Faculty of Mathematics and Computer Science

Babeş-Bolyai University

Erika Nagy

Numerical methods for approximating zeros of operators and for solving variational inequalities with applications

Doctoral Thesis Summary

Supervisor: **Prof. Dr. Gábor Kassay**

Cluj-Napoca

2013

Contents

1 Introduction **5**

2 Background of the theory of monotone operators **8**

2.1 Operators . . . 8

2.2 Functions . . . 9

3 The primal-dual splitting technique **12**

3.1 Monotone inclusion problem formulation . . . 12

3.2 The primal-dual splitting algorithm . . . 14

3.3 Applications to convex minimization problems . . . 16

4 Numerical experiments **25**

4.1 Image processing . . . 25

4.1.1 Image deblurring . . . 25

4.1.2 Experimental setting . . . 28

4.2 Multifacility location problems . . . 32

4.2.1 Problem manipulation . . . 34

4.2.2 Computational experience . . . 36

4.3 Average consensus for colored networks . . . 38

4.3.1 Proposed algorithm . . . 38

4.3.2 Performance evaluation . . . 40

4.4 Support-vector machines classification . . . 43

4.4.1 The soft-margin hyperplane . . . 44

4.4.2 Experiments with digit recognition . . . 45

5 Interdisciplinary application of our algorithm **48**

5.1 Optimization in biology 48
5.2 Pattern recognition techniques 49

References **51**

Chapter 1

Introduction

The existence of optimization methods can be traced to the days of Newton, Lagrange, and Cauchy. The development of differential calculus methods for optimization was possible because of the contributions of Newton and Leibnitz to calculus. The foundations of calculus of variations, which deals with the minimization of functions, were laid by Bernoulli, Euler, Lagrange and Weierstrass. The method of optimization for constrained problems, which involve the addition of unknown multipliers, became known by the name of its inventor, Lagrange. Cauchy made the first application of the steepest descent method to solve unconstrained optimization problems. By the middle of the twentieth century, the high-speed digital computers made implementation of the complex optimization procedures possible and stimulated further research on newer methods. Spectacular advances followed, producing a massive literature on optimization techniques. This advancement also resulted in the emergence of several well defined new areas in optimization theory [67].

In recent years several splitting algorithms [43] have emerged for solving monotone inclusion problems involving parallel sums and compositions with linear continuous operators, which eventually are reduced to finding the zeros of the sum of a maximal monotone operator and a cocoercive or a monotone and Lipschitz continuous operator. The later problems were solved by employing in an appropriate product space forward-backward or forward-backward-forward algorithms, respectively, and gave rise to so-called primal-dual splitting methods (see [9, 16, 20, 23, 61] and the references therein).

Recently, one can remark the interest of researchers in solving systems of monotone inclusion problems [1,5,6,22]. This is motivated by the fact that convex optimization problems arising, for instance, in areas like image processing [13], multifacility location problems [18, 31], average consensus in network coloring [46, 47] and support vector machines classification [27,28] are to be solved with respect to multiple variables, very often linked in different manners, for instance, by linear equations.

The present research is motivated by the investigations made in [1]. The authors propose there an algorithm for solving coupled monotone inclusion problems, where the variables are linked by some operators which satisfy jointly a cocoercivity property. Our aim is to overcome the necessity of having differentiability for some of the functions occurring in the objective of the convex optimization problems in [1]. To this end we consider first a more general system of monotone inclusions, for which the coupling operator satisfies a Lipschitz continuity property, along with its dual system of monotone inclusions in an extended sense of the Attouch-Théra duality (see [2]). The simultaneous solving of the primal and dual system of monotone inclusions is reduced to the problem of finding the zeros of the sum of a maximal monotone operator and a monotone and Lipschitz continuous operator in an appropriate product space. The latter problem is solved by a forward-backward-forward algorithm, fact that allows us to provide for the resulting iterative scheme, which proves to have a high parallelizable formulation, both weak and strong convergence assertions.

The thesis is organized in five chapters and a bibliography. In the first chapter we state the initial problem that stimulated this research. H. Attouch, L.M. Briceno-Arias and P.L. Combettes proposed in [1] a parallel splitting method for solving systems of coupled monotone inclusions in Hilbert spaces, establishing its convergence under the assumption that solutions exist. The method can handle an arbitrary number of variables and nonlinear coupling schemes.

In the second chapter we give some necessary notations and preliminary results in order to facilitate the reading of the manuscript and to help the reader to understand easily the following parts. Notions like nonexpansive operators, parallel sum, cocoercivity, conjugation, Fenchel duality, subdifferentiability and maximal monotone operators are introduced together with some essential splitting algorithms like the Douglas-Rachford algorithm, forward-backward algorithm and Tseng's algorithm. These algorithms have been studied more extensively in [4].

Chapter three is devoted to the primal-dual splitting algorithm for solving the problem considered in this thesis and investigates its convergence behaviour. The operators arising in each of the inclusions of the system are processed in every iteration separately, namely, the single-valued are evaluated explicitly, which is equivalent to a forward step, while the set-valued ones via their resolvents, which is equivalent to a backward step. In addition, most of the steps in the iterative scheme can be executed simultaneously, this making the method applicable to a variety of convex minimization problems. The solving of convex optimization problems with multiple variables is also presented in this chapter.

In Chapter four, the numerical performances and effectiveness of the proposed splitting algorithm are emphasized through several numerical experiments, like image processing, where a noisy and blurry image is cleared; multifacility location problems, where the objective is to minimize the energy path between existing facilities and multiple new facilities, which have to be located among the existing ones; average consensus on colored networks, which consists in calculating the average of the value of each node in a recursive and distributed way; image classification via support vector machines, where the aim is to construct a decision function with the help of training data, which should assign every new data correctly with a low misclassification rate.

Finally, the last part of the thesis presents a collaboration with a research team in biology studying the presence of picoalgae in Transylvanian salt lakes. The aim was to adapt the proposed algorithm for image processing and pattern recognition for microscopic images and electrophoretograms. The developing process included the work with epifluorescence microscopic images of picoalgae and the goal was to count and distinguish picoalgae from the background noise.

Keywords

convex optimization, coupled systems, forward-backward-forward algorithm, Lipschitz continuity, monotone inclusion, operator splitting, pattern recognition

Chapter 2

Background of the theory of monotone operators

The mathematical notion of a Hilbert space was named after the German mathematician David Hilbert. The earliest Hilbert spaces were studied as infinite-dimensional function spaces in the first decade of the 20^{th} century by David Hilbert, Frigyes Riesz and Erhard Schmidt. Throughout this work $\mathbb{R}_+$ denotes the set of non negative real numbers, $\mathbb{R}_{++}$ the set of strictly positive real numbers and $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ the extended real-line. We consider Hilbert spaces endowed with the *scalar (or inner) product* $\langle\cdot|\cdot\rangle$ and the associated *norm* $\|\cdot\| = \sqrt{\langle\cdot|\cdot\rangle}$. In order to avoid confusion, when needed, appropriate indices will be used for the inner product and norm. The symbols $\rightharpoonup$ and $\to$ denote weak and strong convergence, respectively.

2.1 Operators

Let $\mathcal{H}$ be a real Hilbert space and let $2^{\mathcal{H}}$ be the *power set* of $\mathcal{H}$, i.e., the family of all subsets of $\mathcal{H}$. The notation $M : \mathcal{H} \to 2^{\mathcal{H}}$ means that M is a *set-valued operator*, i.e. M maps every point $x \in \mathcal{H}$ to a set $Mx \subset \mathcal{H}$. We denote by $\operatorname{zer} M = \{x \in \mathcal{H} : 0 \in Mx\}$ the set of *zeros* of M and by $\operatorname{gra} M = \{(x,u) \in \mathcal{H} \times \mathcal{H} : u \in Mx\}$ the *graph* of M. The *domain* and the *range* of M are

$$\operatorname{dom} M = \{x \in \mathcal{H} : Mx \neq \emptyset\} \ \text{ and } \ \operatorname{ran} M = M(\mathcal{H}),$$

respectively. In addition, the closure of $\operatorname{dom} M$ is denoted by $\overline{\operatorname{dom}} M$ and the closure of $\operatorname{ran} M$ is denoted by $\overline{\operatorname{ran}} M$. The *reversal* of M is $\check{M} : x \mapsto M(-x)$.

The *inverse* of M, denoted by $M^{-1} : \mathcal{H} \to 2^{\mathcal{H}}$, is defined through its graph

$$\operatorname{gra} M^{-1} = \{(u, x) \in \mathcal{H} \times \mathcal{H} : (x, u) \in \operatorname{gra} M\}.$$

Thus, for every $(x, u) \in \mathcal{H} \times \mathcal{H}, u \in Mx \Leftrightarrow x \in M^{-1}u$. Moreover, $\operatorname{dom} M^{-1} = \operatorname{ran} M$ and $\operatorname{ran} M^{-1} = \operatorname{dom} M$.

The *sum* and the *parallel sum* of two set-valued operators M, $N : \mathcal{H} \to 2^{\mathcal{H}}$ are defined as

$$M + N : \mathcal{H} \to 2^{\mathcal{H}}, (M + N)(x) = M(x) + N(x), \quad \forall x \in \mathcal{H} \tag{2.1.1}$$

and

$$M \,\square\, N : \mathcal{H} \to 2^{\mathcal{H}}, M \,\square\, N = \left(M^{-1} + N^{-1}\right)^{-1}. \tag{2.1.2}$$

Let $\mathcal{G}$ be another real Hilbert space and consider the *single-valued operator* $L : \mathcal{H} \to \mathcal{G}$ (also called mapping), that maps every point x in $\mathcal{H}$ to a point Lx in $\mathcal{G}$. The *norm* of the linear continuous operator L is defined as $\|L\| = \sup\{\|Lx\| : x \in \mathcal{H}, \|x\| \leq 1\}$, while $L^* : \mathcal{G} \to \mathcal{H}$, defined by $\langle Lx | y\rangle = \langle x | L^* y\rangle$ for all $(x, y) \in \mathcal{H} \times \mathcal{G}$, denotes the *adjoint operator* of L.

2.2 Functions

Let $\mathcal{H}$ be a real Hilbert space.

Definition 2.2.1. *Consider the function $f : \mathcal{H} \to \overline{\mathbb{R}}$. The effective domain of f is*

$$\operatorname{dom} f = \{x \in \mathcal{H} : f(x) < +\infty\},$$

the graph of f is

$$\operatorname{gra} f = \{(x, \xi) \in \mathcal{H} \times \mathbb{R} : f(x) = \xi\},$$

the epigraph of f is

$$\operatorname{epi} f = \{(x,\xi) \in \mathcal{H} \times \mathbb{R} : f(x) \le \xi\},$$

the lower level set of f at height $\xi \in \mathbb{R}$ is

$$\operatorname*{lev}_{\le \xi} f = \{x \in \mathcal{H} : f(x) \le \xi\},$$

and the strict lower level set of f at height $\xi \in \mathbb{R}$ is

$$\operatorname*{lev}_{< \xi} f = \{x \in \mathcal{H} : f(x) < \xi\}.$$

The function f is called *proper* if $\operatorname{dom} f \neq \varnothing$ and $f(x) > -\infty$ for all $x \in \mathcal{H}$. Furthermore, the closures of $\operatorname{dom} f$ and $\operatorname{epi} f$ are respectively denoted by $\overline{\operatorname{dom}} f$ and $\overline{\operatorname{epi}} f$.

We denote by $\Gamma(\mathcal{H})$ the set of proper, convex and lower semicontinuous functions $f : \mathcal{H} \to \overline{\mathbb{R}}$.

The *indicator function* $\delta_C : \mathcal{H} \to \overline{\mathbb{R}}$ of a set $C \subseteq \mathcal{H}$ is defined by

$$\delta_C(x) = \begin{cases} 0, & \text{for } x \in C; \\ +\infty, & \text{otherwise.} \end{cases} \tag{2.2.1}$$

Note that the indicator function δ_C is lower semicontinuous if and only if C is closed.

Definition 2.2.2. *Let f and g be two proper functions from $\mathcal{H}$ to $\overline{\mathbb{R}}$. The infimal convolution (or epi-sum) of f and g is defined by*

$$f \,\square\, g : \mathcal{H} \to \overline{\mathbb{R}}, f \,\square\, g(x) = \inf_{y \in \mathcal{H}} \{f(y) + g(x-y)\}.$$

Definition 2.2.3. *Let $f : \mathcal{H} \to \overline{\mathbb{R}}$. The conjugate function (or Fenchel conjugate) of f is $f^* : \mathcal{H} \to \overline{\mathbb{R}}$, $f^*(u) = \sup\{\langle u, x\rangle - f(x) : x \in \mathcal{H}\}$ for all $u \in \mathcal{H}$ and the biconjugate of f is $f^{**} = (f^*)^*$.*

Note that, if $f \in \Gamma(\mathcal{H})$, then $f^* \in \Gamma(\mathcal{H})$, as well.

The following version of Fenchel's duality theorem was obtained by R. T. Rockafellar [53].

Theorem 2.2.1. *Let $\mathcal{H}$ be a Hilbert space. Let $f, g : \mathcal{H} \to \mathbb{R} \cup +\infty$ be proper convex functions.*

(i) If $\operatorname{dom} f \cap \operatorname{dom} g$ contains a point at which f or g is continuous, then

$$(f+g)^* = f^* \,\square\, g^*$$

with exact infimal convolution.

(ii) Suppose that f and g belong to $\Gamma(\mathcal{H})$. If $\operatorname{dom} f^* \cap \operatorname{dom} g^*$ contains a point at which f^* or g^* is continuous, then

$$f \,\square\, g = (f^* + g^*)^*$$

with exact infimal convolution.

Definition 2.2.4. *The primal problem associated with the sum of two proper functions $f : \mathcal{H} \to \overline{\mathbb{R}}$ and $g : \mathcal{H} \to \overline{\mathbb{R}}$ is*

$$\min_{x \in \mathcal{H}} \{f(x) + g(x)\},$$

its dual problem is

$$\min_{u \in \mathcal{H}} \{f^*(u) + g * (u)\},$$

the primal optimal value is $\mu = \inf(f+g)(\mathcal{H})$, the dual optimal value is $\mu^* = \inf(f^* + \breve{g}^*)(\mathcal{H})$, and the duality gap is

$$\Delta(f,g) = \begin{cases} 0, \ \ \mu = -\mu^* \in \overline{\mathbb{R}}; \\ \mu + \mu^*, \ \ \textit{otherwise}. \end{cases} \tag{2.2.2}$$

Chapter 3

The primal-dual splitting technique

The following results presented in th thesis are based on the article [10], that resulted from my cooperation with professor Radu Ioan Boţ and Ernő Robert Csetnek.

3.1 Monotone inclusion problem formulation

The problem under consideration is a more general system of monotone inclusions, than the one presented in [1], for which the coupling operator satisfies a Lipschitz continuity property, along with its dual system of monotone inclusions in an extended sense of the Attouch-Théra duality [2].

