance of the mirror descent method is significantly affected by the choice of parameters and step sizes. This dependence influences its ability to converge to the exact optimal value, highlighting the importance of carefully selecting these parameters to balance accuracy and computational efficiency. It can be argued that implementing the mirror descent method in our context requires balancing computational efficiency with solution accuracy.

- Although more time-consuming than its splitting concurrents, the mirror descent method maintains a relatively stable number of iterations across different accuracy levels, highlighting its robustness in achieving high accuracy without a significant increase in iterations.

2.8 Perspectives on facility location optimization problems

2.8.1 Perspectives related to the Chambolle-Pock algorithm

We showed in the previous section that, in general, the Chambolle-Pock method outperforms significantly both the parallel splitting and the mirror descent. Let us now discuss the main downside that we have identify regarding the implementation of the Chambolle-Pock in the context of our minimax location problem.
In fact, in all the examples that we presented previously, we considered Special case two 2.4.2 in which case $T^{C_i}_{\Omega_i,\delta_{L_i}}$ is the extended minimal time function.

In that instance, while in [11], the authors considered the primal formulation

$$(P_{\mathcal{T}}) \qquad \min_{\substack{t>0, x\in\mathbb{R}^d,\\ y\in(\mathbb{R}^d)^n, z\in(\mathbb{R}^d)^n}} \left\{ t + \sum_{i=1}^n \delta_{\mathrm{epi}\gamma_{C_i}}(x-y_i-z_i,t) + \sum_{i=1}^n \delta_{\Omega_i}(y_i) + \sum_{i=1}^n \delta_{L_i}(z_i) \right\},$$

while we opted for the following one

$$(P_{\mathcal{T}}) \qquad \inf_{(x,t)\in X\times\mathbb{R}} \left\{ t + \sum_{i=1}^n \delta_{\mathrm{epi}T^{C_i}_{\Omega_i,\delta_{L_i}}}(x,t) \right\}. \tag{2.22}$$

Essentially, in order to implement their parallel splitting algorithm in [11], the authors leveraged the splitting property of their method to decompose the infimal convolution in the objective function. Unfortunately, we do not have this flexibility, as the conditions required for applying the Chambolle-Pock algorithm do not permit such a choice of decomposition.
Indeed, in our previous investigations, we set

$$F(K(x,t)) = \sum_{i=1}^n \delta_{\mathrm{epi}T^{C_i}_{\Omega_i,\delta_{L_i}}}(x,t).$$

Eventually, by taking $\gamma_{C_i} = \|.\|, \Omega_i = \{p_i\}$ and $L_i = \{0_{\mathbb{R}^d}\}$, for $i = 1,..,n$, we managed to identify a scenario where the infimal convolution reduces to a translated norm. Precisely, $T^{C_i}_{\{p_i\},\delta_{\{0_{\mathbb{R}^d}\}}} = \|.\| \ \square \ \delta_{\{p_i\}} \ \square \ \delta_{\{0_{\mathbb{R}^d}\}} = \|. - p_i\|$ which yields in that case

$$F(K(x,t)) = \sum_{i=1}^n \delta_{\mathrm{epi}\|.-p_i\|}(x,t),$$

where F is a separable function and K a uniquely defined continuous linear operator. As the results given in [14] allowed us to compute the proximal operator of the function F, we were able to implement the Chambolle-Pock method described previously.
However, if we assume that, for $i = 1,\cdots,n$, Ω_i is a less restrictive set, for example, in [11, Example 4.2], the authors considered the set $\Omega_i = \left\{x \in \mathbb{R}^d : \|x-p_i\|_\infty \le a_i\right\}$, where a_i is a nonnegative real number, then the application of this method becomes compromised.
Indeed, in that instance, taking $L_i = \{0_{\mathbb{R}^d}\}$, $T^{C_i}_{\Omega_i,\delta_{L_i}} = \gamma_{C_i} \ \square \ \delta_{\Omega_i} \ \square \ \delta_{L_i} = \|.\| \ \square \ \delta_{\Omega_i}$. While the Douglas-Rachford splitting algorithm accommodates the inclusion in the objective function of the terms $\sum_{i=1}^n \delta_{\mathrm{epi}\gamma_{C_i}}(x-y_i,t) + \sum_{i=1}^n \delta_{\Omega_i}(y_i)$, the Chambolle-Pock algorithms restricts this flexibility by requiring a single linear operator, thereby prohibiting the use of such splitting properties. In that regard, we would need to handle the following terms

$$F(K(x,t)) = \sum_{i=1}^n \delta_{\mathrm{epi}\left(\|.\| \ \square \ \delta_{\Omega_i}\right)}(x,t).$$

which rewrites, as $\|.\| \ \square \ \delta_{\Omega_i} = \mathrm{d}(.,\Omega_i)$ is the distance function to Ω_i,

$$F(K(x,t)) = \sum_{i=1}^n \delta_{\mathrm{epi}(\mathrm{d}(.,\Omega_i))}(x,t). \tag{2.23}$$

Unfortunately, to our knowledge, there are currently no explicit formulae for projecting onto the epigraphs of functions defined by infimal convolutions (and in particular the sum 2.23). This is an area where further research could prove valuable in the future.

2.8.2 Perspectives related to the mirror descent algorithm

Although we identified in Example 2.6.2 an instance where it is not the case, our proceedings showcased that, in general, the Chambolle-Pock algorithm outperforms the mirror descent method in terms of efficiency and convergence speed. However, it is important to note that the Chambolle-Pock algorithm operates under stricter assumptions. The mirror descent method , on the other hand, offers the flexibility to handle cases that lie beyond the scope of the Chambolle-Pock algorithm. To illustrate this point, let us consider an instance where $\Omega_i = \left\{x \in \mathbb{R}^d : \|x - p_i\|_\infty \leq a_i\right\}, i = 1, \cdots, m$. The assumptions required for the Chambolle-Pock algorithm prevent us from addressing this case (we have no choice but to impose $a_i = 0$ in order to deal with translated norms) which can be effectively handled by the mirror descent method.

Example 2.8.1. Let $d = 2$ and $n = 7$. Consider the points $p_1 = (4, 3)^T$, $p_2 = (2, 5)^T$, $p_3 = (4, -3)^T$, $p_4 = (1, -5)^T$, $p_5 = (7, -6)^T$, $p_6 = (6, 1)^T$ and $p_7 = (3, -5)^T$ $a_1 = 1, a_2 = 2, a_3 = 3, a_4 = 0.5, a_5 = 2, a_6 = 1$ and $a_7 = 1$.
We will compare the results obtained using our mirror descent method with those achieved by both the primal and dual splittings methods.

We run our MATLAB programs for various stepsizes $\nu > 0$ as well as multiple parameters $\delta > 0$ and the best performances are illustrated in Table 2.32 - Table 2.35.

MATLAB computed,

- via the primal parallel splitting algorithm, an optimal solution illustrated in Figure 2.7a given by
$$\bar{x} = (2.7652, -0.8567)^\top, \quad \bar{t} = 3.8567,$$
while the optimal primal objective value is $v(P) = 3.8567$.

- via the mirror descent algorithm, an optimal solution illustrated in Figure 2.7b given by
$$\bar{x} = (2.7642, -0.8574)^\top, \quad \bar{t} = 3.8511,$$
while the optimal primal objective value is $v(P) = 3.8511$.

Algorithm	**Primal**	**Dual**	**Mirror descent** ($\delta = 0.6127$)	
	($\nu = 24$)	($\nu = 0.076$)	($\nu = 0.05$)	($\nu = 0.005$)
Number of iterations	292	170	542	768
CPU time (seconds)	0.0913	0.0406	0.2162	0.2121
Objective value	3.8576	3.8555	3.8028	4.1137

Table 2.32: Comparison of splitting and mirror descent methods ($d = 2, n = 7$) for $\epsilon_1 = 10^{-3}$

As demonstrated in the previous examples, unlike the splitting algorithms, the mirror descent method does not achieve the exact objective value. To approach the exact objective value more closely, the parameter ν must be reduced further. However, this adjustment leads to a significant increase in both the number of iterations and CPU time. Consequently, although mirror descent can improve the objective value, its overall efficiency is diminished compared to the primal and dual splitting methods due to the higher computational costs associated with smaller values of ν.

A small ν can lead to a more precise approximation but requires more iterations and computational time, as seen in the results while a larger ν can "expedite" convergence but at the cost

Algorithm	**Primal**	**Dual**	**Mirror descent** ($\delta = 0.6127$)	
	($\nu = 24$)	($\nu = 0.076$)	($\nu = 0.05$)	($\nu = 0.005$)
Number of iterations	399	268	926	5372
CPU time (seconds)	0.1110	0.0497	0.2508	1.1448
Objective value	3.8568	3.8555	3.8005	3.8528

Table 2.33: Comparison of splitting and mirror descent methods ($d = 2, n = 7$) for $\epsilon_2 = 10^{-4}$

Algorithm	**Primal**	**Dual**	**Mirror descent** ($\delta = 0.6127$)	
	($\nu = 24$)	($\nu = 0.076$)	($\nu = 0.05$)	($\nu = 0.005$)
Number of iterations	627	459	1691	12911
CPU time (seconds)	0.1369	0.0523	0.4832	2.9053
Objective value	3.8567	3.8567	3.8004	3.8511

Table 2.34: Comparison of splitting and mirror descent methods ($d = 2, n = 7$) for $\epsilon_3 = 10^{-6}$

Algorithm	**Primal**	**Dual**	**Mirror descent** ($\delta = 0.6127$)	
	($\nu = 24$)	($\nu = 0.076$)	($\nu = 0.05$)	($\nu = 0.005$)
Number of iterations	857	652	2456	20434
CPU time (seconds)	0.1714	0.0807	0.6811	4.8432
Objective value	3.8567	3.8567	3.8004	3.8511

Table 2.35: Comparison of splitting and mirror descent methods ($d = 2, n = 7$) for $\epsilon_4 = 10^{-8}$

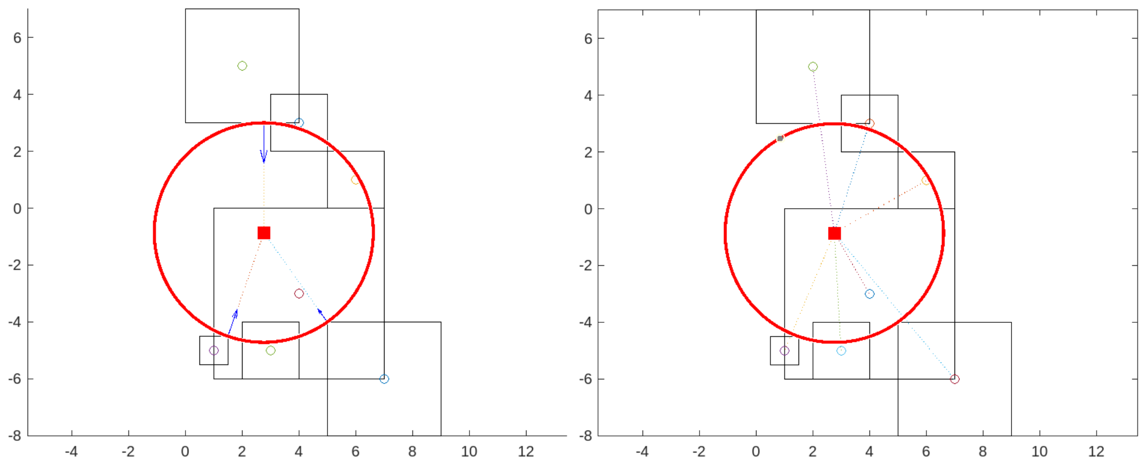
of precision which may yield a solution that does not exactly correspond to the optimum of the original objective function. Indeed, the smoothing technique introduces a bias in the objective value to gain computational efficiency. The objective value obtained from mirror descent reflects the minimum of the smoothed function, which is slightly different from the true minimum of the original problem. This explains why this method does not lead the exact objective value seen with the other methods, which do not rely on such smoothing and thus directly minimize the original function.

From our conclusions, we infer that parallel splitting algorithms could serve as a middle ground, balancing the computational efficiency of the Chambolle-Pock algorithm with the broader applicability of the mirror descent method. With this in mind, the final phase of our internship focused on adapting an inertial version of these parallel splitting methods to enhance their performance.

2.8.3 Perspectives related to the parallel splitting algorithms

We will introduce our parallel splitting algorithm which is based on [12] in the context of Special case one 2.4.1, in which case, for $i = 1, \cdots, n$, $f_i = \gamma_{G_i}$ and the generalized minimal time function $\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}$ appears in the objective.

Let X_i, for $i = 1, \cdots, m$, and Y be Hilbert spaces and let E be a closed linear subspace of Y. Assume that, for every $i = 1, \cdots, m$, $f_i \in \Gamma_0(X_i)$ and $L_i \in \mathcal{B}(Y, X_i)$ is such that $L_i(E)$ is closed.



(a) Optimal solution via parallel splitting (b) Optimal solution via mirror descent

Figure 2.7: Visualization of an optimal solution to problem $(d = 2, n = 7)$ via parallel splitting and mirror descent methods

In [12], the authors considered the problem

$$\min_{y\in E}\sum_{i=1}^{m} f_i(L_i y). \tag{P_L}$$

and proposed Algorithm 7 and proved a the convergence result given in Proposition 3.6.3.

For our numerical experiments, we will consider a particular instance of (P_L).

$$\min_{y\in X}\sum_{i=1}^{m} f_i(y), \tag{P_{Id}}$$

where, for every $i = 1, \cdots, n$, $f_i \in \Gamma_0(X)$.

In this instance, we opted for the following version of the inertial parallel splitting algorithm 7.