Problem 3.1.1. *Let $m \geq 1$ be a positive integer, $(\mathcal{H}_i)_{1\leq i\leq m}$ be real Hilbert spaces and for $i = 1, ..., m$ let $B_i : \mathcal{H}_1 \times \ldots \times \mathcal{H}_m \to \mathcal{H}_i$ be a μ_i-Lipschitz continuous operator with $\mu_i \in \mathbb{R}_{++}$ jointly satisfying the monotonicity property*

$$\forall (x_1, \ldots, x_m) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_m,\ \forall (y_1, \ldots, y_m) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_m$$
$$\sum_{i=1}^{m} \langle x_i - y_i | B_i(x_1, \ldots, x_m) - B_i(y_1, \ldots, y_m)\rangle_{\mathcal{H}_i} \geq 0. \tag{3.1.1}$$

For every $i = 1, \ldots, m$, let $\mathcal{G}_i$ be a real Hilbert space, $A_i : \mathcal{G}_i \to 2^{\mathcal{G}_i}$ a maximal monotone operator, $C_i : \mathcal{G}_i \to 2^{\mathcal{G}_i}$ a monotone operator such that C_i^{-1} is ν_i-Lipschitz continuous with $\nu_i \in \mathbb{R}_+$ and $L_i : \mathcal{H}_i \to \mathcal{G}_i$ a linear continuous operator. The problem is to solve *the system of coupled inclusions*

$$\text{find } \overline{x}_1 \in \mathcal{H}_1, \dots, \overline{x}_m \in \mathcal{H}_m \text{ such that } \begin{cases} 0 \in L_1^*(A_1 \square C_1)(L_1\overline{x}_1) + B_1(\overline{x}_1, \dots, \overline{x}_m) \\ \vdots \\ 0 \in L_m^*(A_m \square C_m)(L_m\overline{x}_m) + B_m(\overline{x}_1, \dots, \overline{x}_m) \end{cases} \tag{3.1.2}$$

together with its dual system

$$\begin{array}{r} \text{find } \overline{v}_1 \in \mathcal{G}_1, \dots, \overline{v}_m \in \mathcal{G}_m \text{ such that} \\ \exists x_1 \in \mathcal{H}_1, \dots, \exists x_m \in \mathcal{H}_m \end{array} \begin{cases} 0 = L_1^*\overline{v}_1 + B_1(x_1, \dots, x_m) \\ \vdots \\ 0 = L_m^*\overline{v}_m + B_m(x_1, \dots, x_m) \\ \overline{v}_1 \in (A_1 \square C_1)(L_1 x_1) \\ \vdots \\ \overline{v}_m \in (A_m \square C_m)(L_m x_m) \end{cases}. \tag{3.1.3}$$

We say that $(\overline{x}_1, \dots, \overline{x}_m, \overline{v}_1, \dots, \overline{v}_m) \in \mathcal{H}_1 \times \dots \times \mathcal{H}_m \times \mathcal{G}_1 \dots \times \mathcal{G}_m$ is a primal-dual solution to Problem 3.1.1, if

$$0 = L_i^*\overline{v}_i + B_i(\overline{x}_1, \dots, \overline{x}_m) \text{ and } \overline{v}_i \in (A_i \square C_i)(L_i\overline{x}_i),\ i = 1, \dots, m. \tag{3.1.4}$$

If $(\overline{x}_1, \dots, \overline{x}_m, \overline{v}_1, \dots, \overline{v}_m) \in \mathcal{H}_1 \times \dots \times \mathcal{H}_m \times \mathcal{G}_1 \dots \times \mathcal{G}_m$ is a primal-dual solution to Problem 3.1.1, then $(\overline{x}_1, \dots, \overline{x}_m)$ is a solution to (3.1.2) and $(\overline{v}_1, \dots, \overline{v}_m)$ is a solution to (3.1.3). Notice also that

$$(\overline{x}_1, \dots, \overline{x}_m) \text{ solves (3.1.2)} \Leftrightarrow 0 \in L_i^*(A_i \square C_i)(L_i\overline{x}_i) + B_i(\overline{x}_1, \dots, \overline{x}_m),\ i = 1, \dots, m \Leftrightarrow$$

$$\exists\, \overline{v}_1 \in \mathcal{G}_1, \dots, \overline{v}_m \in \mathcal{G}_m \text{ such that } \begin{cases} 0 = L_i^*\overline{v}_i + B_i(\overline{x}_1, \dots, \overline{x}_m),\ i = 1, \dots, m \\ \overline{v}_i \in (A_i \square C_i)(L_i\overline{x}_i),\ i = 1, \dots, m. \end{cases}.$$

Thus, if $(\overline{x}_1, \dots, \overline{x}_m)$ is a solution to (3.1.2), then there exists $(\overline{v}_1, \dots, \overline{v}_m) \in \mathcal{G}_1 \times \dots \mathcal{G}_m$ such that $(\overline{x}_1, \dots, \overline{x}_m, \overline{v}_1, \dots, \overline{v}_m)$ is a primal-dual solution to Problem 3.1.1 and, if $(\overline{v}_1, \dots, \overline{v}_m) \in \mathcal{G}_1 \times \dots \mathcal{G}_m$ is a solution to (3.1.3), then there exists $(\overline{x}_1, \dots, \overline{x}_m) \in \mathcal{H}_1 \times \dots \times \mathcal{H}_m$ such that $(\overline{x}_1, \dots, \overline{x}_m, \overline{v}_1, \dots, \overline{v}_m)$ is a primal-dual solution to Problem 3.1.1.

3.2 The primal-dual splitting algorithm

The aim of this section is to provide an algorithm for solving Problem 3.1.1 and to furnish weak and strong convergence results for the sequences generated by it. The proposed iterative scheme has the property that each single-valued operator is processed explicitly, while each set-valued operator is evaluated via its resolvent. Absolutely summable sequences make the algorithm error-tolerant.

Algorithm 3.2.1.

For every $i = 1, \ldots, m$ let $(a_{1,i,n})_{n\geq 0}$, $(b_{1,i,n})_{n\geq 0}$, $(c_{1,i,n})_{n\geq 0}$ be absolutely summable sequences in $\mathcal{H}_i$ and $(a_{2,i,n})_{n\geq 0}$, $(b_{2,i,n})_{n\geq 0}$, $(c_{2,i,n})_{n\geq 0}$ absolutely summable sequences in $\mathcal{G}_i$. Furthermore, set

$$\beta = max \left\{ \sqrt{\sum_{i=1}^{m} \mu_i^2}, \nu_1, \ldots, \nu_m \right\} + \max_{i=1,\ldots,m} \|L_i\|, \tag{3.2.1}$$

let $\varepsilon \in]0, 1/(\beta+1)[$ and $(\gamma_n)_{n\geq 0}$ be a sequence in $[\varepsilon, (1-\varepsilon)/\beta]$. For every $i = 1, \ldots, m$ let the initial points $x_{i,0} \in \mathcal{H}_i$ and $v_{i,0} \in \mathcal{G}_i$ be chosen arbitrary and set

$$\forall n \geq 0 \quad \left\lfloor \begin{array}{l} \text{For } i = 1, \ldots, m \\ \quad y_{i,n} = x_{i,n} - \gamma_n (L_i^* v_{i,n} + B_i(x_{1,n}, \ldots, x_{m,n}) + a_{1,i,n}) \\ \quad w_{i,n} = v_{i,n} - \gamma_n (C_i^{-1} v_{i,n} - L_i x_{i,n} + a_{2,i,n}) \\ \quad p_{i,n} = y_{i,n} + b_{1,i,n} \\ \quad r_{i,n} = J_{\gamma_n A_i^{-1}} w_{i,n} + b_{2,i,n} \\ \quad q_{i,n} = p_{i,n} - \gamma_n (L_i^* r_{i,n} + B_i(p_{1,n}, \ldots, p_{m,n}) + c_{1,i,n}) \\ \quad s_{i,n} = r_{i,n} - \gamma_n (C_i^{-1} r_{i,n} - L_i p_{i,n} + c_{2,i,n}) \\ \quad x_{i,n+1} = x_{i,n} - y_{i,n} + q_{i,n} \\ \quad v_{i,n+1} = v_{i,n} - w_{i,n} + s_{i,n}. \end{array} \right.$$

The convergence of Algorithm 3.2.1 is established by showing that its iterative scheme can be reduced to the error-tolerant version of the forward-backward-forward algorithm of Tseng (see [59] for the error-free case) recently provided in [16]. Our algorithmic framework will hinge on the following splitting algorithm, which is of interest in its own right.

Theorem 3.2.1. [16] *Let $\mathcal{H}$ be a real Hilbert space, let $M : \mathcal{H} \to 2^{\mathcal{H}}$ be maximal monotone, and let $L : \mathcal{H} \to \mathcal{H}$ be monotone. Suppose that $\operatorname{zer}(M+L) \neq \emptyset$ and that L is β-Lipschitz continuous for some $\beta \in]0,+\infty[$. Let $(a_n)_{n\in\mathbb{N}}$, $(b_n)_{n\in\mathbb{N}}$ and $(c_n)_{n\in\mathbb{N}}$ be sequences in $\mathcal{H}$ such that*

$$\sum_{n\in\mathbb{N}} \|a_n\| < +\infty, \ \sum_{n\in\mathbb{N}} \|b_n\| < +\infty \text{ and } \sum_{n\in\mathbb{N}} \|c_n\| < +\infty,$$

let $x_0 \in \mathcal{H}$, let $\varepsilon \in]0, 1/(\beta+1)[$, let $(\gamma_n)_{n\in\mathbb{N}}$ be a sequence in $[\varepsilon, (1-\varepsilon)/\beta]$, and set

$$\forall n \in \mathbb{N} \quad \left\lfloor \begin{array}{l} y_n = x_n - \gamma_n(Lx_n + a_n) \\ p_n = J_{\gamma_n M} y_n + b_n \\ q_n = p_n - \gamma_n(Lp_n) + c_n) \\ x_{n+1} = x_n - y_n + q_n. \end{array} \right.$$

Then the following hold for some $\overline{x} \in \operatorname{zer}(M+L)$:

(i) $\sum_{n\in\mathbb{N}} \|x_n - p_n\|^2 < +\infty$ *and* $\sum_{n\in\mathbb{N}} \|y_n - q_n\|^2 < +\infty$.

(ii) $x_n \rightharpoonup \overline{x}$ *and* $p_n \rightharpoonup \overline{x}$.

(iii) Suppose that one of the following is satisfied:

(a) $M + L$ *is demiregular at* $\overline{x}$.

(b) M *or* L *is uniformly monotone at* $\overline{x}$.

(c) $\operatorname{int} \operatorname{zer}(M+L) \neq \emptyset$.

Then $x_n \to \overline{x}$ *and* $p_n \to \overline{x}$.

For a more detailed insight on the problem of simultaneously solving a large class of composite monotone inclusions and their duals, while reducing it to the problem of finding a zero of the sum of a maximal monotone operator and a linear skew-adjoint operator, we recommend the work of Luis M. Briceño-Arias and Patrick L. Combettes in [16].

The following theorem establishes the convergence of Algorithm 3.2.1:

Theorem 3.2.2. [10] *Suppose that Problem 3.1.1 has a primal-dual solution. For the sequences generated by Algorithm 3.2.1 the following statements are true:*

(i) $\forall i \in \{1,\dots,m\}$ $\sum_{n\geq 0} \|x_{i,n} - p_{i,n}\|^2_{\mathcal{H}_i} < +\infty$ *and* $\sum_{n\geq 0} \|v_{i,n} - r_{i,n}\|^2_{\mathcal{G}_i} < +\infty$.

(ii) There exists a primal-dual solution $(\overline{x}_1,\dots,\overline{x}_m,\overline{v}_1,\dots,\overline{v}_m)$ *to Problem 3.1.1 such that:*

(a) $\forall i \in \{1,\dots,m\}$ $x_{i,n} \rightharpoonup \overline{x}_i$, $p_{i,n} \rightharpoonup \overline{x}_i$, $v_{i,n} \rightharpoonup \overline{v}_i$ *and* $r_{i,n} \rightharpoonup \overline{v}_i$ *as* $n \to +\infty$.

(b) if $C_i^{-1}, i = 1,...,m$, *is uniformly monotone and there exists an increasing function* $\phi_B : \mathbb{R}_+ \to \mathbb{R}_+ \cup \{+\infty\}$ *vanishing only at 0 and fulfilling*

$$\begin{aligned}
&\forall (x_1,\dots,x_m) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_m,\ \forall (y_1,\dots,y_m) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_m\\
&\sum_{i=1}^m \langle x_i - y_i | B_i(x_1,\dots,x_m) - B_i(y_1,\dots,y_m)\rangle_{\mathcal{H}_i} \geq\\
&\phi_B\Big(\|(x_1,\dots,x_m) - (y_1,\dots,y_m)\| \Big),
\end{aligned} \tag{3.2.2}$$

then $\forall i \in \{1,\dots,m\}$ $x_{i,n} \to \overline{x}_i$, $p_{i,n} \to \overline{x}_i$, $v_{i,n} \to \overline{v}_i$ *and* $r_{i,n} \to \overline{v}_i$ *as* $n \to +\infty$.

3.3 Applications to convex minimization problems

In this section we turn our attention to the solving of convex minimization problems with multiple variables via the primal-dual algorithm presented and investigated in this thesis.

Problem 3.3.1. *Let* $m \geq 1$ *and* $p \geq 1$ *be positive integers,* $(\mathcal{H}_i)_{1\leq i\leq m}$, $(\mathcal{H}_i^{'})_{1\leq i\leq m}$ *and* $(\mathcal{G}_j)_{1\leq j\leq p}$ *be real Hilbert spaces,* $f_i, h_i \in \Gamma(\mathcal{H}_i^{'})$ *such that* h_i *is* ν_i^{-1}*-strongly convex with* $\nu_i \in \mathbb{R}_{++}, i = 1,...,m$, *and* $g_j \in \Gamma(\mathcal{G}_j)$ *for* $i = 1,...,m, j = 1,...,p$. *Further, let be* $K_i : \mathcal{H}_i \to \mathcal{H}_i^{'}$ *and* $L_{ji} : \mathcal{H}_i \to \mathcal{G}_j, i = 1,...,m, j = 1,...,p$ *linear continuous operators. Consider the convex optimization problem*

$$\inf_{(x_1,\dots,x_m)\in\mathcal{H}_1\times\ldots\times\mathcal{H}_m} \left\{ \sum_{i=1}^m (f_i \,\Box\, h_i)(K_i x_i) + \sum_{j=1}^p g_j \left(\sum_{i=1}^m L_{ji} x_i \right) \right\}. \tag{3.3.1}$$

In what follows we show that under an appropriate qualification condition solving the convex optimization problem (3.3.1) can be reduced to the solving of a system of monotone inclusions of type (3.1.2).

Let us define the following proper convex and lower semicontinuous function

$$f : \mathcal{H}_1^{'} \times \ldots \times \mathcal{H}_m^{'} \rightarrow \overline{\mathbb{R}}, \quad (y_1, \ldots, y_m) \mapsto \sum_{i=1}^{m} (f_i \Box h_i)(y_i),$$

and the linear continuous operator

$$K : \mathcal{H}_1 \times \ldots \times \mathcal{H}_m \rightarrow \mathcal{H}_1^{'} \times \ldots \times \mathcal{H}_m^{'}, \quad (x_1, \ldots, x_m) \mapsto (K_1 x_1, \ldots, K_m x_m),$$

having as adjoint

$$K^* : \mathcal{H}_1^{'} \times \ldots \times \mathcal{H}_m^{'} \rightarrow \mathcal{H}_1 \times \ldots \times \mathcal{H}_m, \quad (y_1, \ldots, y_m) \mapsto (K_1^* y_1, \ldots, K_m^* y_m).$$

Further, consider the linear continuous operators

$$L_j : \mathcal{H}_1 \times \ldots \times \mathcal{H}_m \rightarrow \mathcal{G}_j, \quad (x_1, \ldots, x_m) \mapsto \sum_{i=1}^{m} L_{ji} x_i, j = 1, ..., p,$$

having as adjoints

$$L_j^* : \mathcal{G}_j \rightarrow \mathcal{H}_1 \times \ldots \times \mathcal{H}_m, \quad y \mapsto (L_{j1}^* y, \ldots, L_{jm}^* y), j = 1, ..., p,$$

respectively. We have

$$\begin{aligned} & (\overline{x}_1, \ldots, \overline{x}_m) \text{ is an optimal solution to (3.3.1)} \\ \Leftrightarrow\ & (0, \ldots, 0) \in \partial \left(f \circ K + \sum_{j=1}^{p} g_j \circ L_j \right) (\overline{x}_1, \ldots \overline{x}_m). \end{aligned} \tag{3.3.2}$$

In order to split the above subdifferential in a sum of subdifferentials a so-called *qualification condition* must be fulfilled. In this context, we consider the following *interiority-type* qualification conditions:

$$(\mathrm{QC}_1)\left|\begin{array}{l}\text{there exists } x_i' \in \mathcal{H}_i \text{ such that}\\ K_i x_i' \in (\operatorname{dom} f_i + \operatorname{dom} h_i) \text{ and } f_i \square h_i \text{ is continuous at } K_i x_i', i = 1, ..., m,\\ \text{and } \sum_{i=1}^m L_{ji} x_i' \in \operatorname{dom} g_j \text{ and } g_j \text{ is continuous at } \sum_{i=1}^m L_{ji} x_i', j = 1, ..., p\end{array}\right.$$

and

$$(\mathrm{QC}_2)\left|\begin{array}{l}(0,\dots,0) \in \ \operatorname{sqri}\Big(\prod_{i=1}^m (\operatorname{dom} f_i + \operatorname{dom} h_i) \times \prod_{j=1}^p \operatorname{dom} g_j\\ -\{(K_1 x_1, \dots, K_m x_m, \sum_{i=1}^m L_{1i} x_i, \dots, \sum_{i=1}^m L_{pi} x_i) :\\ (x_1, \dots, x_m) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_m\}\Big).\end{array}\right.$$

We notice that $(QC_1) \Rightarrow (QC_2)$, these implications being in general strict, and refer the reader to [4, 7, 8, 29, 58, 68, 69] and the references therein for other qualification conditions in convex optimization.

Remark 3.3.1. *As already pointed out, for* $i = 1, ..., m$, $f_i \square h_i \in \Gamma(\mathcal{H}_i')$, *hence, it is continuous on* $\operatorname{int}(\operatorname{dom} f_i + \operatorname{dom} h_i)$, *providing this set is nonempty (see [29, 69]). For other results regarding the continuity of the infimal convolution of convex functions we invite the reader to consult [57].*

Remark 3.3.2. *In finite-dimensional spaces the qualification condition* (QC_2) *is equivalent to*

$$(QC_2)\left|\begin{array}{l}\textit{there exists } x_i' \in \mathcal{H}_i \textit{ such that } K_i x_i' \in \operatorname{ri}\operatorname{dom} f_i + \operatorname{ri}\operatorname{dom} h_i, i = 1, ..., m,\\ \textit{and } \sum_{i=1}^m L_{ji} x_i' \in \operatorname{ri}\operatorname{dom} g_j, j = 1, ..., p.\end{array}\right.$$

Assuming that one of the qualification conditions above is fulfilled, we have that

$$\begin{aligned}
&(\overline{x}_1, \dots, \overline{x}_m) \text{ is an optimal solution to } (3.3.1)\\
\Leftrightarrow\ &(0, \dots, 0) \in K^* \partial f\big(K(\overline{x}_1, \dots \overline{x}_m)\big) + \sum_{j=1}^p L_j^* \partial g_j\big(L_j(\overline{x}_1, \dots \overline{x}_m)\big)\\
\Leftrightarrow\ &(0, \dots, 0) \in \Big(K_1^* \partial (f_1 \square h_1)(K_1 \overline{x}_1), \dots, K_m^* \partial (f_m \square h_m)(K_m \overline{x}_m)\Big)\\
&\qquad + \sum_{j=1}^p L_j^* \partial g_j\big(L_j(\overline{x}_1, \dots \overline{x}_m)\big). \qquad (3.3.3)
\end{aligned}$$

The strong convexity of the functions h_i imply that $\operatorname{dom} h_i^* = \mathcal{H}_i'$ and so $\partial(f_i \Box h_i) = \partial f_i \Box \partial h_i, i = 1, ..., m$. Thus, (3.3.3) is further equivalent to

$$(0, \ldots, 0) \in \Big(K_1^*(\partial f_1 \Box \partial h_1)(K_1\overline{x}_1), \ldots, K_m^*(\partial f_m \Box \partial h_m)(K_m\overline{x}_m)\Big) + \sum_{j=1}^{p} L_j^* v_j,$$

where

$$\overline{v}_j \in \partial g_j\big(L_j(\overline{x}_1, \ldots \overline{x}_m)\big) \;\Leftrightarrow\; \overline{v}_j \in \partial g_j\Big(\sum_{i=1}^{m} L_{ji}\overline{x}_i\Big) \;\Leftrightarrow\; \sum_{i=1}^{m} L_{ji}\overline{x}_i \in \partial g_j^*(\overline{v}_j), j = 1, ..., p.$$

Then $(\overline{x}_1, \ldots, \overline{x}_m)$ is an optimal solution to (3.3.1) if and only if $(\overline{x}_1, \ldots, \overline{x}_m, \overline{v}_1, \ldots, \overline{v}_p)$ is a solution to

$$\begin{cases} 0 \in K_1^*(\partial f_1 \Box \partial h_1)(K_1\overline{x}_1) + \sum_{j=1}^{p} L_{j1}^*\overline{v}_j \\ \vdots \\ 0 \in K_m^*(\partial f_m \Box \partial h_m)(K_m\overline{x}_m) + \sum_{j=1}^{p} L_{jm}^*\overline{v}_j \\ 0 \in \partial g_1^*(\overline{v}_1) - \sum_{i=1}^{m} L_{1i}\overline{x}_i \\ \vdots \\ 0 \in \partial g_p^*(\overline{v}_p) - \sum_{i=1}^{m} L_{pi}\overline{x}_i. \end{cases} \tag{3.3.4}$$

One can see now that (3.3.4) is a system of coupled inclusions of type (3.1.2), by taking

$$A_i = \partial f_i,\; C_i = \partial h_i,\; L_i = K_i, i = 1, ..., m,$$

$$A_{m+j} = \partial g_j^*,\; C_{m+j}(x) = \begin{cases} \mathcal{G}_j, & x = 0 \\ \emptyset, & \text{otherwise} \end{cases},$$

$$L_{m+j} = \mathrm{Id}_{\mathcal{G}_j}, j = 1, ..., p,$$

and, for $(x_1, ..., x_m, v_1, ..., v_p) \in \mathcal{H}_1 \times \ldots \mathcal{H}_m \times \mathcal{G}_1 \times \ldots \times \mathcal{G}_p$, as coupling operators

$$B_i(x_1, \ldots, x_m, v_1, \ldots, v_p) = \sum_{j=1}^{p} L_{ji}^* v_j, i = 1, \ldots, m,$$

and

$$B_{m+j}(x_1,\dots,x_m,v_1,\dots,v_p) = -\sum_{i=1}^{m} L_{ji}x_i, j=1,\dots,p.$$

Define

$$\begin{aligned} B(x_1,\dots,x_m,v_1,\dots,v_p) &= (B_1,\dots,B_{m+p})(x_1,\dots,x_m,v_1,\dots,v_p) & (3.3.5)\\ &= \left(\sum_{j=1}^{p} L_{j1}^* v_j,\dots,\sum_{j=1}^{p} L_{jm}^* v_j, -\sum_{i=1}^{m} L_{1i}x_i,\dots,-\sum_{i=1}^{m} L_{pi}x_i\right). & (3.3.6)\end{aligned}$$

It follows that $C_i^{-1} = (\partial h_i)^{-1} = \partial h_i^* = \{\nabla h_i^*\}$ is ν_i-Lipschitz continuous for $i=1,...,m$. On the other hand, C_{m+j}^{-1} is the zero operator for $j=1,...,p$, thus 0-Lipschitz continuous.

Furthermore, the operators $B_i, i=1,...,m+p$ are linear and Lipschitz continuous, having as Lipschitz constants

$$\mu_i = \sqrt{\sum_{j=1}^{p} \|L_{ji}\|^2},\ i=1,...,m,\ \text{and}\ \mu_{m+j} = \sqrt{\sum_{i=1}^{m} \|L_{ji}\|^2},\ j=1,...,p,$$

respectively. For every $(x_1,\dots,x_m,v_1,\dots,v_p),(y_1,\dots,y_m,w_1,\dots,w_p) \in \mathcal{H}_1\times\ldots\times\mathcal{H}_m\times\mathcal{G}_1\times\ldots\times\mathcal{G}_p$ it holds

$$\begin{aligned}&\sum_{i=1}^{m}\langle x_i-y_i|B_i(x_1,\dots,x_m,v_1,\dots,v_p)-B_i(y_1,\dots,y_m,w_1,\dots,w_p)\rangle_{\mathcal{H}_i}\\ &+\sum_{j=1}^{p}\langle v_j-w_j|B_{m+j}(x_1,\dots,x_m,v_1,\dots,v_p)-B_{m+j}(y_1,\dots,y_m,w_1,\dots,w_p)\rangle_{\mathcal{G}_j}\\ &=\sum_{i=1}^{m}\left\langle x_i-y_i|\sum_{j=1}^{p}L_{ji}^*v_j-\sum_{j=1}^{p}L_{ji}^*w_j\right\rangle_{\mathcal{H}_i}-\sum_{j=1}^{p}\left\langle v_j-w_j|\sum_{i=1}^{m}L_{ji}x_i-\sum_{i=1}^{m}L_{ji}y_i\right\rangle_{\mathcal{G}_j}=0,\end{aligned}$$

thus (3.1.1) is fulfilled. This proves also that the linear continuous operator B is skew (i.e. $B^*=-B$).

Remark 3.3.3. *Due to the fact that the operator B is skew, it is not cocoercive, hence,* *the approach presented in [1] cannot be applied in this context. On the other hand, in the light of the characterization given in* (3.3.2)*, in order to determine an optimal solution of the optimization problem* (3.3.1) *(and an optimal solution of its Fenchel-type dual as well) one can use the primal-dual proximal splitting algorithms which have been recently introduced in [23, 61]. These approaches have the particularity to deal in an efficient way with sums of compositions of proper, convex and lower semicontinuous function with linear continuous operators, by evaluating separately each function via a backward step and each linear continuous operator (and its adjoint) via a forward step. However, the iterative scheme we propose in this section for solving* (3.3.1) *has the advantage of exploiting the separable structure of the problem.*

Let us also mention that the dual inclusion problem of (3.3.4) reads (see (3.1.3))

$$\begin{array}{l}\text{find } \overline{w}_1 \in \mathcal{H}_1', \dots, \overline{w}_m \in \mathcal{H}_m',\\ \overline{w}_{m+1} \in \mathcal{G}_1, \dots, \overline{w}_{m+p} \in \mathcal{G}_p \text{ such that}\\ \exists x_1 \in \mathcal{H}_1, \dots, \exists x_m \in \mathcal{H}_m, \exists v_1 \in \mathcal{G}_1, \dots, \exists v_p \in \mathcal{G}_p\end{array} \begin{cases} 0 = K_1^*\overline{w}_1 + \sum_{j=1}^p L_{j1}^* v_j \\ \vdots \\ 0 = K_m^*\overline{w}_m + \sum_{j=1}^p L_{jm}^* v_j \\ 0 = \overline{w}_{m+1} - \sum_{i=1}^m L_{1i} x_i \\ \vdots \\ 0 = \overline{w}_{m+p} - \sum_{i=1}^m L_{pi} x_i \\ \overline{w}_1 \in (\partial f_1 \Box \partial h_1)(K_1 x_1) \\ \vdots \\ \overline{w}_m \in (\partial f_m \Box \partial h_m)(K_m x_m) \\ \overline{w}_{m+1} \in \partial g_1^*(v_1) \\ \vdots \\ \overline{w}_{m+p} \in \partial g_p^*(v_p). \end{cases} \tag{3.3.7}$$

Then $(\overline{x}_1, ...\overline{x}_m, \overline{v}_1, ..., \overline{v}_p, \overline{w}_1, ..., \overline{w}_m, \overline{w}_{m+1}, ..., \overline{w}_{m+p})$ is a primal-dual solution to (3.3.4) - (3.3.7), if

$$\overline{w}_i \in (\partial f_i \Box \partial h_i)(K_i \overline{x}_i), \overline{w}_{m+j} \in \partial g_j^*(\overline{v}_j),$$
$$0 = K_i^* \overline{w}_i + \sum_{j=1}^p L_{ji}^* \overline{v}_j \text{ and } 0 = \overline{w}_{m+j} - \sum_{i=1}^m L_{ji} \overline{x}_i, i = 1, ..., m, j = 1, ..., p.$$

Provided that $(\overline{x}_1, ...\overline{x}_m, \overline{v}_1, ..., \overline{v}_p, \overline{w}_1, ..., \overline{w}_m, \overline{w}_{m+1}, ..., \overline{w}_{m+p})$ is a primal-dual solution to (3.3.4) - (3.3.7), it follows that $(\overline{x}_1, ...\overline{x}_m)$ is an optimal solution to (3.3.1) and $(\overline{w}_1, ..., \overline{w}_m, \overline{v}_1, ..., \overline{v}_p)$ is an optimal solution to its *Fenchel-type* dual problem

$$\sup_{\substack{(w_1,...,w_m,w_{m+1},...,w_{m+p})\in\mathcal{H}_1'\times...\times\mathcal{H}_m'\times\mathcal{G}_1\times...\times\mathcal{G}_p \\ K_i^* w_i+\sum_{j=1}^p L_{ji}^* w_{m+j}=0, i=1,...,m}} \left\{ -\sum_{i=1}^m \left(f_i^*(w_i) + h_i^*(w_i)\right) - \sum_{j=1}^p g_j^*(w_{m+j}) \right\}. \tag{3.3.8}$$

Algorithm 3.2.1 gives rise to the following iterative scheme for solving (3.3.4) - (3.3.7).

Algorithm 3.3.1.