Algorithm 4 Inertial Parallel Splitting

```
1: Input: Choose $(t_{i,0})_{1\le i\le m} \in X$, $(p_{i,-1})_{1\le i\le m} \in X$, and constants $\omega_i > 0$, $\epsilon_i \in [0,1[$, $\lambda_n \in ]0,2[$.
   Set $y_0 = \frac{1}{m}\sum_{i=1}^{m}\omega_i t_{i,0}$.
2: for $k \ge 0$ do
3:     for $i = 1, \dots, n$ do
4:         $p_{i,n} \leftarrow \text{prox}_{(1-\epsilon_i)f_i/\omega_i}\left((1-\epsilon_i)t_{i,n} + \epsilon_i p_{i,n-1}\right) + a_{i,n}$
5:         $c_n \leftarrow \frac{1}{m}\sum_{i=1}^{m}\omega_i p_{i,n}$
6:     end for
7:     for $i = 1, \dots, n$ do
8:         $t_{i,n+1} \leftarrow t_{i,n} + \lambda_n\left(2c_n - y_n - p_{i,n}\right)$
9:     end for
10:    $y_{n+1} \leftarrow y_n + \lambda_n\left(c_n - y_n\right)$
11: end for
```

In this context, Proposition 3.6.3 reformulates as follow.

Proposition 2.8.2. *Let $\epsilon \in [0,1[$ and $\lambda \in \mathbb{R}$. Let $\{y_k\}_{k\in\mathbb{N}} \subset X$, $\{c_k\}_{k\in\mathbb{N}} \subset X$, $\{t_k\}_{k\in\mathbb{N}} \subset X$, and $\{p_k\}_{k\geq -1} \subset X$ be generated by Algorithm 3.*
Suppose that the following assumptions hold:

1. $0 \in \operatorname{sqri}\{(z - x_1, \ldots, z - x_n) \mid z \in X, \quad x_1 \in \operatorname{dom} f_1, \quad \ldots, \quad x_n \in \operatorname{dom} f_n\}$.
2. $\lambda \in]0,2[$.

If the set of solutions to (P_{Id}) is nonempty, then $(y_k)_{n\in\mathbb{N}}$, $(c_k)_{k\in\mathbb{N}}$, and $(p_{i,k})_{k\in\mathbb{N}}$ converge weakly to $\hat{y}$ where $\hat{y}$ is a solution to (P_{Id}).

In the examples proposed in the previous section, the objective function involved $\mathcal{T}^{C_i}_{\Omega_i,\delta_{L_i}}$, for $i = 1, \cdots, n$, as extended minimal time function while in this following two scenarios, the objective function incorporates $\mathcal{T}^{C_i}_{\Omega_i,\gamma_{G_i}}$, for $i = 1, \cdots, n$.

Example 2.8.3. Consider the case where $d = 2$ and $n = 10$, using the same set of points as in Example 2.6.2. In Table 2.36 - Table 2.39, we compare the performance of the inertial version of the parallel splitting algorithm with its non-inertial counterpart to evaluate potential improvements.

MATLAB computed, via the inertial parallel splitting method (as well as the non-inertial version), an optimal solution given by

$$\bar{x} = (2.8333, -1.000)^\top, \quad \bar{t} = 6.000,$$

while the optimal primal objective value is $v(P) = 6.000$.

Algorithm	**Primal** ($\nu = 6$)	**Inertial primal** ($\mu = 1, \epsilon = 0.0001$)	**Dual** ($\nu = 0.08$)	**Inertial dual** ($\mu = 1, \epsilon = 0.0001$)
Number of iterations	4667	3730	1180	3462
CPU time (seconds)	1.7894	1.3294	0.2520	0.6722
Objective value	5.9966	5.9968	5.9713	6.0234

Table 2.36: Comparison of primal and dual splittings methods with their inertial versions ($d = 2, n = 3$) for $\epsilon_1 = 10^{-3}$

Algorithm	**Primal** ($\nu = 6$)	**Inertial primal** ($\mu = 1, \epsilon = 0.0001$)	**Dual** ($\nu = 0.08$)	**Inertial dual** ($\mu = 1, \epsilon = 0.0001$)
Number of iterations	6723	5786	2716	5019
CPU time (seconds)	2.5164	2.0965	0.5432	1.1278
Objective value	6.0003	6.0003	6.0027	5.9977

Table 2.37: Comparison of primal and dual splittings methods with their inertial versions ($d = 2, n = 10$) for $\epsilon_2 = 10^{-4}$

The inertial version of the primal algorithm shows a noticeable reduction in iterations. Specifically, it consistently converges with nearly $1,000$ fewer iterations across different error tolerances. Additionally, this method demonstrates an improvement in computational time. For example, at $\epsilon_3 = 10^{-6}$, the inertial method reduces CPU time from 4.4539 to 3.8582 seconds. Although the improvement might not be huge, it is indeed significant and suggests that inertia can accelerate

Algorithm	**Primal** ($\nu = 6$)	**Inertial primal** ($\mu = 1, \epsilon = 0.0001$)	**Dual** ($\nu = 0.08$)	**Inertial dual** ($\mu = 1, \epsilon = 0.0001$)
Number of iterations	10816	9900	5828	8101
CPU time (seconds)	4.4539	3.8582	1.1569	1.8257
Objective value	6.0000	6.0000	6.0000	6.0000

Table 2.38: Comparison of primal and dual splittings methods with their inertial versions ($d = 2, n = 10$) for $\epsilon_3 = 10^{-6}$

Algorithm	**Primal** ($\nu = 6$)	**Inertial primal** ($\mu = 1, \epsilon = 0.0001$)	**Dual** ($\nu = 0.08$)	**Inertial dual** ($\mu = 1, \epsilon = 0.0001$)
Number of iterations	14929	14013	8941	11213
CPU time (seconds)	5.8706	5.8645	1.9910	2.6003
Objective value	6.0000	6.0000	6.0000	6.0000

Table 2.39: Comparison of primal and dual splittings methods with their inertial versions ($d = 2, n = 10$) for $\epsilon_4 = 10^{-8}$

convergence both in terms of iterations and computational time.
However, in the dual setting, the introduction of inertia appears to have a negative impact. Regardless of the error tolerance, the inertial dual algorithm consistently requires more iterations and CPU time to reach convergence. For instance, at the same accuracy ϵ_3, the inertial method increases CPU time from 1.1569 to 1.8257 second. Despite experimenting with various parameters and step sizes, inertia seems to slow down the convergence in the dual case. This indicates that while inertia can enhance performance in the primal algorithm, it may not provide the same benefits -and might even hinder performance- in the dual algorithm.

We will examine whether this trend is consistent in the next example which involves more given points.

Example 2.8.4.

Consider an example with $n = 20$ points. The points were generated using MATLAB's *rand* function, which produces uniformly distributed random numbers.
MATLAB computed,

- via the primal parallel splitting method, an optimal solution given by

$$\bar{x} = (0.2602, 0.4294)^\top, \quad \bar{t} = 1.9210,$$

  while the optimal primal objective value is $v(P) = 1.9210$.

- via the inertial primal parallel splitting method, an optimal solution given by

$$\bar{x} = (0.2375, 0.4294)^\top, \quad \bar{t} = 1.9210,$$

  while the optimal primal objective value is $v(P) = 1.9210$.

Algorithm	**Primal**	**Inertial primal**	**Dual**	**Inertial dual**
	($\nu = 6$)	($\mu = 1, \epsilon = 0.0001$)	($\nu = 0.08$)	($\mu = 1, \epsilon = 0.0001$)
Number of iterations	10601	7785	1955	7305
CPU time (seconds)	10.6118	7.7175	0.7852	3.5868
Objective value	1.9179	1.9242	1.9310	1.9063

Table 2.40: Comparison of primal and dual splittings methods with their inertial versions ($d = 2, n = 20$) for $\epsilon_1 = 10^{-3}$

Algorithm	**Primal**	**Inertial primal**	**Dual**	**Inertial dual**
	($\nu = 6$)	($\mu = 1, \epsilon = 0.0001$)	($\nu = 0.08$)	($\mu = 1, \epsilon = 0.0001$)
Number of iterations	17860	15129	9523	13249
CPU time (seconds)	20.6331	17.1656	4.0129	6.7343
Objective value	1.9207	1.9207	1.9203	1.9225

Table 2.41: Comparison of primal and dual splittings methods with their inertial versions ($d = 2, n = 20$) for $\epsilon_2 = 10^{-4}$

Algorithm	**Primal**	**Inertial primal**	**Dual**	**Inertial dual**
	($\nu = 6$)	($\mu = 1, \epsilon = 0.0001$)	($\nu = 0.08$)	($\mu = 1, \epsilon = 0.0001$)
Number of iterations	32504	29690	21301	24919
CPU time (seconds)	59.4297	50.3599	11.8184	20.2882
Objective value	1.9210	1.9210	1.9210	1.9210

Table 2.42: Comparison of primal and dual splittings methods with their inertial versions ($d = 2, n = 20$) for $\epsilon_3 = 10^{-6}$

Algorithm	**Primal**	**Inertial primal**	**Dual**	**Inertial dual**
	($\nu = 6$)	($\mu = 1, \epsilon = 0.0001$)	($\nu = 0.08$)	($\mu = 1, \epsilon = 0.0001$)
Number of iterations	47147	44333	33080	36698
CPU time (seconds)	116.1680	113.1483	27.8191	40.6156
Objective value	1.9210	1.9210	1.9210	1.9210

Table 2.43: Comparison of primal and dual splittings methods with their inertial versions ($d = 2, n = 20$) for $\epsilon_4 = 10^{-8}$

In this instance, the results presented in Table 2.40 - Table 2.43 confirm that introducing inertia into the primal algorithm can lead to both faster convergence and reduced computational time. Indeed, for $\epsilon_4 = 10^{-8}$, the inertial primal algorithm requires fewer iterations ($44,333$) compared to the non-inertial primal version ($47,147$) (a reduction of approximately $2,800$ iterations) while demonstrating a slight improvement in computational time, reducing it from 113.1483 seconds to 116.1680 seconds. This suggests that inertia is effectively beneficial in improving the performance of the primal splitting algorithm.
However, similarly to the previous example, regardless of the accuracy chosen, inertia negatively impacts the dual algorithm's performance. For instance, for $\epsilon_3 = 10^{-6}$ accuracy the inertial dual

parallel splitting method increases both the number of iterations (from 21301 to 24919) and CPU time (from 11.8184 to 20.2882 seconds).
The effect of inertia on splitting methods within our location problems appears to be mixed. Although the initial primal version benefits from inertia, showing improved convergence speeds, the dual version did not exhibit similar enhancements despite testing various step sizes and parameter ranges. Further investigations may prove to be useful.

Chapter 3

Entropy and portfolio optimization

Portfolio optimization is a concept in finance consisting on the selection of the best allocation of financial assets to achieve a desired return while managing risk. The objective is to maximize the expected return of the portfolio for a given level of risk or to minimize risk for a given level of expected return. In the 1950s, Harry Markowitz is the pioneer of modern portfolio optimization for which he is awarded the 1990 Nobel Price of Economy.

3.1 Some elements on modern portfolio theory

Modern portfolio theory assumes that investors are risk averse, meaning that given two portfolios that offer the same expected return, investors will prefer the less risky one. Thus, an investor will take on increased risk only if compensated by higher expected returns. Conversely, an investor who wants higher expected returns must accept more risk. The exact trade-off will be the same for all investors, but different investors will evaluate the trade-off differently based on individual risk aversion characteristics.
Two assets are considered uncorrelated if the returns of one asset do not predict the returns of the other. In statistical terms, the correlation coefficient between their returns is zero. Investing in uncorrelated assets can significantly reduce portfolio risk as the performance of one asset does not affect the performance of the other. By combining uncorrelated assets, the overall volatility of the portfolio can be reduced without necessarily sacrificing expected returns.
Two assets are negatively correlated if the returns of one asset tend to move in the opposite direction to the returns of the other. In statistical terms, the correlation coefficient between their returns is less than zero, often significantly so (close to -1). Investing in negatively correlated assets can provide a natural hedge against market volatility. When one asset loses value, the other tends to gain value, therefore offsetting the loss. Combining negatively correlated assets can lead to a portfolio with much lower overall volatility, as the opposing movements of the assets smooth out the volatility.
The modern portfolio theory is used to diversify a portfolio in order to get a better return overall without a bigger risk. Another benefit of this theory is that it can reduce volatility. The best way to do that is to choose assets that have a negative correlation. Ultimately, the goal of the modern portfolio theory is to create the most efficient portfolio possible.
A risk measure is a tool that helps us understand how risky a financial situation is. In simple terms, it does this by giving us numbers that show how likely and how severe potential losses could be.
During our investigations, inspired by the work presented in [3] we concentrated on a particular risk measure. The following section will outline general properties of risk measures and define

specifically the ones we covered in the context of portfolio optimization.

3.2 Risk measures and Portfolio Optimization

3.2.1 Risk measures and EVaR

A risk measure is a function ρ that assigns a real value to a random variable $\mathbf{X}$ in order to quantify its risk degree. To precisely define the concept of a risk measure, consider a probability space $(\Omega, \mathbf{F}, \mathbb{P})$ where Ω is a set of all simple events, $\mathbf{F}$ is a σ-algebra of subsets of Ω, and $\mathbb{P}$ is a probability measure on $\mathbf{F}$. Also, let $\mathbf{L}^0(\Omega, \mathbf{F}, \mathbb{P})$ be the set of all Borel measurable functions (random variables) $\mathbf{X} : \Omega \to \mathbb{R}$, and let $\mathbf{X} \subseteq \mathbf{L}^0$ be the model space, which is a subspace including all real numbers (constant functions). Then, a risk measure is defined as $\rho : \mathbf{L}^0(\Omega, \mathbf{F}, \mathbb{P}) \to \bar{\mathbb{R}}$.
The literature introduces several properties for a suitable risk measure. The following may be the most important properties for the risk measure ρ.

1. Translation invariance: $\rho(X + c) = \rho(X) + c$ for any $X \in \mathbf{X}$ and $c \in \mathbb{R}$.
2. Subadditivity: $\rho(X + Y) \leq \rho(X) + \rho(Y)$ for all $X, Y \in \mathbf{X}$.
3. Monotonicity: if $X \leq Y$ and $X, Y \in \mathbf{X}$, then $\rho(X) \leq \rho(Y)$.
4. Positive Homogeneity: $\rho(\lambda X) = \lambda \rho(X)$ for all $X \in \mathbf{X}$ and $\lambda \geq 0$.

A risk measure is called coherent if it satisfies properties 1-4.

In finance, X often represents a gain or profit, and thus a risk measure is called coherent if the functional $\psi(X) = \rho(-X)$ satisfies the above four properties. For definitions and comparisons of various risk measures, we refer [3]. We will focus in the subsequent sections on one specific risk measure which is the entropic value-at risk.

The entropic value-at-risk (EVaR) is a new coherent risk measure introduced in [2], which is an upper bound for both the value-at-risk (VaR) and conditional value-at-risk (CVaR). For more details regarding risk measures, we refer to [2] and [3].
The EVaR at risk level α is defined as

$$\mathrm{EVaR}_{1-\alpha}(X) := \inf_{\theta > 0} \left\{ \frac{1}{\theta} \ln \left(\frac{E\left(e^{\theta X}\right)}{\alpha} \right) \right\}, \quad \forall X \in \mathbf{L}_M, \quad \forall \alpha \in (0, 1], \tag{3.1}$$

where $\mathbf{L}_M$ stands for the set of all random variables X whose moment-generating function $E\left(e^{\theta X}\right)$ is finite for all $\theta \in \mathbb{R}$. The EVaR is strongly monotone over its domain and strictly monotone over a broad sub-domain including all continuous distributions. A key feature for a risk measure, besides its financial properties, is its applicability in large-scale sample-based portfolio optimization.

3.2.2 Portfolio Optimization with EVaR

Let $\mathbf{w} \in \mathbb{R}^n, n \geq 1$, denote a portfolio vector indicating the fraction of investment of some available budget (assumed to equal 1) in each one of the n financial instruments. Let $\mathbf{R} \in \mathbb{R}^k$ be a random vector with a known probability distribution, representing risk factors required to specify the portfolio return. The random vector $\mathbf{R}$ is modeled over the underlying probability space $(\Omega, \mathbf{F}, \mathbb{P})$. The loss of the portfolio (the negative return) is also a random variable denoted by $G(\mathbf{w}, \mathbf{R})$, which depends on the vector of risk factors $\mathbf{R}$ and portfolio vector $\mathbf{w}$. Assuming that no short positions are allowed, an optimal portfolio can be determined by solving the following optimization problem:

$$\min_{\substack{\mathbf{w}\geq 0,\\ \mathbf{w^T 1}=1,\\ \mathbf{w}\in D}} \rho(G(\mathbf{w},\mathbf{R})) \tag{$P_{\mathbf{R}}$}$$

where ρ is a risk measure, representing the risk of the portfolio $\mathbf{w}$, and D is a compact set, representing user-specified requirements.
To have a well-defined problem, for all $\mathbf{w} \in D$, it is required to assume that $G(\mathbf{w},\mathbf{R})$ is a Borel measurable function and that the objective function in $(P_{\mathbf{R}})$ is finite. The random variable $G(\mathbf{w},\mathbf{R})$ always lies in the model space considered for the risk measure ρ.
In [3], the authors proved that the objective function of $(P_{\mathbf{R}})$ is convex if the risk measure ρ is coherent and $G(.,s)$ is convex for all $s \in \operatorname{supp}(\mathbf{R})$, where $\operatorname{supp}(\mathbf{R})$ denotes the support of the random vector $\mathbf{R}$ so that $(P_{\mathbf{R}})$ is a convex optimization problem.
We denote the samples from the random vector $\mathbf{R}$ by $a^j = (a_1^j, \cdots, a_k^j)^T$, with probability $p_j, j = 1, \cdots, N$, where N is the sample size. This setting is called in [3] *sample-based setting* and will be considered in the following. In most practical cases, we have $p_j = \frac{1}{N}, j = 1, \cdots, N$.
Under this sample-based setting, if we replace in $(P_{\mathbf{R}})$ the generic risk measure ρ by the EVaR (3.1), we obtain the following optimization problem

$$\min_{\substack{\mathbf{w}\geq 0,\\ \mathbf{w^T 1}=1,\\ \mathbf{w}\in D}} \inf_{\theta>0} \left\{ \frac{1}{\theta} \ln \left(\frac{E\left(e^{\theta(G(\mathbf{w},\mathbf{R}))}\right)}{\alpha} \right) \right\}$$

or equivalently setting $t = \frac{1}{\theta}$ and rewriting the expectation

$$\min_{\substack{\mathbf{w}\geq 0, t>0,\\ \mathbf{w^T 1}=1,\\ \mathbf{w}\in D}} \left\{ t \ln \left(\sum_{j=1}^{N} p_j \left(e^{\frac{1}{t}(G(\mathbf{w},\mathbf{a^j}))} \right) \right) - t \ln \alpha \right\}. \tag{P_{EVaR}}$$

Our purpose is to solve portfolio optimization problems where the risk is modelled by EVaR via parallel splitting methods. Indeed, in [11], the authors showed that (P_{EVaR}) corresponds in fact to the Lagrange-dual problem of some generalization of a primal entropy constrained linear optimization problem treated by [7].

3.3 Portfolio optimization via splitting

Let us first recall that for any subset A of a Hilbert space X, its stands that $\delta_A^* = \sigma_A$ and $\partial\delta_A = N_A$ and precise that much of this section is based on the theoretical results in [4]. We will first formulate the general primal constrained optimization whose dual models the EVaR portfolio optimization problem. Then we will consider a particular instance which slightly generalizes the one given in [4].

3.3.1 General case

Let us consider the following entropy constrained optimization problem

$$\inf_{\substack{x\in\mathbb{R}_+^m,\quad y\in\mathbb{R}^d,\quad A_p x + By \leq h,\\ \sum_{j=1}^m p_j x_j = 1,\quad \sum_{j=1}^m p_j x_j \ln x_j \leq -H}} \left\{ c_p^T x + \sigma_D(y) \right\}, \tag{P}$$

where

$$\begin{aligned}
&x = (x_1, \cdots, x_m)^T \in \mathbb{R}^m, \quad y = (y_1, \cdots, y_d)^T \in \mathbb{R}^d \\
&0 < p_j < 1, \quad j = 1, \cdots, m, \quad c = (c_1, \cdots, c_m)^T \in \mathbb{R}^m, \quad c_p = (c_1 p_1, \cdots, c_m p_m)^T \in \mathbb{R}^m \\
&A = (a_{ij})_{i,j=1}^{n,m} \in \mathbb{R}^{n \times m}, \quad A_p = (a_{ij} p_j)_{i,j=1}^{n,m} \in \mathbb{R}^{n \times m}, \quad B = (b_{ij})_{i,j=1}^{n,d} \in \mathbb{R}^{n \times d}, \\
&h = (h_1, \cdots, h_n)^T \in \mathbb{R}^n, \quad H \in \mathbb{R},
\end{aligned} \tag{3.2}$$

are given and $D \subseteq \mathbb{R}^d$ is a nonempty closed convex set.
To guarantee continuity at the origin, we use the convention $0 \ln 0 = 0$.

Remark 33. In [4], the authors showed the following facts.

- If $H > \ln\left(\sum_{j=1}^m p_j\right)$, then the feasible set of (P) is empty.
- If $H = \ln\left(\sum_{j=1}^m p_j\right)$, the optimal solution to (P) is given by $\bar{x}_j = \ln\left(\sum_{j=1}^m p_j\right), j = 1, \cdots, m$ while (P) reduces to the convex optimization problem

$$c_p^T \bar{x} + \inf_{y \in \mathbb{R}^d, \quad A_p x + By \le h,} \{\sigma_D(y)\} \tag{3.3}$$

  which can be solved via proximal splitting using conjugate duality.
- Using *Kullback-Leibler entropy* the minimal value of $\sum_{j=1}^m p_j x_j \ln x_j$, subject to the constraints $\sum_{j=1}^m p_j x_j = 1$, and $x_j \ge 0, j = 1, \cdots, m$, is given by $\min\{\ln p_1, \cdots, \ln p_m\}$ and is attained when $x_l = \frac{1}{p_l}$ with $p_l = \min\{p_1, \cdots, p_m\}$ and $x_j = 0, j = 1, \cdots, m, j \ne l$. In that case, (P) becomes a convex optimization problem similar to the one appearing in (3.3).

To exclude trivial cases presented in the previous remark, we assume that

$$\min\{\ln p_1, \cdots, \ln p_m\} < H < \ln\left(\sum_{j=1}^m p_j\right). \tag{3.4}$$

For any $y \in \mathbb{R}^d$, the set defined by

$$S_y = \left\{ x \in \mathbb{R}_+^m, \quad A_p x + By \le h, \quad \sum_{j=1}^m p_j x_j = 1, \quad \sum_{j=1}^m p_j x_j \ln x_j \le -H \right\} \tag{3.5}$$

is closed, convex and bounded, so that by Weierstrass theorem, the infimum in (P) is attained for any $y \in \mathbb{R}^d$.

By introducing dual variables corresponding to the constraints, the Lagrange dual formulation of(P) is expressed as

$$\begin{aligned}
&\sup_{\lambda \in \mathbb{R}_+^n, \quad \gamma \in \mathbb{R}, \quad \beta \in \mathbb{R}_+} \inf_{x \in \mathbb{R}_+^m, \quad y \in \mathbb{R}^d} \Bigg\{ c_p^T x + \sigma_D(y) \\
&+ \lambda^T (A_p x + By - h) + \gamma \left(\sum_{j=1}^m p_j x_j - 1\right) + \beta \left(\sum_{j=1}^m p_j x_j \ln x_j + H\right) \Bigg\}
\end{aligned} \tag{D_L}$$

Remark 34. In order to present a simplified formulation of (D_L), we use the following observations made in [4].

- $\inf_{y\in\mathbb{R}^d}\left\{\lambda^T By + \sigma_D(y)\right\} = -\sigma_D^*(-B^T\lambda) = -\delta_D(-B^T\lambda)$.
- Let $v > 0, w \in \mathbb{R}$. Considering the function $\phi : \mathbb{R}_+ \to \mathbb{R}$ defined by $\phi(x) := vx\ln x + wx$, ϕ attains its minimum at $x = e^{-\frac{v+w}{v}}$ and the associated minimal value is $\phi(e^{-\frac{v+w}{v}}) = -ve^{-\frac{v+w}{v}}$.
- Let $w, \beta > 0, k \in \mathbb{R}$. Considering the function $\psi : \mathbb{R} \to \mathbb{R}$ defined by $\psi(\gamma) := -\beta we^{-\frac{\gamma}{\beta}} - \gamma + k$, ψ attains its maximum at $\gamma = \beta\ln w$ and the associated maximal value is $\phi(\beta\ln w) = k - \beta\ln(ew)$.

Taking into account the abovementioned arguments, the dual problem (D_L) associated to the primal problem (P) rewrites

$$\sup_{\substack{\beta>0, \quad \lambda\geq 0\\ -B^T\lambda\in D}} \left\{ -\beta\ln\left(\sum_{j=1}^m p_j \exp^{-\frac{\lambda^T A_j + c_j}{\beta}}\right) + \beta H \right\}. \qquad (D'_L)$$

Remark 35. We observe similarities between the formulations (D'_L) and (P_{EVaR}). Indeed, it suffices to let $H = \ln\alpha$ and to specify an appropriate matrix B (D'_L) while choosing accordingly $G(\mathbf{w}, \mathbf{R})$ in (P_{EVaR}).

Weak duality always holds between the primal problem (P) and its dual counterpart (D'_L). Indeed, the optimal objective value of the primal problem is always greater than or equal to the optimal objective value of the dual problem, $v(P)\geq v(D'_L)$. To attain strong duality -in which case the previous inequality becomes an equality- a necessary condition is to have a constraint qualification -which is a refinement of Slater's condition- fulfilled. We give the result as stated in [4].

Theorem 3.3.1. *Assume that the constrained qualification*

$$\exists(x,y)\in \operatorname{ri}\left(\mathbb{R}_+^m \times \operatorname{dom}\sigma_D\right) : \quad A_p x + By \leq h, \quad \sum_{j=1}^m p_j x_j = 1, \quad \sum_{j=1}^m p_j x_j \ln x_j < -H, \qquad (3.6)$$

is fulfilled, then strong duality holds between the (P) and (D'_L), $v(P)= v(D'_L)$ and the dual problem has an optimal solution $(\bar\beta, \bar\lambda) \in \mathbb{R}_+ \times \mathbb{R}_+^n$.

The following result states the necessary and sufficient optimality conditions given in [4].