For every $i = 1, \ldots, m$ *and every* $j = 1, \ldots, p$ *let* $(a_{1,i,n})_{n\geq 0}$, $(b_{1,i,n})_{n\geq 0}$, $(c_{1,i,n})_{n\geq 0}$, *be absolutely summable sequences in* $\mathcal{H}_i$, $(a_{2,i,n})_{n\geq 0}$, $(b_{2,i,n})_{n\geq 0}$, $(c_{2,i,n})_{n\geq 0}$ *be absolutely summable sequences in* $\mathcal{H}_i'$ *and* $(a_{1,m+j,n})_{n\geq 0}$, $(a_{2,m+j,n})_{n\geq 0}$, $(b_{1,m+j,n})_{n\geq 0}$, $(b_{2,m+j,n})_{n\geq 0}$, $(c_{1,m+j,n})_{n\geq 0}$ *and* $(c_{2,m+j,n})_{n\geq 0}$ *be absolutely summable sequences in* $\mathcal{G}_j$. *Furthermore, set*

$$\beta = max \left\{ \sqrt{\sum_{i=1}^{m+p} \mu_i^2}, \nu_1, \ldots, \nu_m \right\} + \; max\; \{\|K_1\|, \ldots, \|K_m\|, 1\}, \tag{3.3.9}$$

where

$$\mu_i = \sqrt{\sum_{j=1}^p \|L_{ji}\|^2},\; i = 1, \ldots, m,\; and\; \mu_{m+j} = \sqrt{\sum_{i=1}^m \|L_{ji}\|^2},\; j = 1, \ldots, p, \tag{3.3.10}$$

let $\varepsilon \in]0, 1/(\beta+1)[$ *and* $(\gamma_n)_{n\geq 0}$ *be a sequence in* $[\varepsilon, (1-\varepsilon)/\beta]$. *Let the initial points* $(x_{1,1,0}, \ldots, x_{1,m,0}) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_m$, $(x_{2,1,0}, \ldots, x_{2,m,0}) \in \mathcal{H}_1' \times \ldots \times \mathcal{H}_m'$ *and*

$(v_{1,1,0},\dots,v_{1,p,0}),\ (v_{2,1,0},\dots,v_{2,p,0}) \in \mathcal{G}_1 \times \ldots \times \mathcal{G}_p$ *be arbitrary chosen and set*

$$
\forall n \geq 0 \quad \left\lfloor \begin{array}{l}
\text{For } i = 1,\dots,m \\
\left\lfloor \begin{array}{l}
y_{1,i,n} = x_{1,i,n} - \gamma_n \left(K_i^* x_{2,i,n} + \sum_{j=1}^{p} L_{ji}^* v_{1,j,n} + a_{1,i,n} \right) \\
y_{2,i,n} = x_{2,i,n} - \gamma_n (\nabla h_i^* x_{2,i,n} - K_i x_{1,i,n} + a_{2,i,n}) \\
p_{1,i,n} = y_{1,i,n} + b_{1,i,n} \\
p_{2,i,n} = \operatorname{Prox}_{\gamma_n f_i^*} y_{2,i,n} + b_{2,i,n}
\end{array} \right. \\
\text{For } j = 1,\dots,p \\
\left\lfloor \begin{array}{l}
w_{1,j,n} = v_{1,j,n} - \gamma_n \left(v_{2,j,n} - \sum_{i=1}^{m} L_{ji} x_{1,i,n} + a_{1,m+j,n} \right) \\
w_{2,j,n} = v_{2,j,n} - \gamma_n (-v_{1,j,n} + a_{2,m+j,n}) \\
r_{1,j,n} = w_{1,j,n} + b_{1,m+j,n} \\
r_{2,j,n} = \operatorname{Prox}_{\gamma_n g_j} w_{2,j,n} + b_{2,m+j,n}
\end{array} \right. \\
\text{For } i = 1,\dots,m \\
\left\lfloor \begin{array}{l}
q_{1,i,n} = p_{1,i,n} - \gamma_n \left(K_i^* p_{2,i,n} + \sum_{j=1}^{p} L_{ji}^* r_{1,j,n} + c_{1,i,n} \right) \\
q_{2,i,n} = p_{2,i,n} - \gamma_n (\nabla h_i^* p_{2,i,n} - K_i p_{1,i,n} + c_{2,i,n}) \\
x_{1,i,n+1} = x_{1,i,n} - y_{1,i,n} + q_{1,i,n} \\
x_{2,i,n+1} = x_{2,i,n} - y_{2,i,n} + q_{2,i,n}
\end{array} \right. \\
\text{For } j = 1,\dots,p \\
\left\lfloor \begin{array}{l}
s_{1,j,n} = r_{1,j,n} - \gamma_n \left(r_{2,j,n} - \sum_{i=1}^{m} L_{ji} p_{1,i,n} + c_{1,m+j,n} \right) \\
s_{2,j,n} = r_{2,j,n} - \gamma_n (-r_{1,j,n} + c_{2,m+j,n}) \\
v_{1,j,n+1} = v_{1,j,n} - w_{1,j,n} + s_{1,j,n} \\
v_{2,j,n+1} = v_{2,j,n} - w_{2,j,n} + s_{2,j,n}.
\end{array} \right.
\end{array} \right.
$$

The following convergence result for Algorithm 3.3.1 is a consequence of Theorem 3.2.2.

Theorem 3.3.1. [10] *Suppose that the optimization problem* (3.3.1) *has an optimal solution and that one of the qualification conditions* (QC_i), $i = 1, 2$, *is fulfilled. For the sequences generated by Algorithm 3.3.1 the following statements are true:*

(i) $\forall i \in \{1,\dots,m\}$ $\sum_{n\geq 0} \|x_{1,i,n} - p_{1,i,n}\|^2_{\mathcal{H}_i} < +\infty$, $\sum_{n\geq 0} \|x_{2,i,n} - p_{2,i,n}\|^2_{\mathcal{H}_i} < +\infty$ *and* $\forall j \in \{1,\dots,p\}$ $\sum_{n\geq 0} \|v_{1,j,n} - r_{1,j,n}\|^2_{\mathcal{G}_j} < +\infty$, $\sum_{n\geq 0} \|v_{2,j,n} - r_{2,j,n}\|^2_{\mathcal{G}_j} < +\infty$.

(ii) There exists an optimal solution $(\overline{x}_1,\dots,\overline{x}_m)$ *to* (3.3.1) *and an optimal solution*

$(\overline{w}_1, \ldots, \overline{w}_m, \overline{w}_{m+1}, \ldots, \overline{w}_{m+p})$ *to* (3.3.8)*, such that* $\forall i \in \{1, \ldots, m\}$ $x_{1,i,n} \rightharpoonup \overline{x}_i$, $p_{1,i,n} \rightharpoonup \overline{x}_i$, $x_{2,i,n} \rightharpoonup \overline{w}_i$ *and* $p_{2,i,n} \rightharpoonup \overline{w}_i$ *and* $\forall j \in \{1, \ldots, p\}$ $v_{1,j,n} \rightharpoonup \overline{w}_{m+j}$ *and* $r_{1,j,n} \rightharpoonup \overline{w}_{m+j}$ *as* $n \rightarrow +\infty$.

Remark 3.3.4. *Recently, in [22], another iterative scheme for solving systems of monotone inclusions, that is also able to handle with the solving of optimization problems of type* (3.3.1)*, in case when the functions* $g_j, j = 1, ..., p$*, are not necessarily differentiable, was proposed. Different to our approach, which assumes that the variables are coupled by the single-valued operator* B*, in [22] the coupling is made by some compositions of parallel sums of maximal monotone operators with linear continuous ones.*

Chapter 4

Numerical experiments

The obtained theoretical results are highly applicable in various fields of mathematics. In this section we present four numerical experiments which emphasize the performances of the primal-dual algorithm and some of its variants for systems of coupled monotone inclusions, namely image processing, multifacility location problems, average consensus for colored networks and image classification via support-vector machines.

4.1 Image processing

The first numerical experiment solves the problem of image processing via the primal-dual splitting algorithm developed in this thesis. The aim is to estimate the unknown original image form the blurred and noisy image, with the help of the algorithm.

4.1.1 Image deblurring

Let $\mathcal{H}$ and $(\mathcal{H}_i')_{1\leq i\leq m}$ be real Hilbert spaces, where $m \geq 1$. For every $i \in \{1,\ldots,m\}$, let $f_i \in \Gamma(\mathcal{H}_i')$ and let $K_i : \mathcal{H} \to \mathcal{H}_i'$ be continuous linear operators. Consider the initial convex optimization problem

$$\inf_{x\in\mathcal{H}} \Big\{ \sum_{i=1}^{m} f_i(K_i x) \Big\}. \tag{4.1.1}$$

In the case of multiple variables, (4.1.1) can be reformulated as:

$$\inf_{x_1,\dots,x_{m+1}} \Big\{ \sum_{i=1}^{m} f_i(K_i x_i) + \sum_{i=1}^{m} \delta_{\{0\}}(x_i - x_{m+1}) \Big\}. \tag{4.1.2}$$

This is a special instance of Problem 3.3.1 with $(m,p) := (m+1,m)$, $f_{m+1} = 0$, $K_{m+1} = 0$ and $\forall i \in \{1,\dots,m+1\}$, $h_i = \delta_{\{0\}}$ and $\forall j \in \{1,\dots,m\}$, $\mathcal{G}_j = \mathcal{H}$, $g_j = \delta_{\{0\}}$ and

$$L_{ji} : \mathcal{H}_i \to \mathcal{H}_j \mapsto \begin{cases} \mathrm{Id}, \ \text{if } i = j; \\ -\mathrm{Id}, \ \text{if } i = m+1; \\ 0, \ \text{otherwise.} \end{cases} \tag{4.1.3}$$

Let $(x_{1,1,0},\dots,x_{1,m+1,0})$, $(x_{2,1,0},\dots,x_{2,m+1,0}) \in \mathcal{H}_1 \times \dots \times \mathcal{H}_{m+1}$ and $(v_{1,1,0},\dots,v_{1,m,0})$, $(v_{2,1,0},\dots,v_{2,m,0}) \in \mathcal{H}_1 \times \dots \times \mathcal{H}_m$. In this case (3.3.9) yields

$$\beta = 2\sqrt{m} + \max \Big\{ \|K_1\|, \dots, \|K_{m+1}\|, 1 \Big\}. \tag{4.1.4}$$

Algorithm 4.1.1.

Hence, the iterative scheme in Algorithm 3.3.1 reads

$$\forall n \geq 0 \quad \left\lfloor \begin{array}{l} \textit{For } i = 1,\dots,m \\ \left\lfloor \begin{array}{l} y_{1,i,n} = x_{1,i,n} - \gamma_n \big(K_i^* x_{2,i,n} + v_{1,i,n}\big) \\ w_{1,i,n} = v_{1,i,n} - \gamma_n \big(v_{2,i,n} - x_{1,i,n} + x_{1,m+1,n}\big) \\ y_{2,i,n} = x_{2,i,n} + \gamma_n K_i x_{1,i,n} \\ w_{2,i,n} = v_{2,i,n} + \gamma_n v_{1,i,n} \end{array} \right. \\ y_{1,m+1,n} = x_{1,m+1,n} + \gamma_n \sum_{i=1}^{m} v_{1,i,n} \\ \textit{For } i = 1,\dots,m \\ \left\lfloor \begin{array}{l} x_{1,i,n+1} = x_{1,i,n} - \gamma_n \big(K_i^* prox_{\gamma_n f_i^*} y_{2,i,n} + w_{1,i,n}\big) \\ v_{1,i,n+1} = v_{1,i,n} + \gamma_n \big(y_{1,i,n} - y_{1,m+1,n}\big) \\ x_{2,i,n+1} = x_{2,i,n} - y_{2,i,n} + prox_{\gamma_n f_i^*} y_{2,i,n} + \gamma_n K_i y_{1,i,n} \\ v_{2,i,n+1} = v_{2,i,n} - w_{2,i,n} + \gamma_n w_{1,i,n} \end{array} \right. \\ x_{1,m+1,n+1} = x_{1,m+1,n} + \gamma_n \sum_{i=1}^{m} w_{1,i,n} \end{array} \right.$$

Then $(x_{1,1,n}, \dots, x_{1,m+1,n}) \to (\overline{x}, \dots, \overline{x})$.

Consider a matrix $A \in \mathbb{R}^{n\times n}$ representing the *blur operator* and a given vector $b \in \mathbb{R}^n$ describing the *blurred and noisy image*. The goal is to estimate $x^* \in \mathbb{R}^n$, which is the *unknown original image* and fulfills the following

$$Ax = b.$$

The problem to be solved can be equivalently written as

$$\inf_{x\in S} \left\{ f_1(x) + f_2(Ax) \right\}, \tag{4.1.5}$$

for $f_1 : \mathbb{R}^n \to \overline{\mathbb{R}}$, $f_1(x) = \lambda \|x\|_1 + \delta_S(x)$ and $f_2 : \mathbb{R}^n \to \mathbb{R}$, $f_2(y) = \|y - b\|^2$. Thus f is proper, convex and lower semicontinuous with bounded domain and g is a 2-strongly convex function with full domain, differentiable everywhere and with Lipschitz continuous gradient having as Lipschitz constant 2. $\lambda > 0$ represents the regularization parameter and $S \subseteq \mathbb{R}^n$ is an n-dimensional cube representing the range of the pixels. Since each pixel furnishes a greyscale value which is between 0 and 255, a natural choice for the convex set S would be the n-dimensional cube $[0, 255]^n \subseteq \mathbb{R}^n$. In order to reduce the Lipschitz constant which appears in the developed approach, we scale the pictures that we use to exemplify the application, such that each of their pixels ranges in the interval $\left[0, \frac{1}{10}\right]$.

4.1.2 Experimental setting

We concretely look at the 256×256 *Cameraman, Lena and Barbara test images.*

The *Cameraman test image* is part of the image processing toolbox in Matlab and its vectorized and scaled image dimension equals $n = 256^2 = 65536$. By making use of the Matlab functions `imfilter` and `fspecial`, this image is blurred as follows:

```
1 H=fspecial('gaussian',9,4);    % gaussian blur of size 9 times 9
2                                % and standard deviation 4
3 B=imfilter(X,H,'conv','symmetric');  % B=observed blurred image
4                                      % X=original image
```

In row 1 the function `fspecial` returns a rotationally symmetric Gaussian lowpass filter of size 9×9 with standard deviation 4. The entries of H are nonnegative and their sum adds up to 1. In row 3 the function `imfilter` convolves the filter H with the image $X \in \mathbb{R}^{256\times 256}$ and outputs the blurred image $B \in \mathbb{R}^{256\times 256}$. The boundary option "symmetric" corresponds to reflexive boundary conditions. For more interesting applications concerning Matlab test images deblurred with different techniques, we refer the reader to [11–15, 19].

Thanks to the rotationally symmetric filter H, the linear operator $A \in \mathbb{R}^{n\times n}$ given by the Matlab function `imfilter` is symmetric, too. After adding a zero-mean white Gaussian noise with standard deviation 10^{-4}, we obtain the blurred and noisy image $b \in \mathbb{R}^n$.

In this particular case $m := 2$, $K_1 = \mathrm{Id}$, $K_2 = A$ and β equals

$$\beta = 2\sqrt{2} + \max\left\{ \|A\|, 1 \right\}. \tag{4.1.6}$$

By making use of the real spectral decomposition of A, it shows that $\|A\|^2 = 1$. Hence $\beta = 2\sqrt{2} + 1$.