Theorem 3.3.2. *1. Asssume that the constraint qualification (3.6) is fulfilled and let $(\bar x, \bar y) \in \mathbb{R}_+^m \times \mathbb{R}^d$ be an optimal solution to (P). Then there exists $(\bar\beta, \bar\lambda) \in \mathbb{R}_+ \times \mathbb{R}_+^n$, an optimal solution to (D'_L), such that*

$$c_p^\top \bar x + \sigma_D(\bar y) = -\bar\beta\ln\left(\sum_{j=1}^m p_j e^{-\frac{\bar\lambda^\top A_j + c_j}{\beta}}\right) + \bar\beta H - \bar\lambda^\top h$$

$$= \min_{\substack{\bar x\in\mathbb{R}_+^m,\\ \bar y\in\mathbb{R}^n}} \left\{ c_p^\top \bar x + \sigma_D(\bar y) + \bar\lambda^\top(A_p\bar x + B\bar y - h) + \gamma\left(\sum_{j=1}^m p_j\bar x_j - 1\right) + \bar\beta\left(\sum_{j=1}^m p_j \bar x_j\ln\bar x_j + H\right)\right\}, \qquad (i)$$

$$\sigma_D(\bar{y}) = -\bar{\lambda}^\top B\bar{y}, \tag{ii}$$

$$\bar{\lambda}^\top(A_p\bar{x} + B\bar{y} - h) = 0, \tag{iii}$$

$$\sum_{j=1}^{m} p_j\bar{x}_j \ln\bar{x}_j = -H, \tag{iv}$$

$$\bar{\beta}\sum_{j=1}^{m} p_j\bar{x}_j \ln\bar{x}_j + \bar{\beta}\ln\left(\sum_{j=1}^{m} p_j e^{-\frac{\bar{\lambda}^\top A_j + c_j}{\bar{\beta}}}\right) = -c_p^\top\bar{x} - \bar{\lambda}^\top A_p\bar{x}, \tag{v}$$

$$\bar{\lambda}_i \geq 0,\ i = 1,\ldots,n, \quad \bar{\beta} > 0, \quad -B^\top\bar{\lambda} \in D, \quad \sum_{j=1}^{m} p_j\bar{x}_j = 1. \tag{vi}$$

2. *Conversely, if there exists $(\bar{x},\bar{y}) \in \mathbb{R}^m_+ \times \mathbb{R}^d$ such that for some $(\bar{\beta},\bar{\lambda}) \in \mathbb{R}_+ \times \mathbb{R}^n_+$ the condtions (i)-(vi) are fulfilled, then $(\bar{x},\bar{y})$ is an optimal solution to (P), $(\bar{\beta},\bar{\lambda})$ is an optimal solution to (D'_L) and v(P)= v(D'_L).*

Building on the authors' work on their specific case, we propose a method for the general case that exploits optimality conditions to recover an optimal dual solution from an optimal primal solution, and vice versa. We will then adapt it to our special case.

Proposition 3.3.3. 1. *Asssume that the constraint qualification (3.6) is fulfilled and let $(\bar{x},\bar{y}) \in \mathbb{R}^m_+ \times \mathbb{R}^d$ be an optimal solution to (P). Then, an optimal solution $(\bar{\beta},\bar{\lambda})$ to (D'_L) is solution to the constrained linear system*

$$\begin{cases} \sum_{i=1}^n \bar{\lambda}_i\,(a_{ij} - h_i) + \bar{\beta}\,(\ln\bar{x}_j + H) = v(P) - c_j, \quad j = 1,\cdots,m \\ -\bar{\lambda}^T B \in \partial\sigma_D(\bar{y}), \\ \lambda \in D \cap \mathbb{R}^n_+ \\ \beta > 0. \end{cases} \tag{S_1}$$

2. *Conversely, if $(\bar{\beta},\bar{\lambda})$ is an optimal solution to (D'_L), then an optimal solution $(\bar{x},\bar{y})$ to(P) can be determined by solving the constrained linear system*

$$\begin{cases} \bar{x}_j = \exp\left(\frac{1}{\bar{\beta}}\left(v(D) - c_j - \sum_{i=1}^n \bar{\lambda}_i\,(a_{ij} - h_i)\right) - H\right), \quad j = 1,\cdots,m \\ \sum_{i=1}^m p_j c_j x_j + \sum_{i=1}^n \left(\sum_{i=1}^m \bar{\lambda}_i p_j a_{ij} x_j - h_i\right) = v(D), \\ \bar{y} \in N_D(-\bar{\lambda}^T B). \end{cases} \tag{S_2}$$

Proof. Suppose that $(\bar{x},\bar{y})$ is an optimal solution to (P). Using the second equality of optimality condition (i) in the previous theorem, we obtain the following optimality conditions.

$$\begin{cases} -\bar{\lambda}^T B \in \partial\sigma_D(\bar{y}), \\ p_j c_j + \sum_{i=1}^n \bar{\lambda}_i p_j a_{ij} + \bar{\gamma} p_j + \bar{\beta} p_j\,(\ln\bar{x}_j + 1) = 0, \quad j = 1,\cdots,m. \end{cases} \tag{3.7}$$

Multiplying the equality in (3.7) by $\bar{x}_j$ and taking the sum over $j = 1,\cdots,m$, we get

$$\sum_{i=1}^{m} p_j c_j\bar{x}_j + \sum_{i=1}^{m}\sum_{i=1}^{n} \bar{\lambda}_i p_j a_{ij}\bar{x}_j + \bar{\gamma}\sum_{i=1}^{m} p_j\bar{x}_j + \bar{\beta}\sum_{i=1}^{m} p_j\bar{x}_j\,(\ln\bar{x}_j + 1) = 0. \tag{3.8}$$

From feasibility and in particular as $\sum_{j=1}^{m} p_j \bar{x_j} = 1$, equation (3.8) rewrites

$$\sum_{i=1}^{m} p_j c_j \bar{x_j} + \sum_{i=1}^{m} \sum_{i=1}^{n} \bar{\lambda}_i p_j a_{ij} \bar{x_j} + \bar{\gamma} + \bar{\beta} \sum_{i=1}^{m} p_j \bar{x}_j \left(\ln \bar{x}_j + 1\right) = 0$$

which yields by optimality condition (iv)

$$\sum_{i=1}^{m} p_j c_j \bar{x_j} + \sum_{i=1}^{m} \sum_{i=1}^{n} \bar{\lambda}_i p_j a_{ij} \bar{x_j} + \bar{\gamma} - \bar{\beta} \left(H - 1\right) = 0. \tag{3.9}$$

From optimality condition (ii), we deduce that

$$v(P) = c_p^\top \bar{x} - \bar{\lambda}^\top B \bar{y} = c_p^\top \bar{x} + \bar{\lambda}^\top (A_p \bar{x} - h) = \sum_{i=1}^{m} p_j c_j \bar{x_j} + \sum_{i=1}^{n} \bar{\lambda}_i \left(\sum_{i=1}^{m} p_j a_{ij} \bar{x_j} - h_i \right).$$

so that equation (3.9) rewrites

$$v(P) + \bar{\lambda}^\top h + \bar{\gamma} - \bar{\beta} \left(H - 1\right) = 0.$$

leading to

$$\bar{\gamma} = -v(P) - \bar{\lambda}^\top h - \bar{\beta} \left(1 - H\right). \tag{3.10}$$

Combining expression (3.10) with optimality condition (ii), we deduce that

$$p_j c_j + \sum_{i=1}^{n} \bar{\lambda}_i p_j a_{ij} + p_j \left(-v(P) - \sum_{i=1}^{n} \bar{\lambda}_i h_i - \bar{\beta} \left(1 - H\right) \right) + \bar{\beta} p_j \left(\ln \bar{x}_j + 1\right) = 0, \quad j = 1, \cdots, m$$

or equivalently

$$\sum_{i=1}^{n} \bar{\lambda}_i \left(a_{ij} - h_i\right) + \bar{\beta} \left(\ln \bar{x}_j + H\right) = v(P) - c_j, \quad j = 1, \cdots, m. \tag{3.11}$$

Finally, by the first condition in (3.7) and (3.11), an optimal solution $(\bar{\beta}, \bar{\lambda})$ to (D'_L) is solution to the linear system

$$\sum_{i=1}^{n} \bar{\lambda}_i \left(a_{ij} - h_i\right) + \bar{\beta} \left(\ln \bar{x}_j + H\right) = v(P) - c_j, \quad j = 1, \cdots, m$$

supplemented by the constraints

$$-\bar{\lambda}^T B \in \partial \sigma_D(\bar{y}), \quad \lambda \in D \cap \mathbb{R}^n_+ \quad \beta > 0.$$

Conversely, if $(\bar{\beta}, \bar{\lambda})$ is an optimal solution to (D'_L), then, using optimality conditions (i)-(vi), an optimal solution $(\bar{x}, \bar{y})$ to (P) can be determined. Indeed, (3.11) yields

$$\ln \bar{x}_j = \frac{1}{\bar{\beta}} \left(v(P) - c_j - \sum_{i=1}^{n} \bar{\lambda}_i \left(a_{ij} - h_i\right) \right) - H, \quad j = 1, \cdots, m. \tag{3.12}$$

We deduce from strong duality $v(P)= v(D'_L)$ and (3.12) that $\bar{x}$ satisfies

$$\bar{x}_j = \exp \left(\frac{1}{\bar{\beta}} \left(v(D) - c_j - \sum_{i=1}^{n} \bar{\lambda}_i \left(a_{ij} - h_i\right) \right) - H \right), \quad j = 1, \cdots, m \tag{3.13}$$

where the dual objective value is

$$v(D) = \sum_{i=1}^{m} p_j c_j \bar{x}_j + \sigma_D(\bar{y}). \tag{3.14}$$

Using optimality conditions (*ii*)-(*iii*), one has

$$\sigma_D(\bar{y}) = -\bar{\lambda}^\top B\bar{y} = \bar{\lambda}^\top (A_p\bar{x} - h). \tag{3.15}$$

From (3.12)-(3.15), we deduce that $\bar{x}$ can be found by solving the linear system

$$\begin{cases} \bar{x}_j = \exp\left(\frac{1}{\beta}\left(v(D) - c_j - \sum_{i=1}^n \bar{\lambda}_i\left(a_{ij} - h_i\right)\right) - H\right), \quad j = 1, \cdots, m \\ \sum_{i=1}^m p_j c_j x_j + \sum_{i=1}^n \bar{\lambda}_i\left(\sum_{i=1}^m p_j a_{ij} x_j - h_i\right) = v(D). \end{cases}$$

Finally, using $-\bar{\lambda}^T B \in \partial\sigma_D(\bar{y})$ and Theorem 1.1.13, $\bar{y}$ can be determined by

$$\bar{y} \in \partial\delta_D(-\bar{\lambda}^T B) = N_D(-\bar{\lambda}^T B).$$

□

Remark 36. Next, we will demonstrate that the optimality condition involving the matrix B and the subdifferential of the support function of D simplifies when an appropriately chosen matrix B is used in our specific case. This choice will be crucial for presenting our numerical applications.

3.3.2 Special case

Let us consider a particular instance of our general entropy optimization problem (P) by taking

$$d = n+1, \quad D = \tilde{D} \times \{1\} \in \mathbb{R}^{n+1} \text{ with } \tilde{D} \subseteq \mathbb{R}^n \text{ and } B = -\begin{pmatrix} & 1 \\ \tilde{B} & \vdots \\ & 1 \end{pmatrix} \in \mathbb{R}^{n \times n+1} \text{ with } \tilde{B} \in \mathbb{R}^{n\times n}.$$

Remark 37. While, in [4], the authors imposed $c = 0$, $h = 0$ and $\tilde{B} = \mathrm{Id}$, we consider, in the following, general vectors h and c along with a generic matrix $\tilde{B}$. For the numerical tests and to simplify the resolution -especially the optimality conditions-, we will later set $\tilde{B} = \mathrm{Id}$.

Letting $y = (\tilde{y}, y_{n+1})$, the primal entropy constrained optimization problem (P) transforms into

$$\inf_{\substack{x\in\mathbb{R}^m_+, \quad \tilde{y}\in\mathbb{R}^n, \quad y_{n+1}\in\mathbb{R}, \\ \sum_{j=1}^m p_j a_{ij} x_j - \sum_{j=1}^n b_{ij} y_i - h_i \le y_{n+1}, \quad i=1,\cdots,n, \\ \sum_{j=1}^m p_j x_j = 1, \quad \sum_{j=1}^m p_j x_j \ln x_j \le -H}} \left\{ c_p^T x + \sigma_{\tilde{D}\times\{1\}}(\tilde{y}, y_{n+1}) \right\} \tag{3.16}$$

which rewrites, setting $t := y_{n+1}$ and $y := \tilde{y}$

$$\inf_{\substack{x\in\mathbb{R}^m_+, \quad y\in\mathbb{R}^n, \quad t\in\mathbb{R}, \\ \sum_{j=1}^m p_j a_{ij} x_j - \sum_{j=1}^n b_{ij} y_i - h_i \le t, \quad i=1,\cdots,n, \\ \sum_{j=1}^m p_j x_j = 1, \quad \sum_{j=1}^m p_j x_j \ln x_j \le -H}} \left\{ c_p^T x + t + \sigma_{\tilde{D}}(y) \right\}. \tag{P_s}$$

Remark 38. A connection to the previous chapter can be established as (P_s) can be seen as a minimax problem

$$\inf_{\substack{x\in\mathbb{R}^m_+, \quad y\in\mathbb{R}^n, \\ \sum_{j=1}^m p_j x_j = 1, \quad \sum_{j=1}^m p_j x_j \ln x_j \le -H}} \left\{ \max_{1\le i\le n} \left\{ \sum_{j=1}^m p_j a_{ij} x_j - \sum_{j=1}^n b_{ij} y_i - h_i \right\} + c_p^T x + \sigma_{\tilde{D}}(y) \right\}.$$

The Lagrange dual problem of (P_s) is given by

$$\sup_{\lambda\in\mathbb{R}^n_+,\quad \gamma\in\mathbb{R},\quad \beta\in\mathbb{R}_+} \inf_{x\in\mathbb{R}^m_+,\quad y\in\mathbb{R}^n, t\in\mathbb{R}} \Bigg\{ c_p^T x + t + \sigma_D(y) + \sum_{i=1}^n \lambda_i \left(\sum_{j=1}^m p_j a_{ij} x_j - \sum_{j=1}^n b_{ij} y_i - h_i - t \right) + \gamma \left(\sum_{j=1}^m p_j x_j - 1 \right) + \beta \left(\sum_{j=1}^m p_j x_j \ln x_j + H \right) \Bigg\}, \tag{D_s}$$

As $D = \tilde{D} \times \{1\}$, we observe that the constraint $-B^T\lambda \in D$ of (D_L') is reformulated in our specific instance as two constraints

$$\tilde{B}^T\lambda \in \tilde{D}, \quad \sum_{i=1}^n \lambda_i = 1.$$

Therefore, the dual problem of (P_s) is given by

$$\sup_{\substack{\beta>0,\quad \lambda_i\geq 0,\quad i=1,\cdots,n \\ \tilde{B}^T\lambda\in\tilde{D},\quad \sum_{i=1}^n \lambda_i = 1}} \left\{ -\beta \ln \left(\sum_{j=1}^m p_j \exp^{-\frac{\lambda^T A_j + c_j}{\beta}} \right) + \beta H - \lambda^T h \right\}. \tag{D_s'}$$

Remark 39. We can see the importance of imposing the appropriate constraint involving the dual variable λ. Specifically, choosing an apppropriate matrix B yields the constraint $\sum_{i=1}^n \lambda_i = 1$ which ensures that the asset allocation of the financial instruments equals the total budget, which is assumed to be normalized. This demonstrates that the dual problem (D_s') is in fact modeling our EVaR optimization problem.