Algorithm 4.1.2.

Algorithm 4.1.1 can be written as

$$\forall n \geq 0 \quad \left\lfloor \begin{array}{l} y_{1,1,n} = x_{1,1,n} - \gamma_n\big(x_{2,1,n} + v_{1,1,n}\big) \\ w_{1,1,n} = v_{1,1,n} - \gamma_n\big(v_{2,1,n} - x_{1,1,n} + x_{1,3,n}\big) \\ y_{2,1,n} = x_{2,1,n} + \gamma_n x_{1,1,n} \\ w_{2,1,n} = v_{2,1,n} + \gamma_n v_{1,1,n} \\ y_{1,2,n} = x_{1,2,n} - \gamma_n\big(A^* x_{2,2,n} + v_{1,2,n}\big) \\ w_{1,2,n} = v_{1,2,n} - \gamma_n\big(v_{2,2,n} - x_{1,2,n} + x_{1,3,n}\big) \\ y_{2,2,n} = x_{2,2,n} + \gamma_n A x_{1,2,n} \\ w_{2,2,n} = v_{2,2,n} + \gamma_n v_{1,2,n} \\ y_{1,3,n} = x_{1,3,n} + \gamma_n(v_{1,1,n} + v_{1,2,n}) \\ x_{1,1,n+1} = x_{1,1,n} - \gamma_n\big(prox_{\gamma_n f_1^*} y_{2,1,n} + w_{1,1,n}\big) \\ v_{1,1,n+1} = v_{1,1,n} + \gamma_n\big(y_{1,1,n} - y_{1,3,n}\big) \\ x_{2,1,n+1} = x_{2,1,n} - y_{2,1,n} + prox_{\gamma_n f_1^*} y_{2,1,n} + \gamma_n y_{1,1,n} \\ v_{2,1,n+1} = v_{2,1,n} - w_{2,1,n} + \gamma_n w_{1,1,n} \\ x_{1,2,n+1} = x_{1,2,n} - \gamma_n\big(A^* prox_{\gamma_n f_2^*} y_{2,2,n} + w_{1,2,n}\big) \\ v_{1,2,n+1} = v_{1,2,n} + \gamma_n\big(y_{1,2,n} - y_{1,3,n}\big) \\ x_{2,2,n+1} = x_{2,2,n} - y_{2,2,n} + prox_{\gamma_n f_2^*} y_{2,2,n} + \gamma_n A y_{1,2,n} \\ v_{2,2,n+1} = v_{2,2,n} - w_{2,2,n} + \gamma_n w_{1,2,n} \\ x_{1,3,n+1} = x_{1,3,n} + \gamma_n(w_{1,1,n} + w_{1,2,n}). \end{array} \right.$$

Then $(x_{1,1,n}, x_{1,2,n}, x_{1,3,n}) \to (\overline{x}, \overline{x}, \overline{x})$.

Figure 4.1 shows the reconstructed images with 500 and 1000 iterations.

Another test image which is very popular and is often used in image processing is the 256×256 *Lena test image*. The picture undergoes the same blur as described in the previous case. Figure 4.2 shows the reconstructed images with 500 and 1500 iterations.

A third example in image processing is the 256×256 *Barbara test image*. We ran our algorithm on this test image too in order to demonstrate the applicability of the method on different images. Figure 4.3 shows the reconstructed images in the case

(a) Blurred and noisy image (b) 500 iterations (c) 1000 iterations (d) Original image

Figure 4.1: Figure (a) is the obtained image after multiplying with a blur operator and adding white Gaussian noise, (b) shows the averaged sequence generated by Algorithm 4.1.2 after 500 iterations. Figure (c) is the reconstructed image after 1000 iterations, (d) shows the clean 256×256 Cameraman test image.

of 400 and 800 iterations.

A valuable tool for measuring the quality of these images is the so-called *improvement in signal-to-noise ratio (ISNR)*, which is defined as

$$\mathrm{ISNR}(k) = 10 \log_{10} \left(\frac{\|x - b\|^2}{\|x - x_k\|^2} \right)$$

Figure 4.2: Figure (a) is the obtained image after multiplying with a blur operator and adding white Gaussian noise, (b) shows the averaged sequence generated by Algorithm 4.1.2 after 500 iterations. Figure (c) is the reconstructed image after 1500 iterations, (d) shows the clean 256×256 Lena test image.

where x, b and x_k denote the original, observed and estimated image at iteration k, respectively. Figure 4.4 shows the evolution of the ISNR values for the three images presented.

Figure 4.3: Figure (a) is the obtained image after multiplying with a blur operator and adding white Gaussian noise, (b) shows the averaged sequence generated by Algorithm 4.1.2 after 400 iterations. Figure (c) is the reconstructed image after 800 iterations, (d) shows the clean 256×256 Barbara test image.

4.2 Multifacility location problems

The second numerical experiment solves the multifacility location problem. It was as early as the 17$^{\text{th}}$ century that mathematicians, notably Fermat, were concerned with what are known as single facility location problems. However, it was not until the 20$^{\text{th}}$ century that normative approaches to solving symbolic models of these and related problems were addressed in the literature. Each of these solution techniques

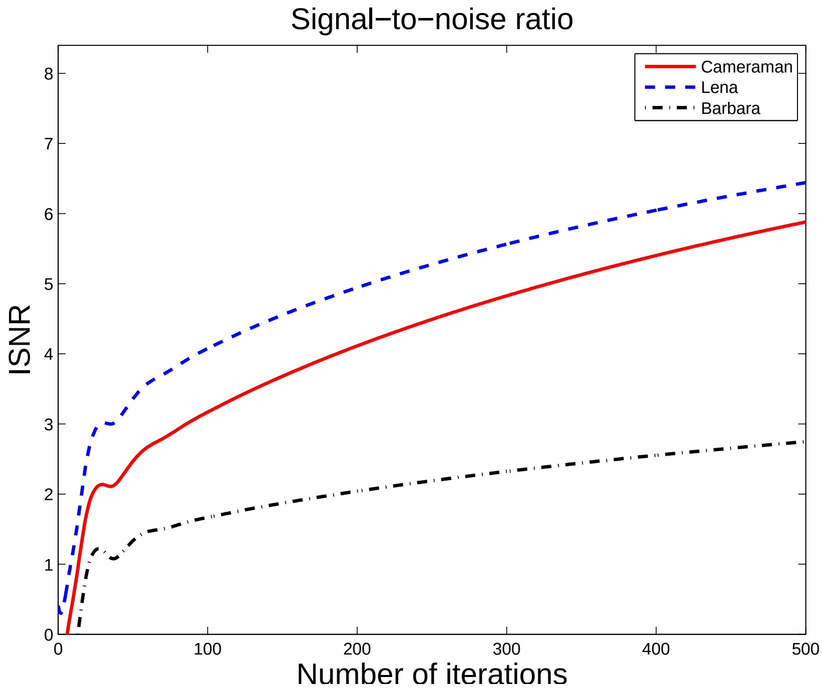

Figure 4.4: Evolution of the signal-to-noise ratio (ISNR) for the Cameraman, Lena and Barbara test images.

concerned themselves with determining the location of a new facility, or new facilities, with respect to the location of existing facilities so as to minimize a cost function based on a weighted interfacility distance measure [17, 18, 49, 51, 52, 64–66].

The Euclidean distance problem for the case of single new facilities was addressed by Weiszfeld [63], Miehle [45], Kuhn and Kuenne [42], and Cooper [24–26] to name a few. However, it was not until the work of Kuhn [41] that the problem was considered completely solved. A computational procedure for minimizing the Euclidean multifacility problem was presented by Vergin and Rogers [60] in 1967; however, their techniques sometimes give suboptimum solutions. Two years later, Love [44] gave a scheme for solving this problem which makes use of convex programming and penalty function techniques. One advantage to this approach is that it considers the existence of various types of spatial constraints. In 1973, Eyster, White and Wierwille [31] presented the hyperboloid approximation procedure (HAP) for both rectilinear and Euclidean distance measures which extended the technique employed in solving the single facility problem to the multifacility case [18].

If one studies a list of references of the work done in the past decade involving facility location problems, it becomes readily apparent that there exists a strong interdisciplinary interest in this area within the fields of applied mathematics, operation research, civil engineering, architecture, management science, systems engineering, logistics, economics, regional science, transportation systems, urban design and industrial engineering, among others. As a result, the term "facility" has taken on a very broad connotation in order to suit applications in each of these areas. For example, a facility can be a hospital, an ambulance, an airport, a police cruise, a manned space vehicle, a machine tool, a school, a student to be bused to a school in an urban environment, a remote computer terminal, a home appliance, a planned community, a warehouse, an office building, a pump in a pipeline, a sewage treatment plant, and so on [31, 33, 37]. Francis and Goldstein [34] provide a fairly recent bibliography of the facility location literature. One of the most complete classifications of these problems is provided in a book by Francis and White [35].

4.2.1 Problem manipulation

We proceed with the presentation of an efficient solution procedure for solving a multifacility location problem, which can be formulated mathematically as follows:

$$\inf_{x_1,x_2}\left\{\sum_{i=1}^{k}\lambda_i\left\|x_1-c_i\right\|_2+\sum_{i=1}^{k}\gamma_i\left\|x_2-c_i\right\|_2+\alpha\left\|x_1-x_2\right\|\right\}, \tag{4.2.1}$$

where we use the following notations

x_j =vector location of a new facility $j, 1 \leq j \leq 2$;

c_i =vector location of an existing facility $i, 1 \leq i \leq k$;

λ_i =non negative weight between the new facility x_1 and an existing facility i, $1 \leq i \leq k$;

γ_i =non negative weight between the new facility x_2 and an existing facility i, $1 \leq i \leq k$;

α =non negative weight between the two new facilities x_1 and x_2.

The discussed multifacility location problem 4.2.1 is in fact a special instance of Problem 3.3.1 with $(m,p) := (2, 2k+1)$, $f_i = 0$, $h_i = \delta_{\{0\}}$, $K_i = 0$, where $i = 1, 2$, and

$$g_j(x) = \begin{cases} \lambda_j \left\| x - c_j \right\|_2, & \text{if } j = 1, \dots, k; \\ \gamma_j \left\| x - c_j \right\|_2, & \text{if } j = k+1, \dots, 2k; \\ \alpha \left\| x \right\|, & \text{if } j = 2k+1, \end{cases}$$

$$L_{j1} = \begin{cases} \mathrm{Id}, & \text{if } j = 1, \dots, k; \\ 0, & \text{if } j = k+1, \dots, 2k; \\ \mathrm{Id}, & \text{if } j = 2k+1, \end{cases} \quad \text{and} \quad L_{j2} = \begin{cases} 0, & \text{if } j = 1, \dots, k; \\ \mathrm{Id}, & \text{if } j = k+1, \dots, 2k; \\ -\mathrm{Id}, & \text{if } j = 2k+1. \end{cases}$$

Algorithm 4.2.1.

Hence, the iterative scheme in Algorithm 3.3.1 reads

$$\forall n \geq 0 \quad \left\lfloor \begin{array}{l} \text{For } i = 1, 2 \\ \left\lfloor \begin{array}{l} y_{1,i,n} = x_{1,i,n} - \gamma_n \big(x_{2,i,n} + \sum_{j=1}^{2k+1} L_{ji}^* v_{1,j,n} \big) \\ y_{2,i,n} = x_{2,i,n} + \gamma_n x_{1,i,n} \end{array} \right. \\ \text{For } j = 1, \dots 2k+1 \\ \left\lfloor \begin{array}{l} w_{1,j,n} = v_{1,j,n} - \gamma_n \big(v_{2,j,n} - L_{j1} x_{1,1,n} - L_{j2} x_{1,2,n} \big) \\ w_{2,j,n} = v_{2,j,n} + \gamma_n v_{1,j,n} \\ r_{2,j,n} = prox_{\gamma_n g_j} w_{2,j,n} \end{array} \right. \\ \text{For } i = 1, 2 \\ \left\lfloor \begin{array}{l} x_{1,i,n+1} = x_{1,i,n} - \gamma_n \big(\sum_{j=1}^{2k+1} L_{ji}^* w_{1,j,n} \big) \\ x_{2,i,n+1} = x_{2,i,n} - y_{2,i,n} + \gamma_n y_{1,i,n} \end{array} \right. \\ \text{For } j = 1, \dots, 2k+1 \\ \left\lfloor \begin{array}{l} v_{1,j,n+1} = v_{1,i,n} - \gamma_n \big(r_{2,j,n} - L_{j1} y_{1,1,n} - L_{j2} y_{1,2,n} \big) \\ v_{2,j,n+1} = \gamma_n \big(w_{1,j,n} - v_{1,j,n} \big) + r_{2,j,n}. \end{array} \right. \end{array} \right.$$

The algorithm implemented in the MatLab programming language works with the parameter $\beta = 2 * \sqrt{k+1} + 1$. We choose ε to be as small as possible from the interval $]0, 1/(\beta+1)[$ and the value of the parameter γ equals $(1-\varepsilon)/\beta$, for an optimal solution.

4.2.2 Computational experience

As an illustration of Algorithm 4.2.1, consider a location problem involving two new facilities and five existing facilities. This example is also used for illustrating the use of the hyperboloid approximation procedure presented in [31]. Since distances between facilities are Euclidean, the facilities might be machines connected by straight-line conveyors, military base connected by air travel, or industrial plants and warehouses where strait line travel can be reasonably approximated. For the example, assume the existing facilities are located at the coordinate points:

$$P_1 = (0,0); P_2 = (2,4); P_3 = (6,2); P_4 = (6,10) \text{ and } P_5 = (8,8).$$

Also, let $\lambda_i = (4,2,3,0,0)$, $\gamma_i = (0,2,1,3,2)$, $i = 1,\ldots,5$ and $\alpha = 2$. With a starting location of $x_1^0 = (0,0)$ and $x_2^0 = (0,0)$, the result of the algorithm is illustrated in Figure 4.5.

Most current techniques used for solving location problems, including Algorithm 4.2.1, define the stopping criterion based on successive changes in the value of the objective function, which is more efficient than the one based on successive changes in the location of the new facilities (convergent solution). The projection technique used in the proposed method takes full advantage of the structure of the location problem and can easily be extended to accommodate problems involving other item movements as well.