Weak duality always holds. To have strong duality, we suppose the fulfillement of a qualification condition.

Theorem 3.3.4. *Assume the constrained qualification*

$$\exists (x,y,t) \in \operatorname{ri}\left(\mathbb{R}^m_+ \times \operatorname{dom}\sigma_{\tilde{D}} \times \mathbb{R}\right): \quad A_p x - \tilde{B} y - t \leq h, \quad \sum_{j=1}^m p_j x_j = 1, \quad \sum_{j=1}^m p_j x_j \ln x_j < -H, \tag{3.17}$$

is fulfilled, then strong duality holds between (P_s) and (D_s'), $v(P_s)= v(D_s')$ and the dual problem (D_s') has an optimal solution $(\bar{\beta}, \bar{\lambda}) \in \mathbb{R}_+ \times \mathbb{R}^n_+$.

Proof. It is a direct consequence of Theorem 3.3.1. It suffices to observe that in this case, $\operatorname{dom}\sigma_D = \operatorname{dom}\sigma_{\tilde{D}\times\{1\}} = \operatorname{dom}\sigma_{\tilde{D}} \times \mathbb{R}$ and $A_p x + By - h = \sum_{j=1}^m p_j a_{ij} x_j - \sum_{j=1}^n b_{ij} y_i - t$. □

The following result states the necessary and sufficient optimality conditions in the context our our specific case.

Theorem 3.3.5. *1. Assume that the constraint qualification (3.17) is fulfilled and let $(\bar{x}, \bar{y}, \bar{t}) \in \mathbb{R}^m_+ \times \mathbb{R}^n \times \mathbb{R}$ be an optimal solution to (P_s). Then there exists $(\bar{\beta}, \bar{\lambda}) \in \mathbb{R}_+ \times \mathbb{R}^n_+$, an optimal solution to (D_s') such that*

$$c_p^T x + \bar{t} + \sigma_D(\bar{y}) = -\bar{\beta} \ln \left(\sum_{j=1}^m p_j e^{-\frac{\bar{\lambda}^\top A_j + c_j}{\bar{\beta}}} \right) + \bar{\beta} H - \lambda^T h$$

$$= \min_{\substack{\bar{x}\in\mathbb{R}^m_+, \\ \bar{y}\in\mathbb{R}^d, \\ \bar{t}\in\mathbb{R}}} \left\{ c_p^T \bar{x} + \bar{t} + \sigma_{\tilde{D}}(\bar{y}) + \bar{\lambda}^\top \left(A_p \bar{x} - \tilde{B} \bar{y} - t - h \right) + \gamma \left(\sum_{j=1}^m p_j \bar{x}_j - 1 \right) + \bar{\beta} \left(\sum_{j=1}^m p_j \bar{x}_j \ln \bar{x}_j + H \right) \right\}, \tag{i}$$

$$\sigma_{\tilde{D}}(\bar{y}) = \bar{\lambda}^\top \tilde{B}\bar{y}, \tag{ii}$$

$$\bar{\lambda}^\top \left(A_p x + \tilde{B}\bar{y} - t - h\right) = 0, \tag{iii}$$

$$\sum_{j=1}^m p_j \bar{x}_j \ln \bar{x}_j = -H, \tag{iv}$$

$$\bar{\beta}\sum_{j=1}^m p_j \bar{x}_j \ln \bar{x}_j + \bar{\beta}\ln\left(\sum_{j=1}^m p_j e^{-\frac{\bar{\lambda}^\top A_j + c_j}{\bar{\beta}}}\right) = -c_p^T x - \bar{\lambda}^\top A_p \bar{x}, \tag{v}$$

$$\bar{\lambda}_i \geq 0,\ i = 1,\dots,n, \quad \bar{\beta} > 0, \quad \tilde{B}^T\bar{\lambda} \in \tilde{D}, \quad \sum_{i=1}^n \bar{\lambda}_i = 1, \quad \sum_{j=1}^m p_j \bar{x}_j = 1, \tag{vi}$$

$$\bar{t} = \max_{1\leq i\leq n}\left\{\sum_{j=1}^m p_j a_{ij}\bar{x}_j - \sum_{j=1}^n b_{ij}\bar{y}_i - h_i\right\}. \tag{vii}$$

2. *Conversely, if there exists $(\bar{x},\bar{y},\bar{t}) \in \mathbb{R}_+^m \times \mathbb{R}^n \times \mathbb{R}$ such that for some $(\bar{\beta},\bar{\lambda}) \in \mathbb{R}_+ \times \mathbb{R}_+^n$ the conditions (i)-(vii) are fulfilled, then $(\bar{x},\bar{y},\bar{t})$ is an optimal solution to (P_s), $(\bar{\beta},\bar{\lambda})$ is an optimal solution to (D_s') and $v(P_s)$= $v(P_s)$.*

Proof. It is a direct consequence of Theorem 3.3.2. Optimality conditions (i)-(vi) are straightforward. The condition (vii) yields from condition (vi) and the feasiblilty constraint related to the dual variable λ. Indeed, we can write

$$\begin{aligned}
&\sum_{i=1}^n \bar{\lambda}_i\left(\sum_{j=1}^m p_j a_{ij}\bar{x}_j - \sum_{j=1}^n b_{ij}\bar{y}_i - h_i\right) = \bar{t}\sum_{i=1}^n \bar{\lambda}_i \\
=&\bar{t} \geq \max_{1\leq i\leq n}\left\{\sum_{j=1}^m p_j a_{ij}\bar{x}_j - \sum_{j=1}^n b_{ij}\bar{y}_i - h_i\right\} \\
=&\sum_{i=1}^n \bar{\lambda}_i \max_{1\leq i\leq n}\left\{\sum_{j=1}^m p_j a_{ij}\bar{x}_j - \sum_{j=1}^n b_{ij}\bar{y}_i - h_i\right\} \geq \sum_{i=1}^n \bar{\lambda}_i\left(\sum_{j=1}^m p_j a_{ij}\bar{x}_j - \sum_{j=1}^n b_{ij}\bar{y}_i - h_i\right).
\end{aligned} \tag{3.18}$$

We deduce that inequalities in (3.18) are in fact equalities which yields optimality condition (vii). □

We propose now a characterization of the optimal dual solution from the optimal primal solution and vice-versa which is analogous to Proposition 3.3.3.

Proposition 3.3.6. *1. Assume that the constraint qualification (3.17) is fulfilled and let $(\bar{x},\bar{y},\bar{t}) \in \mathbb{R}_+^m \times \mathbb{R}^n \times \mathbb{R}$ be an optimal solution to (P_s). Then, an optimal solution $(\bar{\beta},\bar{\lambda})$ to (D_s') is solution to the constrained linear system*

$$\left\{\begin{array}{l}
\sum_{i=1}^n \bar{\lambda}_i\left(a_{ij} - h_i\right) + \bar{\beta}\left(\ln\bar{x}_j + H\right) = v(P) - c_j, \quad j = 1,\cdots,m \\
\sum_{i=1}^n \bar{\lambda}_i = 1, \quad \tilde{B}^T\bar{\lambda} \in \partial\sigma_{\tilde{D}}(\bar{y}), \\
\lambda \in \tilde{D} \cap \mathbb{R}_+^n \\
\beta > 0.
\end{array}\right. \tag{S_3}$$

2. *Conversely, if $(\bar{\beta}, \bar{\lambda})$ is an optimal solution to (D_s'), then, using the optimality conditions, an optimal solution $(\bar{x}, \bar{y}, \bar{t}) \in \mathbb{R}_+^m \times \mathbb{R}^n \times \mathbb{R}$ to (P_s) can be determined by solving the constrained linear system*

$$\left\{\begin{array}{ll} \bar{x}_j = \exp\left(\frac{1}{\bar{\beta}}\left(v(D) - c_j - \sum_{i=1}^n \bar{\lambda}_i\left(a_{ij} - h_i\right)\right) - H\right), & j = 1, \cdots, m \\ \sum_{i=1}^m p_j c_j x_j + \sum_{i=1}^n \bar{\lambda}_i \left(\sum_{j=1}^m p_j a_{ij} \bar{x}_j - h_i\right) = v(D), & \\ \bar{y} \in N_{\tilde{D}}(\tilde{B}^T \bar{\lambda}), & \\ \bar{t} = \sum_{i=1}^n \bar{\lambda}_i \left(\sum_{j=1}^m p_j a_{ij} \bar{x}_j - \sum_{j=1}^n b_{ij} \bar{y}_i - h_i\right). & \end{array}\right. \qquad (S_4)$$

Before delving into the numerical analysis, let us review the formulations of the problems considered in [3] and in [4].

- In [3], the authors considered the problem

$$\min_{\substack{\mathbf{w} \geq 0, t > 0, \\ \mathbf{w^T 1} = 1, \\ \mathbf{w} \in D}} \left\{ t \ln \left(\sum_{j=1}^N p_j \left(e^{\frac{1}{t}(G(\mathbf{w}, \mathbf{a^j}))} \right) \right) - t \ln \alpha \right\},$$

  and solved it by a primal-dual interior point method

- In [4], the authors considered the problem

$$\sup_{\substack{\beta > 0, \quad \lambda_i \geq 0, \quad i = 1, \cdots, n \\ \tilde{B}^T \lambda \in \tilde{D}, \quad \sum_{i=1}^n \lambda_i = 1}} \left\{ -\beta \ln \left(\sum_{j=1}^m p_j \exp^{-\frac{\lambda^T A_j + c_j}{\beta}} \right) + \beta H - \lambda^T h \right\}.$$

  and solved it by a parallel splitting algorithm.

In [3], the authors considered the linear case given by $G(\mathbf{w}, \mathbf{a^j}) = -(\mathbf{a^j})^T \mathbf{w}$. In order to treat our case (which is slightly more general), let us set $G(\mathbf{w}, \mathbf{a^j}) = -(\mathbf{a^j})^T \mathbf{w} + h^T \mathbf{w} + \mathbf{c^j}$ where $h \in \mathbb{R}^n$ is given and corresponds to the right hand side term in the linear constraint of the primal entropy constrained optimization problem while $\mathbf{c^j}$ is a random value for $j = 1, \cdots, N$ and stands for the vector $c \in \mathbb{R}^m$ that appears in the objective function of (P_s). Note that, in this case, the function $G(., \mathbf{a^j})$ is still convex in $\mathbf{w}$ so that the problem remains convex.

Remark 40. We should precise that assimilating the two problems, we remark that regarding the dimensions, $m = N$ and $n = k$ and regarding the variables $\beta = t$ and $\lambda = \mathbf{w}$. We will see later that this observation has numerical ramifications. We choose to adopt -temporarily- the notations presented in [4].
It is worth mentioning that we imposed that the number of risk factors equals the number of financial instruments ($k = n$). This setup -which simplifies the modeling- will enable us to achieve the application of a suitable splitting method.

We rewrite the primal problem (P_s) by means of indicator functions (encoding the constraints) in order to get an unconstrained optimization problem. For that purpose, we introduce additional variables $\tau_j, j = 1, \cdots, m$ which handle the entropy constraints.
Indeed, the constraint

$$\sum_{j=1}^m p_j x_j \ln x_j \leq -H$$

will transform into the $m+1$ constraints

$$p_j x_j \ln x_j \leq \tau_j, \quad j = 1, \cdots, m, \quad \sum_{j=1}^{m} \tau_j \leq -H.$$

(P_s) rewrites then

$$\inf_{\substack{x \in \mathbb{R}^m, \quad \tau \in \mathbb{R}^m, \\ y \in \mathbb{R}^n, \quad t \in \mathbb{R},}} \left\{ c_p^T x + t + \sigma_{\tilde{D}}(y) + \delta_{C_1}(x, y, t) + \delta_{C_2}(x) + \delta_{C_3}(\tau) + \sum_{j=1}^{m} \delta_{\operatorname{epi} h_j}(x_j, \tau_j) \right\}, \qquad (3.19)$$

where we have defined the sets

$$C_1 = \left\{ (x, y, t) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} : A_p x + \tilde{B} y - t \leq h \right\},$$
$$C_2 = \left\{ x \in \mathbb{R}_+^m : \sum_{j=1}^{m} p_j x_j = 1 \right\},$$
$$C_3 = \left\{ \tau \in \mathbb{R}^m : \sum_{j=1}^{m} \tau_j \leq -H \right\},$$

and we introduced the function

$$h_j : \mathbb{R} \to \mathbb{R}, \quad h_j(x) = \begin{cases} p_j x \ln x, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ +\infty, & \text{if } x < 0, \end{cases} \quad \text{for } j = 1, \ldots, m.$$

In order to employ a proximal splitting method for solving (P_s), we need closed formulae for the projection operators onto the sets C_1, C_2, and C_3, as well as projection operators onto the epigraphs of the functions h_j, $j = 1, \ldots, m$. To this end, we will use Theorem 3.6.6, Corollary 3.6.7, Proposition 3.6.8 and Proposition 3.6.9 which are formulated in the appendix.

3.4 Numerical Experiments

In this section, we will compare the results obtained by the Douglas Rachford parallel splitting algorithm provided in 3.6.2 and the primal dual interior point algorithm used in [3, Section 4.2]. We assess the efficiency of our proposed parallel splitting algorithm in the framework of our entropy portfolio optimization problem.

Remark 41. It is important to note that the objective value of the optimization problem (P_{EVaR}) when solved using the primal-dual interior point (PDIP) method, is equal to the negative of the objective value of the dual problem (D'_s), solved using the parallel splitting (PS) algorithm. To clarify these results, we provide in the following tables the opposite of the objective values of (P_{EVaR}) obtained by the PDIP method and the dual objective values of (D'_s) obtained from the PS method.

As in the previous chapter, we choose as stopping criterion the values $\epsilon_1 = 10^{-3}$, $\epsilon_2 = 10^{-4}$, $\epsilon_3 = 10^{-6}$ and $\epsilon_4 = 10^{-8}$ where for $i = 1, 2, 3, 4$, ϵ_i stands for the maximum bounds between the solutions given by two consecutive iterations. The origin was taken as the starting point.

We will present our investigations in finitely dimensional spaces. Specifically, we set $X = \mathbb{R}^d$.

Algorithm	**PS**$(\mu = 1.65, \nu = 1.4)$	**PDIP**
Number of iterations	646	351
CPU time (seconds)	1.0744	3.580059
Objective value	2.299084	2.296731

Table 3.1: Comparison of splitting and primal-dual interior point methods ($n = 10, m = 10$) for $\epsilon_1 = 10^{-3}$

Algorithm	**PS**$(\mu = 1.65, \nu = 1.4)$	**PDIP**
Number of iterations	924	352
CPU time (seconds)	1.3984	3.785709
Objective value	2.297589	2.296807

Table 3.2: Comparison of splitting and primal-dual interior point methods ($n = 10, m = 10$) for $\epsilon_2 = 10^{-4}$

Algorithm	**PS**$(\mu = 1.65, \nu = 1.4)$	**PDIP**
Number of iterations	1369	354
CPU time (seconds)	2.2137	4.394282
Objective value	2.296820	2.296826

Table 3.3: Comparison of splitting and primal-dual interior point methods ($n = 10, m = 10$) for $\epsilon_3 = 10^{-6}$

Algorithm	**PS**$(\mu = 1.65, \nu = 1.4)$	**PDIP**
Number of iterations	1939	357
CPU time (seconds)	2.9496	4.421544
Objective value	2.296827	2.296827

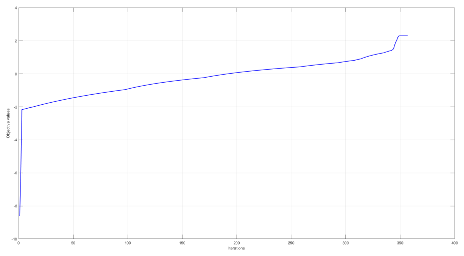
Table 3.4: Comparison of splitting and primal-dual interior point methods ($n = 10, m = 10$) for $\epsilon_4 = 10^{-8}$

Example 3.4.1. We first consider an example where $n = m = 10$.
We run our MATLAB programs for various stepsizes ν and parameters μ. The best performances of our tests are illustrated in Table 3.1 - Table 3.4.

MATLAB computed, via parallel splitting, an optimal solution given by

$$\begin{aligned}
\bar{x} &= (0.9463, 0.9369, 1.3343, 3.3603, 0.0500, 0.8699, 0.0675, 0.5003, 0.1664, 1.7321)^\top,\\
\bar{\tau} &= (-0.0011, -0.0099, 0.0415, 0.4953, -0.0286, -0.0029, -0.0155, -0.0389, -0.0227, 0.0937)^\top,\\
\bar{t} &= 4.7414.
\end{aligned}$$

We can deduce the value for y considering the choice of the set D. Indeed as $D = \mathbb{R}^n$, since the term $\sigma_D(y)$ appears in the objective function, we have $\bar{y} = 0_{\mathbb{R}^{10}}$.
Regarding the concrete stake, we want to find the optimal asset allocation $\bar{\lambda}$ denoted $\mathbf{w}$ in [3].


(a) Objective value via primal-dual interior point

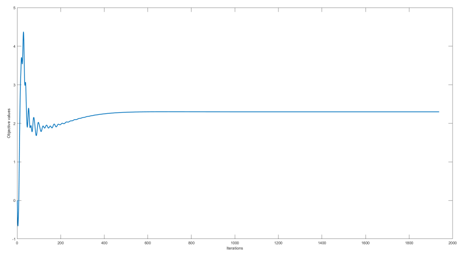
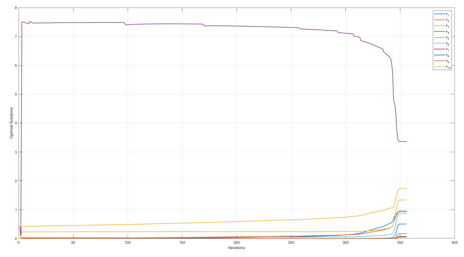

(b) Objective value via parallel splitting

Figure 3.1: Convergence of the objective value of problem $(n = 10, m = 10)$ via parallel splitting and primal-dual interior point methods


(a) Optimal solution via primal-dual interior point

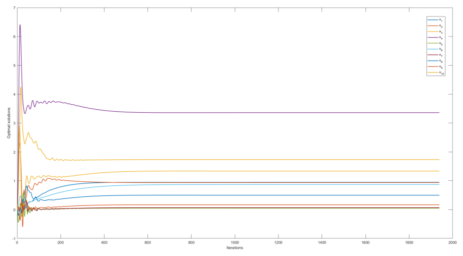

(b) Optimal solution via parallel splitting

Figure 3.2: Convergence of the optimal solution of problem $(n = 10, m = 10)$ via parallel splitting and primal-dual interior point methods

By solving linear system (S_3), MATLAB finds $\bar{\beta} = 3.6973$ and mainly

$$\bar{\lambda} = (0.0000, 0.0516, 0.0000, 0.5575, 0.0000, 0.0000, 0.0000, 0.0000, 0.3909, 0.0000)^\top$$

while the optimal primal objective value is $v(P) = 2.296827$.

Although the primal-dual interior point method requires fewer iterations, the parallel splitting method demonstrated faster computational times for all error tolerances. For instance, our parallel algorithm converged three times faster than the primal-dual interior point method at $\epsilon_1 = 10^{-3}$ (1.0744 compared to 3.580059 seconds) and almost twice as fast at $\epsilon_3 = 10^{-6}$ (2.2137 compared to 4.394282 seconds) while both methods achieved comparable objective values.
The convergence of both the objective function and optimal solution are respectively illustrated in Figure 3.1 and Figure 3.2.

Example 3.4.2. We treat now a case where $n = m = 15$.
The best performances of our tests are illustrated in Table 3.5 - Table 3.8.

MATLAB computed, via parallel splitting, an optimal solution given by

$$\begin{aligned}\bar{x} =&(0.5714, 2.6062, 1.3658, 3.9575, 0.0427, 2.4670, 0.1018, 0.1528,\\ &2.3530, 0.1957, 0.84020.6214, 0.1763, 0.6453, 0.3104)^\top,\\ \bar{\tau} =&(-0.0125, 0.2369, 0.0472, 0.2945, -0.0054, 0.0136, -0.0238, -0.022,\\ &0.0987, -0.0345, -0.0118, -0.0217, -0.0176, -0.0277, -0.0024)^\top,\end{aligned}$$

Algorithm	**PS**$(\mu = 1.83, \nu = 2)$	**PDIP**
Number of iterations	1595	634
CPU time (seconds)	3.4189	5.702284
Objective value	-2.8700	-2.860610

Table 3.5: Comparison of splitting and primal-dual interior point methods for $\epsilon_1 = 10^{-3}$

Algorithm	**PS**$(\mu = 1.83, \nu = 2)$	**PDIP**
Number of iterations	2743	634
CPU time (seconds)	5.7414	5.5653
Objective value	-2.861053	-2.860610

Table 3.6: Comparison of splitting and primal-dual interior point methods ($n = 15, m = 105$) for $\epsilon_2 = 10^{-4}$

Algorithm	**PS**$(\mu = 1.83, \nu = 2)$	**PDIP**
Number of iterations	5432	636
CPU time (seconds)	11.4630	5.5811
Objective value	-2.860556	-2.860555

Table 3.7: Comparison of splitting and primal-dual interior point methods ($n = 15, m = 15$) for $\epsilon_3 = 10^{-6}$

Algorithm	**PS**$(\mu = 1.83, \nu = 2)$	**PDIP**
Number of iterations	8142	639
CPU time (seconds)	17.9040	5.5849
Objective value	-2.860552	-2.860552

Table 3.8: Comparison of splitting and primal-dual interior point methods ($n = 15, m = 15$) for $\epsilon_4 = 10^{-8}$

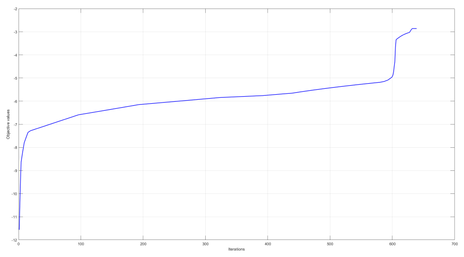
$$\bar{t} = 2.6233.$$

Similarly to Example 3.4.1 we have $\bar{y} = 0_{\mathbb{R}^{15}}$.
By solving linear system (S_3), MATLAB finds $\beta = 5.4949$ as well as the optimal asset allocation $\bar{\lambda}$

$$\begin{aligned}\bar{\lambda} = &(0.0000, 0.0000, 0.7019, 0.0041, 0.2230, 0.0000, 0.0000, 0.0000\\ &0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0710, 0.0000)^\top\end{aligned}$$

while the optimal primal objective value is $v(P) = -2.860552$.

The primal-dual interior point method is still delivering after fewer iterations. Regarding computational time, the parallel splitting method converges faster than its concurrent up to accuracy $\epsilon_2 = 10^{-4}$ but it is no longer the case from $\epsilon_3 = 10^{-6}$ as it is now the primal-pual interior point which demonstrates faster convergence. Both methods yield similar objective values, confirming that they effectively solve the problem, with their efficiency varying depending on the chosen error tolerance.


(a) Objective value via primal-dual interior point

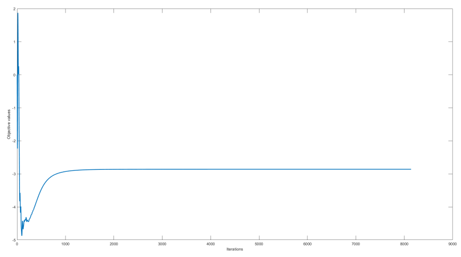
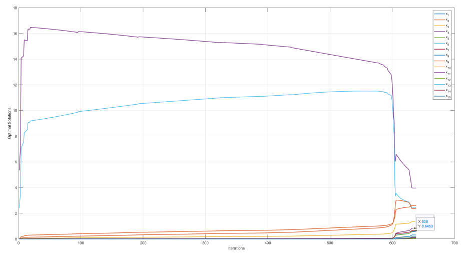

(b) Objective value via parallel splitting

Figure 3.3: Convergence of the objective value of problem $(n = 15, m = 15)$ via parallel splitting and primal-dual interior point methods


(a) Optimal solution via primal-dual interior point

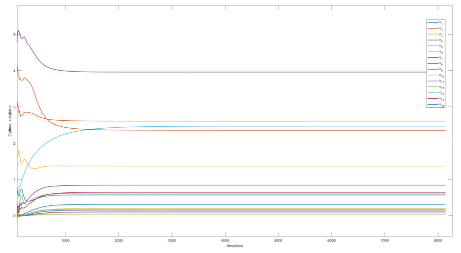

(b) Optimal solution via parallel splitting

Figure 3.4: Convergence of the optimal solution of problem $(n = 15, m = 15)$ via parallel splitting and primal-dual interior point methods

The convergence of both the objective function and optimal solution are respectively illustrated in Figure 3.3 and Figure 3.4.

In the following two examples, we explore two different scenarios based on the number of samples m and the number of financial instruments n. These results will enable us to draw some conjectures. In the case of the parallel splitting method, we executed our MATLAB programs with various step sizes ν and parameters μ, starting from the origin in each case. The same stopping criterion was applied across both examples. The data for these tests, including the weights p, vectors c and h and the matrix A were randomly generated using MATLAB. This approach is the same as in other examples in this chapter on portfolio optimization, ensuring that the comparisons are consistent.

Example 3.4.3. Let $n = 10, m = 5$.