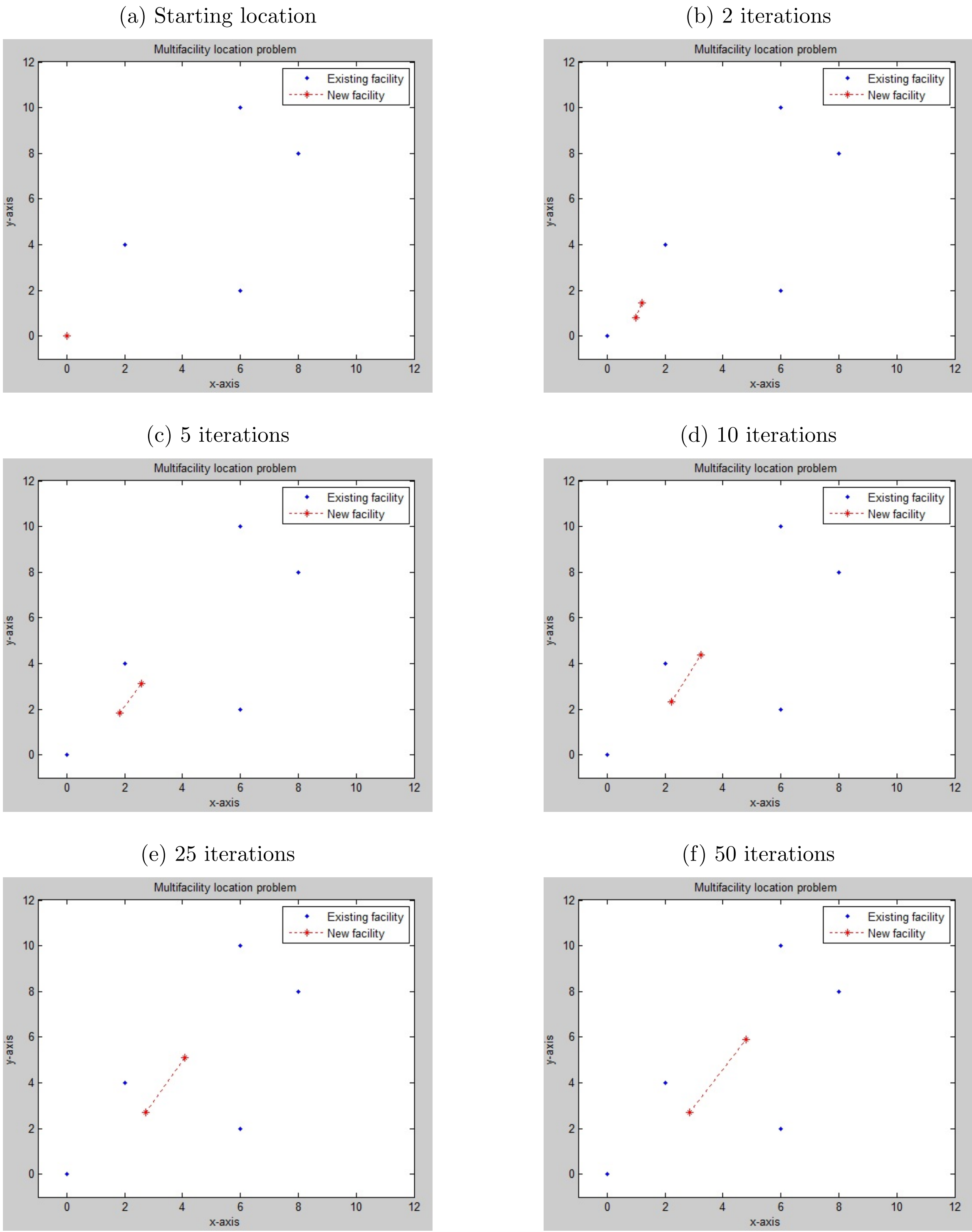

Figure 4.5: A multifacility location problem involving two new facilities and five existing facilities. The existing facilities are located at the coordinate points $P_1 = (0, 0)$, $P_2 = (2, 4)$, $P_3 = (6, 2)$, $P_4 = (6, 10)$ and $P_5 = (8, 8)$. The non negative weight α between the two new facilities equals 2 and the weights between a new facility and an existing facility equals $\lambda_i = (4, 2, 3, 0, 0)$ and $\gamma_i = (0, 2, 1, 3, 2)$, $i = 1, \ldots, 5$.

4.3 Average consensus for colored networks

The third numerical experiment that we consider concerns the problem of average consensus on colored networks.

Given a network, where each node possesses a measurement in form of a real number, the average consensus problem consists in calculating the average of these measurements in a recursive and distributed way, allowing the nodes to communicate information along only the available edges in the network.

4.3.1 Proposed algorithm

Consider a connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of *nodes* and $\mathcal{E}$ represents the set of *edges*. Each edge is uniquely represented as a pair of nodes (i, j), where $i < j$. The nodes i and j can exchange their values if they can communicate directly, in other words, if $(i, j) \in \mathcal{E}$. The set of neighbors of node k is represented with $\mathcal{N}_k$ and its degree with $D_k = |\mathcal{N}_k|$. We assume that each node possesses a measurement in form of a real number, also called *color*, and that no neighboring nodes have the same color. Let C denote the number of colors the network is colored with and $\mathcal{C}_i$ the set of the nodes that have the color i, $i = 1, \dots, C$. Without affecting the generality of the problem we also assume that the first C_1 nodes are in the set $\mathcal{C}_1$, the next C_2 nodes are in the set $\mathcal{C}_2$, etc. Furthermore, we assume that a node coloring scheme is available. For more details concerning the mathematical modeling of the average consensus problem on colored networks we refer the reader to [46, 47].

Let P and E denote the number of nodes and edges in the network, respectively, hence, $\sum_{i=1}^{C} C_i = P$. Denoting by θ_k the measurement assigned to node k, $k = 1, \dots, P$, the problem we want to solve is

$$\min_{x \in \mathbb{R}} \left\{ \sum_{k=1}^{P} \frac{1}{2}(x - \theta_k)^2 \right\}. \tag{4.3.1}$$

The unique optimal solution to the problem (4.3.1) is $\theta^* = \frac{1}{P}\sum_{k=1}^{P} \theta_k$, namely the average of the measurements over the whole set of nodes in the network. The goal is to make this value available in each node in a distributed and recursive way. To this end, we replicate copies of x throughout the entire network, more precisely, for $k = 1, ..., P$, node k will hold the k-th copy, denoted by x_k, which will be updated iteratively during the algorithm. At the end we have to guarantee that all the copies are equal and we express this constraint by requiring that $x_i = x_j$ for each $(i, j) \in \mathcal{E}$. This gives rise to the following optimization problem

$$\min_{\substack{\overline{x}=(x_1,...,x_P)\in\mathbb{R}^P \\ x_i=x_j,\ \forall\{i,j\}\in\mathcal{E}}} \Big\{ \sum_{k=1}^{P} \frac{1}{2}(x_k - \theta_k)^2 \Big\}. \tag{4.3.2}$$

Let $A \in \mathbb{R}^{P\times E}$ be the *node-arc incidence matrix* of the network, which is the matrix having each column associated to an edge in the following manner: the column associated to the edge $(i, j) \in \mathcal{E}$ has 1 at the i-th entry and -1 at the j-th entry, the remaining entries being equal to zero. Consequently, constraints in (4.3.2) can be written with the help of the *transpose* of the node-arc incidence matrix as $A^T\overline{x} = 0$. Taking into consideration the ordering of the nodes and the coloring scheme, we can write $A^T\overline{x} = A_1^T\overline{x}_1 + \ldots + A_C^T\overline{x}_C$, where $\overline{x}_i \in \mathbb{R}^{C_i}$, $i = 1, ..., C$, collects the copies of the nodes in $\mathcal{C}_i$, i.e.

$$\overline{x} = (\underbrace{x_1, ..., x_{C_1}}_{\overline{x}_1}, ..., \underbrace{x_{P-C_C+1}, ..., x_P}_{\overline{x}_C}).$$

Hence, the optimization problem (4.3.2) becomes

$$\min_{\substack{\overline{x}=(\overline{x}_1,...,\overline{x}_C) \\ A_1^T\overline{x}_1+...+A_C^T\overline{x}_C=0}} \Big\{ \sum_{i=1}^{C} f_i(\overline{x}_i) \Big\}, \tag{4.3.3}$$

where for $i = 1, ..., C$, the function $f_i : \mathbb{R}^{C_i} \to \mathbb{R}$ is defined as $f_i(\overline{x}_i) = \sum_{l\in\mathcal{C}_i} \frac{1}{2}(x_l - \theta_l)^2$.

One can easily observe that problem (4.3.3) is a particular instance of the optimization problem (3.3.1), when taking

$$m = C, p = 1, h_i = \delta_{\{0\}}, K_i = \mathrm{Id}, L_{1i} = A_i^T \in \mathbb{R}^{E\times C_i}, i = 1, ..., C, \text{ and } g_1 = \delta_{\{0\}}.$$

Algorithm 4.3.1.

Using that $h_i^* = 0, i = 1, ..., C$, *and* $\mathrm{Prox}_{\gamma g}(x) = 0$ *for all* $\gamma > 0$ *and* $x \in \mathbb{R}^E$, *the iterative scheme in Algorithm 3.3.1 reads, after some algebraic manipulations, in the* *error-free case:*

$$
\forall n \geq 0 \quad \left\lfloor \begin{array}{l}
\textit{For } i = 1, \dots, C \\
\left\lfloor \begin{array}{l}
y_{1,i,n} = x_{1,i,n} - \gamma_n \big(x_{2,i,n} + A_i v_{1,1,n} \big) \\
y_{2,i,n} = x_{2,i,n} + \gamma_n x_{1,i,n} \\
p_{2,i,n} = \mathrm{Prox}_{\gamma_n f_i^*} \, y_{2,i,n}
\end{array} \right. \\
w_{1,1,n} = v_{1,1,n} - \gamma_n \big(v_{2,1,n} - \sum_{i=1}^{C} A_i^T x_{1,i,n} \big) \\
\textit{For } i = 1, \dots, C \\
\left\lfloor \begin{array}{l}
q_{1,i,n} = y_{1,i,n} - \gamma_n \big(p_{2,i,n} + A_i w_{1,1,n} \big) \\
q_{2,i,n} = p_{2,i,n} + \gamma_n y_{1,i,n} \\
x_{1,i,n+1} = x_{1,i,n} - y_{1,i,n} + q_{1,i,n} \\
x_{2,i,n+1} = x_{2,i,n} - y_{2,i,n} + q_{2,i,n}
\end{array} \right. \\
v_{1,1,n+1} = v_{1,1,n} + \gamma_n \sum_{i=1}^{C} A_i^T y_{1,i,n} \\
v_{2,1,n+1} = \gamma_n^2 \big(\sum_{i=1}^{C} A_i^T x_{1,i,n} - v_{2,1,n} \big).
\end{array} \right.
$$

Let us notice that for $i = 1, ..., C$ and $\gamma > 0$ it holds $\mathrm{Prox}_{\gamma f_i^*}(\overline{x}_i) = (1+\gamma)^{-1}(\overline{x}_i - \gamma\overline{\theta}_i)$, where $\overline{\theta}_i$ is the vector in $\mathbb{R}^{C_i}$ whose components are θ_l with $l \in \mathcal{C}_i$.

4.3.2 Performance evaluation

In order to compare the performances of our method with other existing algorithms in literature, we used the networks generated in [47] with the number of nodes ranging between 10 and 1000. The measurement θ_k associated to each node was generated randomly and independently from a normal distribution with mean 10 and standard deviation 100. We worked with networks with 10, 50, 100, 200, 500, 700 and 1000 nodes and measured the performance of our algorithm from the point of view of the number of communication steps, which actually coincides with the number of iterations. As stopping criterion we considered

$$\frac{\|x_n - \mathbb{1}_P \theta^*\|}{\left|\sqrt{P}\theta^*\right|} \leq 10^{-4},$$

where $\mathbb{1}_P$ denotes the vector in $\mathbb{R}^P$ having all entries equal to 1.

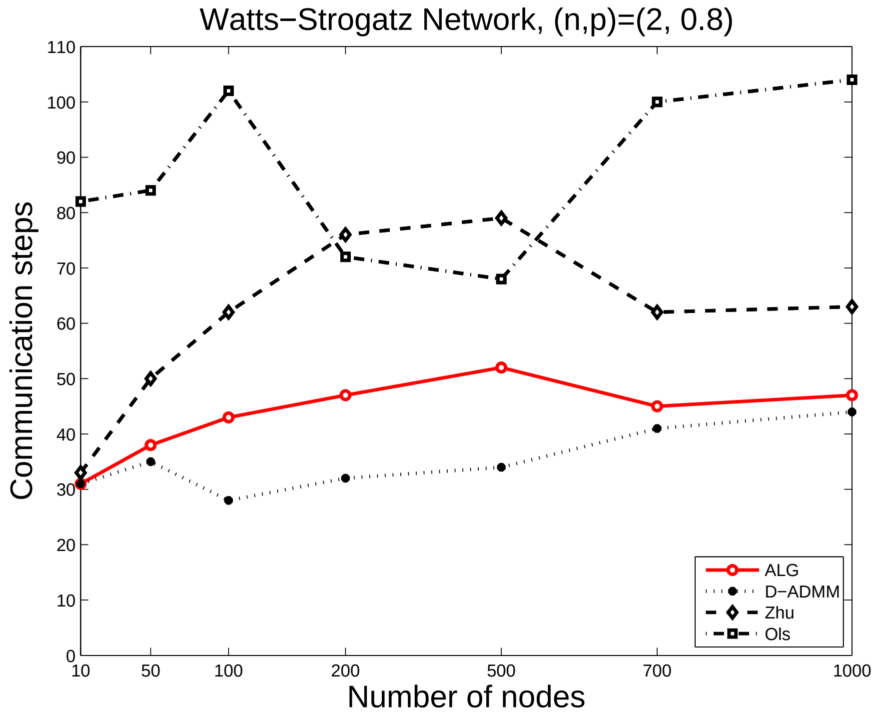

Figure 4.6: The chart shows the communication steps needed by the four algorithms for a Watts-Strogatz network with different number of nodes. ALG stands for the primal-dual algorithm proposed in this work.

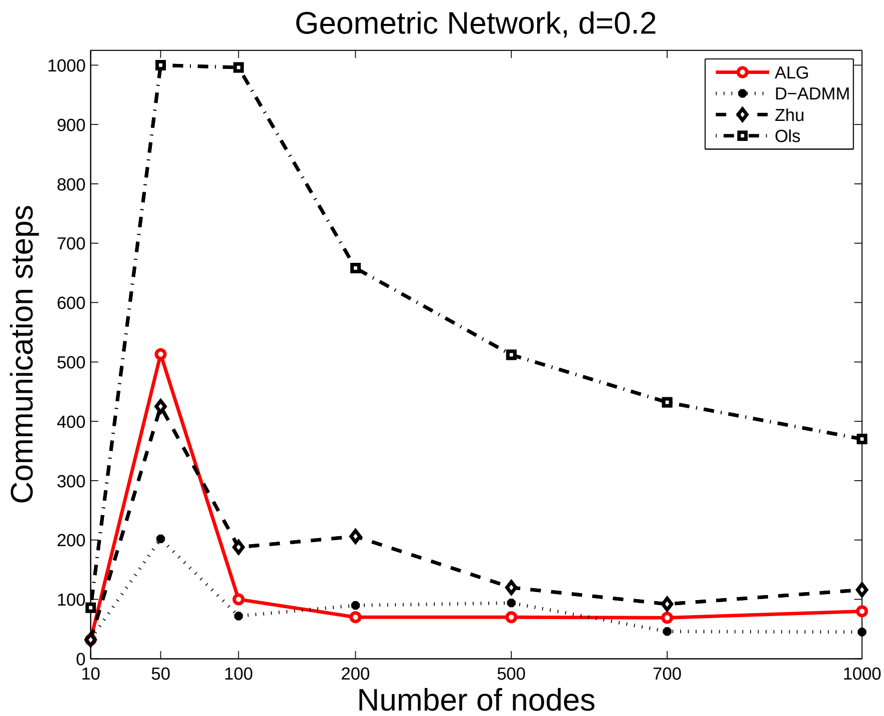

Figure 4.7: The chart shows the communication steps needed by the four algorithms for a Geometric network with different number of nodes. ALG stands for the primal-dual algorithm proposed in the thesis.