MATLAB computed, via parallel splitting, an optimal solution given by

$$\begin{aligned}
\bar{x} &= (2.8000, 0.1145, 0.7090, 0.6939, 0.3219)^\top, \\
\bar{\tau} &= (0.7222, -0.0606, -0.0598, -0.0089, -0.0820)^\top, \\
\bar{t} &= 10.7293.
\end{aligned}$$

As $D = \mathbb{R}^{10}$, we have $\bar{y} = 0_{\mathbb{R}^{10}}$.
By solving linear system (S_3), the optimal asset allocation $\bar{\lambda}$ is given by (as well as $\bar{\beta} = 2.0589$)

$$\bar{\lambda} = (0.0000, 0.8763, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.000, 0.1237)^\top$$

Algorithm	**PS**$(\mu = 1.45, \nu = 2.2)$	**PDIP**
Number of iterations	353	470
CPU time (seconds)	0.3906	4.3591
Objective value	9.667381	9.666822

Table 3.9: Comparison of splitting and primal-dual interior point methods $(n = 10, m = 5)$ for $\epsilon_1 = 10^{-3}$

Algorithm	**PS**$(\mu = 1.45, \nu = 2.2)$	**PDIP**
Number of iterations	460	471
CPU time (seconds)	0.5383	4.1449
Objective value	9.667280	9.667154

Table 3.10: Comparison of splitting and primal-dual interior point methods $(n = 10, m = 5)$ for $\epsilon_2 = 10^{-4}$

Algorithm	**PS**$(\mu = 1.45, \nu = 2.2)$	**PDIP**
Number of iterations	673	474
CPU time (seconds)	0.7169	4.1813
Objective value	9.667242	9.667240

Table 3.11: Comparison of splitting and primal-dual interior point methods $(n = 10, m = 5)$ for $\epsilon_3 = 10^{-6}$

Algorithm	**PS**$(\mu = 1.45, \nu = 2.2)$	**PDIP**
Number of iterations	886	476
CPU time (seconds)	0.8139	4.4157
Objective value	9.667241	9.667241

Table 3.12: Comparison of splitting and primal-dual interior point methods $(n = 10, m = 5)$ for $\epsilon_4 = 10^{-8}$

while the optimal primal objective value is $v(P) = 9.667241$.

The results displayed in Table 3.9 - Table 3.12 highlight that although the primal-dual interior point method typically requires fewer iterations than the parallel splitting method, this is not the case in this example. Notably, for error tolerances $\epsilon_1 = 10^{-3}$ and $\epsilon_2 = 10^{-4}$, our parallel splitting incurs less computational cost (353 iterations compared to 470 for PDIP at ϵ_1) and (460 compared to 471 at ϵ_2). Furthermore, the parallel splitting method demonstrates faster computational times for all error tolerances. Specifically, our parallel algorithm converged 10 times faster than the primal-dual interior point method at $\epsilon_1 = 10^{-3}$ and almost 9 as fast at $\epsilon_4 = 10^{-8}$ while both methods achieved comparable objective values.

Example 3.4.4. Suppose now that $n = 5, m = 10$.

MATLAB computed, via parallel splitting, an optimal solution given by

$$\bar{x} = (1.8950, 0.2273, 4.3454, 0.8383, 0.6533, 0.3573, 0.0609, 0.8706, 0.9573, 0.7764)^\top,$$

$\bar{\tau} = (0.0306, -0.0448, 0.6480, -0.0177, -0.0364, -0.0040, -0.0263, -0.0097, -0.0051, -0.0237)^\top,$
$\bar{t} = 4.8743.$

Once again, considering the choice $D = \mathbb{R}^5$, we have $\bar{y} = 0_{\mathbb{R}^5}$. By solving linear system (S_3), MATLAB finds $\bar{\beta} = 2.2386$ and the optimal asset allocation

$$\bar{\lambda} = (0.3944, 0.000, 0.6056, 0.0000, 0.0000)^\top$$

while the optimal primal objective value is $v(P) = 3.529863$.

Algorithm	**PS**$(\mu = 1.65, \nu = 2.2)$	**PDIP**
Number of iterations	714	17
CPU time (seconds)	1.1523	0.6695
Objective value	3.526018	3.529451

Table 3.13: Comparison of splitting and primal-dual interior point methods $(n = 5, m = 10)$ for $\epsilon_1 = 10^{-3}$

Algorithm	**PS**$(\mu = 1.65, \nu = 2.2)$	**PDIP**
Number of iterations	1003	18
CPU time (seconds)	1.5743	0.7176
Objective value	3.530150	3.529777

Table 3.14: Comparison of splitting and primal-dual interior point methods $(n = 5, m = 10)$ for $\epsilon_2 = 10^{-4}$

Algorithm	**PS**$(\mu = 1.65, \nu = 2.2)$	**PDIP**
Number of iterations	1611	21
CPU time (seconds)	2.3536	0.6885
Objective value	3.529864	3.529862

Table 3.15: Comparison of splitting and primal-dual interior point methods $(n = 5, m = 10)$ for $\epsilon_3 = 10^{-6}$

Algorithm	**PS**$(\mu = 1.65, \nu = 2.2)$	**PDIP**
Number of iterations	2271	24
CPU time (seconds)	3.1418	0.7093
Objective value	3.529863	3.529863

Table 3.16: Comparison of splitting and primal-dual interior point methods $(n = 5, m = 10)$ for $\epsilon_4 = 10^{-8}$

While both methods are reliable in finding an optimal solution, in this instance, as shown in Table 3.13 - Table 3.13, the primal-dual interior point clearly outperforms the parallel splitting algorithm in both computational efficiency and iteration count, making it the preferred method for this problem.

3.5 Discussion

Taking into account the previous four instances, we can draw the following conclusions.

- In general, and for most error tolerances, the primal-dual interior point method requires notably fewer iterations compared to the parallel splitting algorithm. However, we should mention that each iteration of PDIP involves more computational complexity compared to parallel splitting, as it includes operations like line search and the evaluation of the surrogate duality gap. These additional steps increase the computational burden per iteration despite requiring fewer iterations to converge.
- The results highlight that PDIP is particularly advantageous when the number of samples m is large relative to the number of financial instruments n. It converges more rapidly than its concurrent.
- When the number of samples is smaller relative to the number of financial instruments, parallel splitting outperforms in terms of computational speed, even though it requires more iterations. This indicates that parallel splitting is more effective in scenarios where the sample size is less critical.

3.6 Perspectives on parallel splitting methods for entropy optimization

The primal-dual algorithm developed by Ahmadi-Javid is generally more efficient than our splitting algorithm for large sample sizes (m). This advantage arises because the primal-dual algorithm handles a convex/differentiable problem where the number of variables and constraints is independent of the sample size. In contrast, our splitting algorithm addresses the primal problem, which scales with the sample size, leading to decreased efficiency and longer convergence times as m increases. When the sample size is relatively small (say less than 20), our splitting algorithm can be faster, given the right parameter settings. This is because, in scenarios with a small number of samples, the splitting method can be more efficient compared to the primal-dual interior point approach.
In cases where the sample size m is very large, the primal problem becomes less practical for splitting methods due to the high-dimensional nature of the primal variables ($\mathbb{R}^m$). Despite this, our splitting algorithm often demonstrates faster convergence compared to other primal-solving methods. The primal-dual interior point method introduced in [3] solves the dual problem, where the dual variables lie in $\mathbb{R}^n$ (with n typically being smaller than m), which contributes to its efficiency for large sample sizes.

Therefore, in situations where the number of financial instruments n and the number of samples m are similar (such as $5 \times 5, 10 \times 10, 15 \times 15$), our splitting algorithm performs comparably to or even better than the primal-dual algorithm in terms of computational time.

Conclusion

The parallel splitting methods (both primal and dual) have proven to be efficient for solving location problems, particularly minimax location problems that involve nonsmooth objective functions. However, our results demonstrated that, in instances where it can be implemented, the Chambolle-Pock algorithm significantly outperforms both splitting methods in terms of computational cost (number of iterations) and time. Both the Chambolle-Pock and the parallel splitting (which is a version of Douglas-Rachford) algorithms are exact methods, meaning they converge to the exact solution of the problem and deliver the precise optimal objective value.

The second method we explored during this internship is a variant of the mirror descent algorithm. This approach involves smoothing the functions in the objective via the Moreau envelope, which introduces a smoothing parameter and utilizes a mirror map. While this method does converge to a solution that yields a reasonable objective value, it often does not achieve the exact solution. To obtain a result closer to the exact solution, the smoothing parameter must be made progressively smaller, which in turn increases the computational cost in terms of both iterations and CPU time. The challenge lies in finding the optimal balance between accuracy and computational efficiency.

Overall, parallel splitting and Chambolle-Pock are generally more suitable than mirror descent for these types of problems. However, it's important to note that, unlike the Chambolle-Pock algorithm, mirror descent requires less stringent assumptions and offers some minor advantages, such as a modest increase in computational cost across different levels of error tolerance accuracy. The Chambolle-Pock algorithm is a specific type of primal-dual method tailored for problems involving convex functions and linear operators. It exhibits strong convergence properties and an ability to find accurate solutions.
The mirror descent is a more general optimization technique that adapts gradient descent methods to handle constraints and regularizations. It is particularly useful for problems where the feasible set is complex. However, the performance of the mirror descent method depends heavily on the choice of parameters and step sizes, which can affect its convergence to the exact optimal value.

When dealing with applications where exact objective values are crucial, primal-dual splittings and Chambolle-Pock algorithms might be preferred due to their strong theoretical convergence guarantees. The mirror descent method might be used for its flexibility and general applicability, but with the understanding that additional tuning might be needed to achieve the desired accuracy and thus affecting computational speed.
Finally, in the context of our locations problems, inertial approaches have shown promise in accelerating convergence for primal problems, but have not outperformed non-inertial methods for dual problems. Further experimentation could offer a chance to get the desired improvements.

Our investigations related to portfolio optimization showed that parallel splitting methods may offer an alternative to the primal-dual interior point (PDIP) algorithm, particularly when the sample size is small. While PDIP generally excels in handling large sample sizes due to its efficient processing of the dual problem, parallel splitting algorithms prove to be competitive and even advantageous in cases where the number of financial instruments and samples are similar. Although its performance diminishes as the sample size grows, we proved that the parallel splitting algorithm outperformed the PDIP method when the number of samples was relatively small.

While it wasn't feasible to use the Chambolle-Pock algorithm due to unmet assumptions, we attempted to implement the mirror descent method in the context of entropy optimization. However, our investigations revealed that this approach was not suitable, as we couldn't identify the optimal parameters. The problem's dimensionality may have contributed to this difficulty.

Appendix

Theorem 3.6.1. *[11, Theorem 2.1] Assume that C is closed convex and $0_X \in C$, Ω is closed and convex, the function $f : X \to \bar{\mathbb{R}}$ is convex and lowersemicontinuous. Suppose that*

$$C^0 \cap \mathrm{dom} f^* \cap \mathrm{dom}\sigma_\Omega \neq \emptyset. \tag{3.20}$$

Furthermore, we make one of the following hypothesis

- $\mathrm{epi}\gamma_C + \mathrm{epi} f + (\Omega \times \mathbb{R}_+)$ *is closed,*
- $\exists x^* \in C^0 \cap \mathrm{dom} f^* \cap \mathrm{dom}\sigma_\Omega$ *such that two of the functions* $\delta_{C^0}, f^*, \sigma_\Omega$ *are continuous at* x^*.

Then, $\mathcal{T}^C_{\Omega,f}$ is proper, convex and lower semicontinuous and the infimal convolution $\gamma_C \square f \square \delta_\Omega$ is exact, i,e

$$\mathcal{T}^C_{\Omega,f}(x) = \min_{y \in X, z \in \Omega} \{\gamma_C(x - y - z) + f(y)\}, \quad \forall x \in X. \tag{3.21}$$

Theorem 3.6.2. *[5, Theorem 27.8] Let $n \in \mathbb{N}$ $n \geq 2$ and $f_i : \mathbb{R}^d \to \bar{\mathbb{R}}$ be a proper, convex and lower semicontinuous function for $i = 1, \cdots, n$.*
Suppose that

$$\min_{x \in \mathbb{R}^d} \sum_{i=1}^n f_i(x) \tag{P}$$

has at least one solution and that the qualification condition

$$\mathrm{dom}(f_1) \cap \bigcap_{i=2}^n \mathrm{int}\left(\mathrm{dom}(f_i)\right) \neq \emptyset$$

is satisfied. Let $(\mu_k)_{k\in\mathbb{N}} \subseteq [0, 2]$ such that

$$\sum_{k \in \mathbb{N}} \mu_k (2 - \mu_k) = +\infty.$$

Let $\nu > 0$ and $(x_{i,0})_{i=1}^n \in (\mathbb{R}^d)^n$ and consider for all $k \in \mathbb{N}$

$$\begin{aligned}
r_k &= \frac{1}{n} \sum_{i=1}^n x_{i,k}, \\
y_{i,k} &= \mathrm{prox}_{\nu f_i}(x_{i,k}), \quad i = 1, \cdots, n, \\
q_k &= \frac{1}{n} \sum_{i=1}^n y_{i,k}, \\
x_{i,k+1} &= x_{i,k} + \mu_k (2q_k - r_k - u_{i,k}), \quad i = 1, \cdots, n.
\end{aligned}$$

Then, $(r_k)_{k\in\mathbb{N}}$ converges to an optimal solution of (P).