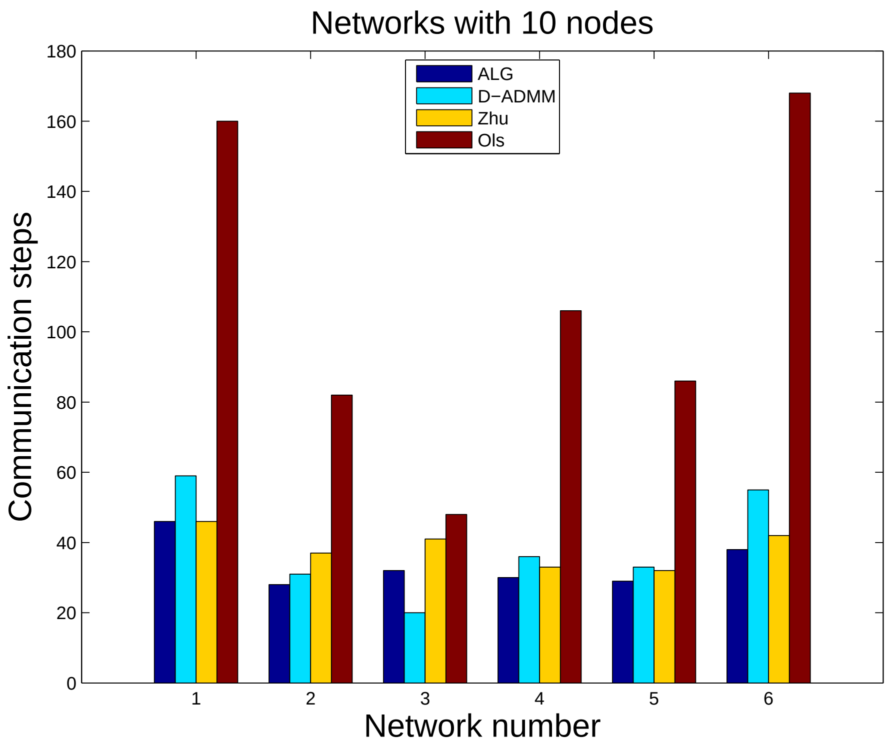

Figure 4.8: Comparison of the four algorithms over six networks with 10 nodes. Here, ALG stands for the proposed primal-dual algorithm.

We present in Figure 4.6 and Figure 4.7 the communication steps needed when dealing with the *Watts-Strogatz network* with parameters (2,0.8) and with the *Geometric network* with a distance parameter 0.2. The Watts-Strogatz network is created from a lattice where every node is connected to 2 nodes, then the links are rewired with a probability of 0.8, while the Geometric network works with nodes in a $[0, 1]^2$ square and connects the nodes whose Euclidean distance is less than the given parameter 0.2. As shown in Figure 4.6 and in Figure 4.7, our algorithm performed comparable to D-AMM, presented in [47], but it performed better then the algorithms presented in [48] and [70].

In order to observe the behavior of our algorithm on different networks, we tested it on 6 models, shown in Table 4.1, with a different number of nodes. The used models and the role of its parameters are explained in Table 4.2. Observing the needed communication steps, we can conclude that our algorithm is communication-efficient and it performs better than or similarly to the algorithms in [47], [48] and [70] (as exemplified in Figure 4.8).

Number	Model	Parameters
1	Erdős-Rényi	0.25
2	Watts-Strogatz	$(2, 0.8)$
3	Watts-Strogatz	$(4, 0.6)$
4	Barabasi-Albert	—
5	Geometric	0.2
6	Lattice	—

Table 4.1: Network parameters.

Name	Parameters	Description
Erdős-Rényi [30]	p	Every pair of nodes {i,j} is connected or not with probability p.
Watts-Strogatz [62]	(n, p)	First, it creates a lattice where every node is connected to n nodes; then, it rewires every link with probability p. If link {i,j} is to be rewired, it removes the link and connects node i or node j (chosen with equal probability) to another node in the network, chosen uniformly.
Barabasi-Albert [3]	—	It starts with one node. At each step, one node is added to the network by connecting it to two existing nodes: the probability to connect it to node k is proportional to D_k.
Geometric [50]	d	It drops P points, corresponding to the nodes of the network, randomly in a $[0, 1]^2$ square; then, it connects node whose Euclidean distance is less than d.
Lattice	—	Creates a lattice of dimensions $m \times n$; m and n are chosen to make the lattice as square as possible.

Table 4.2: Network models.

4.4 Support-vector machines classification

The fourth numerical experiment we present in this section addresses the problem of classifying images via support vector machines.

Having a set training data $a_i \in \mathbb{R}^n$, $i = 1, \ldots, k$, belonging to one of two given classes, denoted by "-1" and "+1", the aim is to construct by it a decision function given in the form of a separating hyperplane which should assign every new data to one of the two classes with a low misclassification rate.

4.4.1 The soft-margin hyperplane

We construct the matrix $A \in \mathbb{R}^{k\times n}$ such that each row corresponds to a data point $a_i, i = 1, ..., k$ and a vector $d \in \mathbb{R}^k$ such that for $i = 1, ..., k$ its i-th entry is equal to -1, if a_i belongs to the class "-1" and it is equal to $+1$, otherwise. Consider the case where the training data cannot be separated without error. In this case one may want to separate the training set with a minimal number of errors. In order to cover the situation when the separation cannot be done exactly, we consider non-negative slack variables $\xi_i \geq 0,\ i = 1, \ldots, k$, thus the goal will be to find $(s, r, \xi) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^k_+$ as optimal solution of the following optimization problem (also called *soft-margin support vector machines problem*)

$$\min_{\substack{(s,r,\xi)\in\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}^k_+ \\ D(As+\mathbb{1}_k r)+\xi\geqq\mathbb{1}_k}} \left\{\|s\|^2 + C\|\xi\|^2\right\}, \tag{4.4.1}$$

where $\mathbb{1}_k$ denotes the vector in $\mathbb{R}^k$ having all entries equal to 1, the inequality $z \geqq \mathbb{1}_k$ for $z \in \mathbb{R}^k$ means $z_i \geq 1, i = 1, ...k$, $D = \text{Diag}(d)$ is the diagonal matrix having the vector d as main diagonal and C is a trade-off parameter. Each new data $a \in \mathbb{R}^n$ will by assigned to one of the two classes by means of the resulting decision function $z(a) = s^T a + r$, namely, a will be assigned to the class "-1", if $z(a) < 0$, and to the class "+1", otherwise. For more theoretical insights in support vector machines we refer the reader to [28, 32].

The soft-margin support vector machines problem (4.4.1) can be written as a special instance of the optimization problem (3.3.1), by taking

$$m = 3, p = 1, f_1(\cdot) = \|\cdot\|^2, f_2 = 0, f_3(\cdot) = C\|\cdot\|^2 + \delta_{\mathbb{R}^k_+}(\cdot), h_i = \delta_{\{0\}}, K_i = \text{Id}, i = 1, 2, 3,$$

$$g_1 = \delta_{\{z\in\mathbb{R}^k : z\geqq\mathbb{1}_k\}}, L_{11} = DA, L_{12} = D\mathbb{1}_k \text{ and } L_{13} = \text{Id}\,.$$

Algorithm 4.4.1.

Thus, Algorithm 3.3.1 gives rise in the error-free case to the following iterative scheme:

$$\forall n \geq 0 \quad \left\lfloor \begin{array}{l} \textit{For } i = 1,2,3 \\ \quad \left\lfloor \begin{array}{l} y_{1,i,n} = x_{1,i,n} - \gamma_n\big(x_{2,i,n} + L_{1i}^T v_{1,1,n}\big) \\ y_{2,i,n} = x_{2,i,n} + \gamma_n x_{1,i,n} \\ p_{2,i,n} = \operatorname{Prox}_{\gamma_n f_i^*} y_{2,i,n} \end{array} \right. \\ w_{1,1,n} = v_{1,1,n} - \gamma_n\big(v_{2,1,n} - \sum_{i=1}^3 L_{1i} x_{1,i,n}\big) \\ w_{2,1,n} = v_{2,1,n} + \gamma_n v_{1,1,n} \\ r_{2,1,n} = \operatorname{Prox}_{\gamma_n g_1} w_{2,1,n} \\ \textit{For } i = 1,2,3 \\ \quad \left\lfloor \begin{array}{l} q_{1,i,n} = y_{1,i,n} - \gamma_n\big(p_{2,i,n} + L_{1i}^T w_{1,1,n}\big) \\ q_{2,i,n} = p_{2,i,n} + \gamma_n y_{1,i,n} \\ x_{1,i,n+1} = x_{1,i,n} - y_{1,i,n} + q_{1,i,n} \\ x_{2,i,n+1} = x_{2,i,n} - y_{2,i,n} + q_{2,i,n} \end{array} \right. \\ s_{1,1,n} = w_{1,1,n} - \gamma_n(r_{2,1,n} - \sum_{i=1}^3 L_{1i} p_{1,i,n}) \\ s_{2,1,n} = r_{2,1,n} + \gamma_n w_{1,1,n} \\ v_{1,1,n+1} = v_{1,1,n} - w_{1,1,n} + s_{1,1,n} \\ v_{2,1,n+1} = v_{2,1,n} - w_{2,1,n} + s_{2,1,n} \end{array} \right.$$

We would also like to notice that for the proximal points needed in the algorithm one has for $\gamma > 0$ and $(s, r, \xi, z) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^k \times \mathbb{R}^k$ the following exact formulae:

$$\operatorname{Prox}_{\gamma f_1^*}(s) = (2+\gamma)^{-1}2s, \operatorname{Prox}_{\gamma f_2^*}(r) = 0, \operatorname{Prox}_{\gamma f_3^*}(\xi) = \xi - \gamma P_{\mathbb{R}_+^k}\big((2C+\gamma)^{-1}\xi\big)$$

and

$$\operatorname{Prox}_{\gamma g_1}(z) = P_{\{x \in \mathbb{R}^k : x \geqq \mathbb{1}_k\}}(z).$$

4.4.2 Experiments with digit recognition

We made use of a data set of 11907 training images and 2041 test images of size 28×28 from the website http://www.cs.nyu.edu/~roweis/data.html. The problem consisted in determining a decision function based on a pool of handwritten digits showing either the number two or the number nine, labeled by -1 and $+1$, respectively (see Figure 4.9). We evaluated the quality of the decision function on a test data set by computing the percentage of misclassified images. Notice that we use only a half of the available images from the training data set, in order to reduce the computational effort.

(a) A sample of data for number 2 (b) A sample of data for number 9

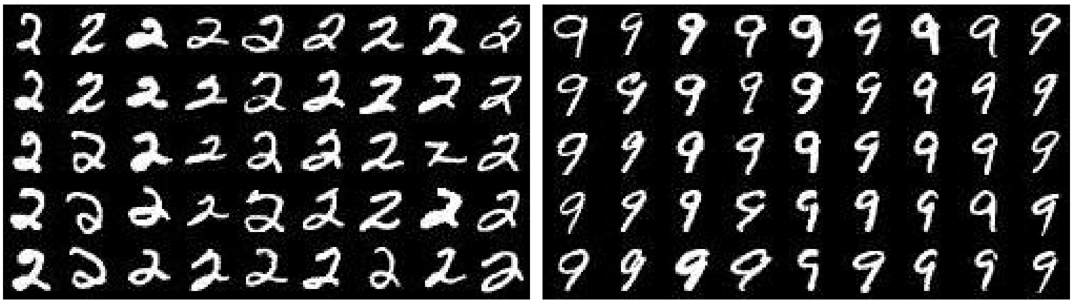

Figure 4.9: A sample of images belonging to the classes -1 and +1, respectively.

With respect to the considered data set, we denote by $\mathcal{D}= \{(X_i, Y_i), i = 1, ..., 6000\} \subseteq \mathbb{R}^{784} \times \{+1, -1\}$ the set of available training data consisting of 3000 images in the class -1 and 3000 images in the class $+1$. A sample from each class of images is shown in Figure 4.9. The images have been vectorized and normalized by dividing each of them by the quantity $\left(\frac{1}{6000}\sum_{i=1}^{6000}\|X_i\|^2\right)^{\frac{1}{2}}$.

Number of iterations	100	1000	2000	3000	5000
Training error	2.95	2.6	2.3	1.95	1.55
Test error	2.95	2.55	2.45	2.15	2

Table 4.3: Misclassification rate in percentage for different numbers of iterations for both the training data and the test data.

We stopped the primal-dual algorithm after different numbers of iterations and evaluated the performances of the resulting decision functions. In Table 4.3we present the misclassification rate in percentage for the training and for the test data (the error for the training data is less than the one for the test data) and observe that the quality of the classification increases with the number of iterations. However, even for a low number of iterations the misclassification rate outperforms the ones reported in the literature dealing with numerical methods for support vector classification. Let us also mention that the numerical results are given for the case $C = 1$. We tested also other choices for C, however we did not observe great impact on the results.

Chapter 5

Interdisciplinary application of our algorithm

During my studies, I collaborated with a team of researchers in biology, studying the biodiversity of picoalgae in Romanian salt lakes. We investigated the phytoplankton communities in several hypersaline lakes in the Transylvanian basin, with the latest microscopy and molecular biological techniques [38–40]. The aim was to adapt our algorithm for image processing and pattern recognition for microscopic images and electrophoretograms. The developing process included the work with epifluorescence microscopic images of picoalgae in order to count and to distinguish them from the background noise.

5.1 Optimization in biology

Most of the molecular techniques lead to the method of electrophoresis in which DNA or proteins can be separated in a gel using electricity. In our case DNA fragments from different organisms collected from the saline lakes were migrated, using agarose gel electrophoresis, in order to separate the DNA fragments. Agarose is a polysaccharide polymer material, generally extracted from seaweed and is frequently used in molecular biology, biochemistry and clinical chemistry for the separation of large molecules, especially a mixed population of DNA, in a matrix of agarose.