Algorithm 5 Chambolle-Pock - general case

[9, Algorithm 1]

```
1: Input: Choose $(x^0, y^0) \in X \times Y$ and constants $\sigma, \tau > 0$, $\theta \in [0,1]$ and set $\bar{x}^0 = x^0$.
2: For $k \geq 0$ do
3: $y^{k+1} = \mathrm{prox}_{F^*}(y^k + \sigma K \bar{x}^k)$
4: $x^{k+1} = \mathrm{prox}_G(x^k - \tau K^* y^{k+1})$
5: $\bar{x}^{k+1} = x^{k+1} + \theta(x^{k+1} - x^k)$
6: End for
```

Algorithm 6 Mirror Descent - general case

[6, Algorithm 4]

```
1: Input: Choose $x^0 \in \mathrm{Im}\nabla H^* \cap C$, the smoothing parameters $\gamma^k > 0$, and the step sizes $\nu^k > 0$,
   for $k \geq 0$.
2: For $k \geq 0$ do
3: $\psi_0^k := x^k$
4: for $i := 1, \ldots, m$ do
5:     $\psi_i^k := \nabla H^* \left(\nabla H(\psi_{i-1}^k) - \epsilon_i^k \frac{\nu^k}{\gamma^k} p_i A_i^* \left(A_i \psi_{i-1}^k - \mathrm{prox}_{\gamma^k f_i}\left(A_i \psi_{i-1}^k\right)\right)\right)$
6: end for
7: $x^{k+1} := \mathrm{prox}_{\nu^k g}^H(\psi_m^k)$
8: End for
```

$\epsilon_i^k \in \{0,1\}$ is a random variable independent of ψ_{i-1}^k and $\mathbb{P}(\epsilon_i^k = 1) = p_i, 1 \leq i \leq m$ and $k \geq 0$.

Algorithm 7 Inertial Parallel Splitting - general case

[12, Proposition 5.2]

```
1: Input: Choose $(t_{i,0})_{1\leq i\leq m} \in X$, $(p_{i,-1})_{1\leq i\leq m} \in X$, and constants $\omega_i > 0$, $\epsilon_i \in [0,1[$, $\lambda \in ]0,2[$,
   and $y_0 \in \mathrm{Arg\,min}_{z\in E} \sum_{i=1}^m \omega_i \|L_i z - t_{i,0}\|_i^2$.
2: For $k \geq 0$ do
3: for $i = 1, \cdots, n$ do
4:     $p_{i,n} = \mathrm{prox}_{\frac{(1-\epsilon_i)}{\omega_i} f_i}\left((1-\epsilon_i)t_{i,n} + \epsilon_i p_{i,n-1}\right) + a_{i,n}$
5:     $c_n \in \mathrm{Arg\,min}_{z\in E} \sum_{i=1}^m \omega_i \|L_i z - p_{i,n}\|_i^2$
6: end for
7: for $i = 1, \cdots, n$ do
8:     $t_{i,n+1} = t_{i,n} + \lambda_n \left(L_i(2c_n - y_n) - p_{i,n}\right)$
9: end for
10: $y_{n+1} = y_n + \lambda_n(c_n - y_n)$
11: End for
```

Proposition 3.6.3. *[12, Proposition 5.4] Let $(\epsilon_1, \dots, \epsilon_n) \in [0,1[^n$ and $(\lambda_k)_{k\in\mathbb{N}}$ be a sequence of real numbers. For every $i = 1, \cdots, n$, let $(a_{i,k})_{k\in\mathbb{N}}$ be a sequence in X_i. Let $\{y_k\}_{n\in\mathbb{N}} \subset E$, $\{c_k\}_{k\in\mathbb{N}} \subset E$, $\{t_k\}_{k\in\mathbb{N}} \subset X$, and $\{p_k\}_{k\geq -1} \subset X$ be generated by Algorithm 7.*
Suppose that the following assumptions hold:

1. $0 \in \operatorname{sqri}\{(L_1 z - x_1, \dots, L_n z - x_n) \mid z \in E, \quad x_1 \in \operatorname{dom} f_1, \quad \dots, \quad x_n \in \operatorname{dom} f_n\}$.
2. *There exists $\lambda \in]0,2[$ such that $\forall k \in \mathbb{N}, \lambda \leq \lambda_{k+1} \leq \lambda_k < 2$.*
3. $\forall i \in \{1, \dots, n\}, \sum_{k\in\mathbb{N}} \|a_{i,k}\|_i < +\infty$.

If the set of solutions to (P_L) is nonempty, then $(L_1 y_k, \dots, L_n y_k)_{k\in\mathbb{N}}$, $(L_1 c_k, \dots, L_n c_k)_{k\in\mathbb{N}}$, and $(p_k)_{k\in\mathbb{N}}$ converge weakly to $(L_1\hat{y}, \dots, L_n\hat{y})$ where $\hat{y}$ is a solution to (P_L).

Corollary 3.6.4. *[14, Corollary 2.3] Let $h : X \to \mathbb{R}$ such that $h(x) = w\|x\|_X, w \geq 1$. Then, for every $(x, \xi) \in X \times \mathbb{R}$,*

$$\mathrm{P}_{\mathrm{epi}(w\|\cdot\|_X)}(x,\xi) = \begin{cases} (x,\xi), & \text{if } w\|x\|_X \leq \xi, \\ (0,0), & \text{if } \|x\|_X \leq -w\xi, \\ \left(\frac{\|x\|_X + w\xi}{\|x\|_X(w^2+1)} x, \frac{w\|x\|_X + w^2\xi}{w^2+1}\right), & \text{otherwise.} \end{cases} \tag{3.22}$$

Lemma 3.6.5. *[14, Lemma 2.1] For $w \geq 1, p_i \in X, i = 1, \cdots, n$,*

$$\mathrm{P}_{\mathrm{epi}(w\|.-p_i\|_X)}(x,\xi) = \mathrm{P}_{\mathrm{epi}(w\|\cdot\|_X)}(x - p_i, \xi) + (p_i, 0). \tag{3.23}$$

Theorem 3.6.6. *[4] Let $p > 0$ and*

$$h : \mathbb{R} \to \mathbb{R}, \quad h(x) = \begin{cases} px\ln x, & \text{if } x \geq 0, \\ +\infty, & \text{if } x < 0. \end{cases} \tag{3.24}$$

Then it holds for all $(x,\xi) \in \mathbb{R} \times \mathbb{R}$ that

$$P_{\mathrm{epi}\,h}(x,\xi) = \begin{cases} (x,\xi), & \text{if } px\ln x \leq \xi, \\ (0,\xi), & \text{if } x \leq 0 \text{ and } \xi \geq 0, \\ (e^{\bar{z}}, p\bar{z}e^{\bar{z}}), & \text{otherwise,} \end{cases} \tag{3.25}$$

where $\bar{z} \in \mathbb{R}$ is the unique solution of the equation

$$(p^2z^2 + p^2z + 1)e^z - p\xi(z+1) - x = 0 \tag{3.26}$$

that fulfills $p\bar{z}e^{\bar{z}} \geq \xi$. More precisely,

1. *if $\xi \leq -\frac{p}{e}$, then $\bar{z} \in \mathbb{R}$,*
2. *if $-\frac{p}{e} < \xi \leq 0$ and $x \leq \frac{1}{e}$, then $\bar{z} \in (-1, W_{-1}(\frac{\xi}{p})]$,*
3. *if $-\frac{p}{e} < \xi \leq 0$ and $x > \frac{1}{e}$, then $\bar{z} \in [W(\frac{\xi}{p}), +\infty)$,*
4. *if $\xi > 0$, then $\bar{z} \in [W(\frac{\xi}{p}), +\infty)$.*

Corollary 3.6.7. *[4] Let $0 < p \leq 1$. Then it holds for all $(x, \xi) \in \mathbb{R} \times \mathbb{R}$ that*

$$P_{\operatorname{epi} h}(x, \xi) = \begin{cases} (x, \xi), & \text{if } px \ln x \leq \xi, \\ (0, \xi), & \text{if } x \leq 0 \text{ and } \xi \geq 0, \\ (e^{\bar{z}}, p\bar{z}e^{\bar{z}}), & \text{otherwise}, \end{cases} \tag{3.27}$$

where $\bar{z} \in \mathbb{R}$ is the unique solution of the equation

$$(p^2 z^2 + p^2 z + 1)e^z - p\xi(z + 1) - x = 0 \tag{3.28}$$

that fulfills $p\bar{z}e^{\bar{z}} \geq \xi$. More precisely, exactly one of the following six cases holds:

1. *$\xi \leq -\frac{p}{e}$ and $x < 0$, then $\bar{z} \in \mathbb{R}$ is unique and $\bar{z} \in (-1, 0]$,*
2. *$\xi \leq -\frac{p}{e}$ and $x \geq 0$, then $\bar{z} \in \mathbb{R}$ is unique and $\bar{z} \in (-1, x]$,*
3. *$-\frac{p}{e} < \xi \leq 0$ and $x < 0$, then $\bar{z} \in \mathbb{R}$ is unique and $\bar{z} \in (-1, W^{-1}(\frac{\xi}{p})]$,*
4. *$-\frac{p}{e} < \xi \leq 0$ and $0 \leq x \leq 1/e$, then $\bar{z} \in \mathbb{R}$ is unique and $\bar{z} \in \Big[W^{-1}(\frac{\xi}{p}) + \ln(x) - 1, W^{-1}(\frac{\xi}{p})\Big]$,*
5. *$-\frac{p}{e} < \xi \leq 0$ and $x > 1/e$, then $\bar{z} \in \mathbb{R}$ is unique and $\bar{z} \in \Big[W(\frac{\xi}{p}), W(\frac{\xi}{p}) + \ln(x) + 2\Big]$,*
6. *$\xi > 0$, then $x > 1$ and $\bar{z} \in \mathbb{R}$ is unique and $\bar{z} \in \Big[W(\frac{\xi}{p}), W(\frac{\xi}{p}) + \ln(x) + 1\Big]$.*

Proposition 3.6.8. *[5, Example 28.15] Let u a nonzero vector in a Hilbert space H. Let $\eta \in \mathbb{R}$ and set*

$$C = \{x \in H \mid \langle x, u \rangle = \eta\}. \tag{3.29}$$

Then

$$\forall x \in H, \quad P_C x = x + \frac{\eta - \langle x, u \rangle}{\|u\|^2} u. \tag{3.30}$$

Proposition 3.6.9. *[5, Proposition 28.16] Let $u \in H$, let $\eta \in \mathbb{R}$, and set*

$$C = \{x \in H \mid \langle x, u \rangle \leq \eta\}. \tag{3.31}$$

Then exactly one of the following holds:

(i) $u = 0$ and $\eta \geq 0$, in which case $C = H$ and $P_C = \mathrm{Id}$.

(ii) $u = 0$ and $\eta < 0$, in which case $C = \emptyset$.

(iii) $u \neq 0$, in which case $C \neq \emptyset$ and

$$\forall x \in H, \quad P_C x = \begin{cases} x, & \text{if } \langle x, u \rangle \leq \eta, \\ x + \frac{\eta - \langle x, u \rangle}{\|u\|^2} u, & \text{if } \langle x, u \rangle > \eta. \end{cases} \tag{3.32}$$

Bibliography

[1] S. Adly, L. Bourdin, and F. Caubet. On a decomposition formula for the proximal operator of the sum of two convex functions, June 2018. arXiv:1707.08509 [math].

[2] A. Ahmadi-Javid. Entropic Value-at-Risk: A New Coherent Risk Measure. *Journal of Optimization Theory and Applications*, 155(3):1105–1123, 2012. Publisher: Springer.

[3] A. Ahmadi-Javid and M. Fallah-Tafti. Portfolio Optimization with Entropic Value-at-Risk. *European Journal of Operational Research*, 279(1):225–241, November 2019. arXiv:1708.05713 [math, q-fin].

[4] N. Baloul, S. Grad, and O. Wilfer. Proximal splitting methods for solving entropy constrained optimization problems, in preparation.

[5] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer International Publishing, Cham, 2017.

[6] S. Bitterlich and S.-M. Grad. Stochastic incremental mirror descent algorithms with Nesterov smoothing. *Numerical Algorithms*, August 2023. Publisher: Springer Verlag.

[7] R. Bot, S. Grad, and G. Wanka. Entropy constrained programs and geometric duality obtained via Fenchel-Lagrange duality approach, 2004.

[8] R. I. Bot, S.-M. Grad, and G. Wanka. *Duality in Vector Optimization*. Vector Optimization. Springer, Berlin, Heidelberg, 2009.

[9] A. Chambolle and T. Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, May 2011.

[10] O. Cornejo and C. Michelot. A Proximal Solution for a Class of Extended Minimax Location Problem. In *Computational Science and Its Applications – ICCSA 2005*, pages 712–721, Berlin, Heidelberg, 2005. Springer.

[11] S.-M. Grad and O. Wilfer. A proximal method for solving nonlinear minmax location problems with perturbed minimal time functions via conjugate duality. *Journal of Global Optimization*, 74(1):121–160, 2019. Publisher: Springer.

[12] J.-C. Pesquet and N. Pustelnik. A Parallel Inertial Proximal Optimization Method. *Pacific Journal of Optimization*, 8(2):273–305, 2012. Publisher: Yokohama Publishers.

[13] G. Wanka and O. Wilfer. Duality results for nonlinear single minimax location problems via multi-composed optimization. *Mathematical Methods of Operations Research*, 86(2):401–439, 2017. Publisher: Springer & Gesellschaft für Operations Research (GOR) & Nederlands Genootschap voor Besliskunde (NGB).

[14] G. Wanka and O. Wilfer. Formulae of epigraphical projection for solving minimax location problems. *Pacific Journal of Optimization*, 16:288–313, 2017.

[15] G. Wanka and O. Wilfer. A Lagrange duality approach for multi-composed optimization problems. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 25(2):288–313, 2017. Publisher: Springer & Sociedad de Estadística e Investigación Operativa.