After the patterns of the migrated DNA fragments are obtained (see Figure

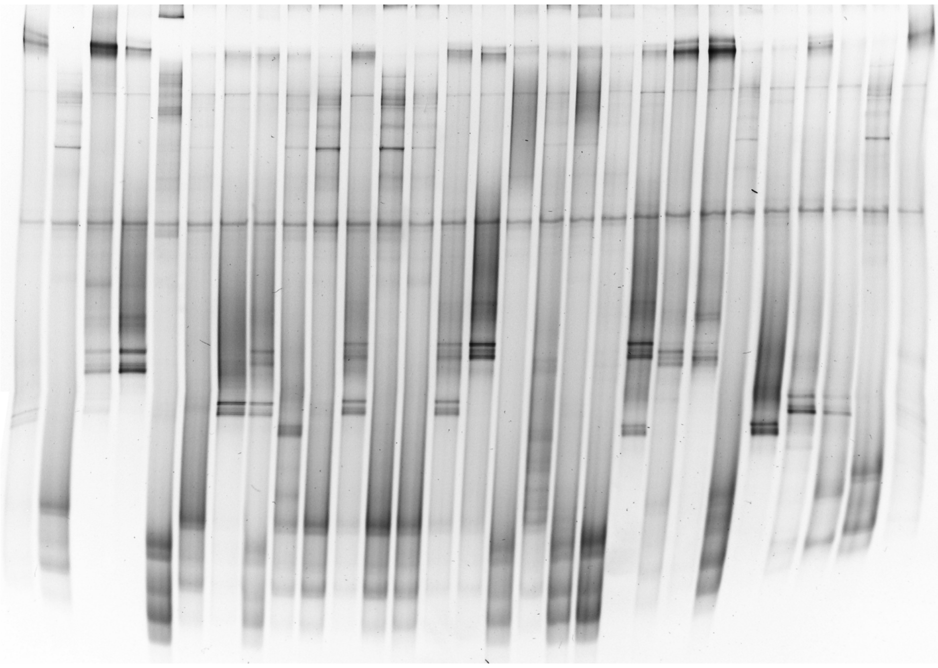

Figure 5.1: Denaturated gradient gel electrophoresis pattern of DNA samples from the salt lakes of the Transylvanian-Basin. Columns represent different water samples and the horizontal bands within the columns represent different species.

5.1), they are analyzed and the different samples are compared to establish differences between the salt lakes. Also, the patterns can be used to establish differences within a lake between different water depths [38]. Molecular processing such as cloning and sequencing can be an expensive and time consuming method. Our aim is to optimize our algorithm to find specific patterns, in order to make the process cheaper and faster. Patterns can be distinguished with a method called pattern recognition. This is actually a classification, which attempts to assign each input value to one of a given set of classes. The procedure is a non-probabilistic binary linear classifier, very similar to the classification via support vector machines (see 4.4). In our case, epifluorescence microscopic images were taken in order to count the cyanobacterial cells from the water samples, as shown in Figure 5.2.

5.2 Pattern recognition techniques

Enumerating the algae content of a water sample typically involves manual counts through a microscope. Recent advanced pattern recognition software enable these counts to be automated. The new techniques have the potential to advance the sensitivity of monitoring systems by providing much more information on the state of the water supply in near-real time, with more statistical significance than conventional microscopy. The availability of this information can yield significant cost savings by allowing system operators to proactively treat water supplies.

In general, different species of picoalgae can seem quite clearly different to the human eye, but they are not as easily distinguished mathematically. Automated pattern recognition in images has been a complex area of research within computer science for many years, because many distinctions we make regularly through our eyes and brain remain very difficult to accomplish computationally.

Our work in this area has a great progress with good results. In the future, we propose to develop our optimization technique for pattern recognition, in order to eliminate the unclassified particles and to obtain better classification results of microscopic images.

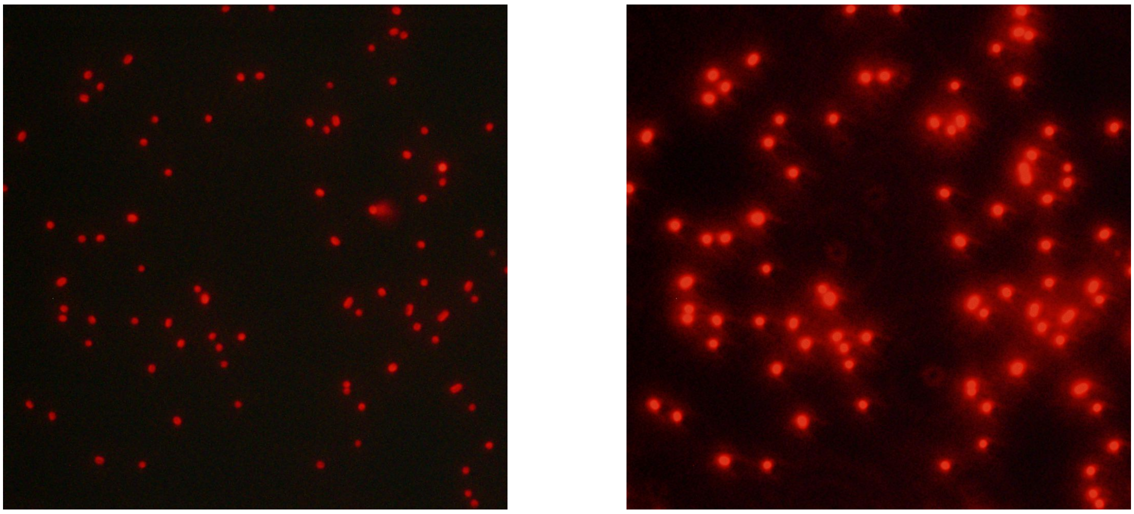

Figure 5.2: Epifluorescence microscopic images (magnification = ×1000) of pikocyanobacteria. The first image shows the pikocyanobacteria with blue-violet excitation and in the second image they are shown with green excitation.

References

[1] H. Attouch, L.M. Briceño-Arias and P.L. Combettes. A parallel splitting method for coupled monotone inclusions. *SIAM Journal on Control and Optimization*, 48(5):3246–3270, 2010.

[2] H. Attouch and M. Théra. A general duality principle for the sum of two operators. *Journal of Convex Analysis*, 3:1–24, 1996.

[3] A. Barabasi and R. Albert. Emergence of scalling in random networks. *Science*, 286:509–512, 1999.

[4] H.H. Bauschke and P.L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics, Springer-Verlag New York, 2011.

[5] H.H. Bauschke, P.L. Combettes and S. Reich. The asymptotic behavior of the composition of two resolvents. *Nonlinear Analysis*, 60(2):283–301, 2005.

[6] A. Beck and M. Teboulle. Smoothing and first order methods: a unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

[7] R.I. Boţ. *Conjugate Duality in Convex Optimization*. Lecture Notes in Economics and Mathematical Systems, Vol. 637, Springer-Verlag Berlin Heidelberg, 2010.

[8] R.I. Boţ and E.R. Csetnek. Regularity conditions via generalized interiority notions in convex optimization: new achievements and their relation to some classical statements. *Optimization*, 61(1):35–65, 2012.

[9] R.I. Boţ, E.R. Csetnek and A. Heinrich. A primal-dual splitting algorithm for finding zeros of sums of maximally monotone operators. *To appear in Siam Journal on Optimization*, 2012.

[10] R.I. Boţ, E.R. Csetnek and **E. Nagy**. Solving systems of monotone inclusions via primal-dual splitting techniques. *To appear in Taiwanese Journal of Mathematics*, 2013.

[11] R.I. Boţ and C. Hendrich. A double smoothing technique for solving unconstrained nondifferentiable convex optimization problems. *Computational Optimization and Applications*, 54(2):239–262, 2013.

[12] R.I. Boţ and C. Hendrich. A Douglas-Rachford type primal-dual method for solving inclusions with mixtures of composite and parallel-sum type monotone operators. 2012. arXiv:1212.0326v1[math.OC]

[13] R.I. Boţ and C. Hendrich. A variable smoothing algorithm for solving convex optimization problems. *Optimization*, 2012. arXiv:1207.3254[math.OC]

[14] R.I. Boţ and C. Hendrich. Convergence analysis for a primal-dual monotone + skew splitting algorithm with applications to total variation minimization. 2012. arXiv:1211.1706v1[math.OC]

[15] R.I. Boţ and C. Hendrich. On the acceleration of the double smoothing technique for unconstrained convex optimization problems. *Optimization*, 2012. http://dx.doi.org/10.1080/02331934.2012.745530

[16] L.M. Briceño-Arias and P.L. Combettes. A monotone + skew splitting model for composite monotone inclusions in duality. *SIAM Journal on Optimization*, 21(4):1230–1250, 2011.

[17] A. V. Cabot, R.L. Francis and M.A. Stary. A Network Flow Solution to a Rectilinear Distance Facility Location Problem. *American Institute of Industrial Engineers Transactions*, 2:132–141, 1970.

[18] P. Calamai and C. Charalambous. Solving multifacility location problems involving Euclidean distances. *Naval Research Logistics Quarterly*, 27(4):609–620, 1980.

[19] 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, 2011.

[20] P.L. Combettes. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 53(5–6):475–504, 2004.

[21] P.L. Combettes. Iterative construction of the resolvent of a sum of maximal monotone operators. *Journal of Convex Analysis*, 16(3–4):727–748, 2009.

[22] P.L. Combettes. Systems of Structured Monotone Inclusions: Duality, Algorithms, and Applications. 2012. arXiv:1212.6631v1[math.OC]

[23] P.L. Combettes and J.-C. Pesquet. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators. *Set-Valued and Variational Analysis*, 20(2):307–330, 2012.

[24] L. Cooper. Location-Allocation Problems. *Operations Research*, 11:331–344, 1963.

[25] L. Cooper. Heuristic Methods for Location-Allocation Problems. *SIAM Review*, 6:37–52, 1964.

[26] L. Cooper. Solutions of Generalized Locational Equilibrium Problems. *Journal of Regional Science*, 7:1–18, 1967.

[27] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[28] N. Cristianini and J.S. Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, 2000.

[29] I. Ekeland and R. Temam. *Convex Analysis and Variational Problems*. North-Holland Publishing Company, Amsterdam, 1976.

[30] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[31] J. W. Eyster, J.A.White and W.W. Wierwille. On Solving Multifacility Location Problems using a Hyperboloid Approximation Procedure. *American Institute of Industrial Engineers Transactions*, 5(1):01–06, 1973.

[32] P.A. Forero, A. Cano and G.B. Giannakis. Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, 11:1663–1707, 2010.

[33] R.L. Francis and A.V. Cabot. Properties of a Multifacility Location Problem Involving Euclidean distances. *Naval Research Logistics Quarterly*, 19:335–353, 1972.

[34] R.L. Francis and J.M. Goldstein. Location Theory: A Selective Bibliography. *Operations Research*, 22:400–410, 1974.

[35] R.L. Francis and J. A. White. *Facility Layout and Location: An Analytic Approach.* Prentice-Hall, Englewood Cliffs, New Jersey 1974.

[36] S.L.S. Jacoby, J.S. Kowalik and J.T. Pizzo. *Iterative Methods for Nonlinear Optimization Problems.* Prentice-Hall, Inc., 1972.

[37] H. Juel and R.F. Love. An Efficient Computational Procedure for Solving the Multifacility Rectilinear Facilities Location Problem. *Operational Research Quarterly*, 27:697–703, 1976.

[38] Zs.Gy. Keresztes, T. Felföldi, B. Somogyi, Gy. Székely, N. Dragoş, K. Márialigeti, Cs. Bartha and L. Vörös. First record of picophytoplankton diversity in Central European hypersaline lakes. *Extremophiles*, 16:759–769, 2012.

[39] Zs.Gy. Keresztes, B. Somogyi, E. Boros, Gy. Székely, Cs. Bartha, **E. Nagy**, N. Dragoş and L. Vörös. Picoplankton in soda lakes of the Carpathian Basin. *Contrib. Bot.*, XLV:41–46, 2010.

[40] Zs.Gy. Keresztes, **E. Nagy**, B. Somogyi, B. Németh, Cs. Bartha, Gy. Székely, N. Dragoş and L. Vörös. Trophic conditions of saline lakes in the Transylvanian-Basin. *Hung. Hydrobiol. Soc.*, 91(6):46–48, 2011.

[41] H.W. Kuhn. A Note on Fermat's Problem. *Mathematical Programming*, 4:98–107, 1973.

[42] H.W. Kuhn and R.E. Kuenne. An Efficient Algorithm for the Numerical Solution of the Generalized Weber Problem in Spatial Economics. *Journal of Regional Science*, 4:21–33, 1962.

[43] P.L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[44] R.F. Love. Locating Facilities in Three-Dimensional Space by Convex Programming. *Naval Research Logistics Quarterly*, 16:503–516, 1969.

[45] W. Miehle. Link-Length Minimization in Networks. *Operations Research*, 6:232–243, 1958.

[46] J.F.C. Mota, J.M.F. Xavier, P.M.Q. Aguiar and M. Püschel. ADMM for consensus on colored networks. *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, 5116-5121, 2012.

[47] J.F.C. Mota, J.M.F. Xavier, P.M.Q. Aguiar and M. Püschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. 2012. arXiv:1202.2805[math.OC]

[48] A. Olshevsky and J. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1):33–55, 2009.

[49] L.M. Ostresh. The Multifacility Location Problem: Applications and Descent Theorems. *Journal of Regional Science*, 17:409–419, 1977.

[50] M. Penrose. *Random Geometric Graphs.* Oxford University Press, 2004.

[51] A.A.B. Pritsker and P.M. Ghare. Locating New Facilities with Respect to Existing Facilities. *American Institute of Industrial Engineers Transactions*, 2(4):290–297, 1970.

[52] M.R. Rao. *The Rectilinear Facilities Location Problem.* Working Paper Series No. F7215, System Analysis Program, The Graduate School of Management, University of Rochester, New York, 1972.

[53] R.T. Rockafellar. Extension of Fenchel's duality theorem for convex functions. *Duke Mathematical Journal*, 33(1):81–89, 1966.

[54] R.T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, 1970.

[55] R.T. Rockafellar. On the maximality of sums of nonlinear monotone operators. *Trans. Amer. Math. Soc.*, 149:75–88, 1970.

[56] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[57] T. Strömberg. *The Operation of Infimal Convolution.* Dissertationes Math. (Rozprawy Math.), 352:58, 1996.

[58] S. Simons. *From Hahn-Banach to monotonicity.* Springer-Verlag, Berlin, 2008.

[59] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM Journal on Control and Optimization*, 38(2):431–446, 2000.

[60] R.C. Vergin and J.D. Rogers. An Algorithm and Computational Procedure for Locating Economic Activities. *Management Science*, 13:240–254, 1967.

[61] B.C. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38(3):667–681, 2013.

[62] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):409–419, 1998.

[63] E. Weiszfeld. Sur le Point pour Lequel la Somme des Distances de n Points Donnes Est Minimum. *Tohoku Mathematical Journal*, 43:355–386, 1936.

[64] G.O. Wesolowsky and R.F. Love. The Optimal Location of New Facilities Using Rectangular Distances. *Operations Research*, 19:355–386, 1971.

[65] G.O. Wesolowsky and R.F. Love. A Nonlinear Approximation Method for Solving a Generalized Rectangular Distance Weber Problem. *Management Science*, 18(11):656–663, 1972.

[66] J.A. White. *Facilities Layout and Location: an OR/MS/IE Interface*. Joint ORSA, TIMS, AIIE National Meeting, Atlantic City, New Jersey, November 1972.

[67] D.J. Wilde and C.S. Beightler. *Foundations of Optimization*. Prentice-Hall, Inc., 1967.

[68] C. Zălinescu. A comparison of constraint qualifications in infinite-dimensional convex programming revisited. *Journal of Australian Mathematical Society Series B*, 40(3):353–378, 1999.

[69] C. Zălinescu. *Convex Analysis in General Vector Spaces*. World Scientific, Singapore, 2002.

[70] H. Zhu, G. Giannakis and A.Cano. Distributed in-network channel decoding. *IEEE Transactions on Signal Processing*, 57(10):3970–3983, 2009.