“OVIDIUS” UNIVERSITY OF CONSTANTA
DOCTORAL SCHOOL OF APPLIED SCIENCES
MATHEMATICS DOCTORAL PROGRAM

DOCTORAL THESIS
SUMMARY

Ph.D. Advisor
Prof.univ.dr. **Constantin POPA**

Ph.D. Student
Cristina GHERGHINA (ȘERBAN)

CONSTANȚA, 2014

On some classes of iterative algorithms for systems of linear inequalities

Contents

Introduction

Let us consider the following system of linear inequalities

$$Ax \leq b \tag{1}$$

where $A \in I\!R^{m \times n}, b \in R^m$.

Solving the system (1) is fundamental in linear optimization problems, especially for those of reconstruction of computed tomography images, and can be done in two ways. The first way consists in transforming the system into a convex programming problem and then applying direct methods for finding the solution. There are some limitations of these methods, due to the inconsistency of the system, its large size $\{m, n \geq 10^5\}$ or to the fact that the matrix A is rare and ill-conditioned. A second way of solving system (1) involves using iterative methods, which don't consist of complex matrix calculation techniques, implementation steps being simple and relatively easy to program.

In this paper we follow the latter direction, by performing a thorough study of a special type of iterative algorithms designed to solve the inconsistent (incompatible) system (1) in a least squares sense. From this point of view, the paper is composed of six chapters followed by a section in which there are exposed the original contributions of the thesis, and a bibliography section consisting of the main materials that formed the basis for writing the paper, and the author's publications.

In the first chapter we introduce the notation that will be used throughout the thesis and the main concepts for solving systems of linear equalities and inequalities in a least squares sense.

In Chapter 2 is made a brief presentation of some classical iterative projection algorithms for solving consistent systems of linear equalities and inequalities.

Chapter 3 presents a thorough study of the iterative algorithm designed by S.P. Han and described in [12] (algorithm **H**), the first in its class of algorithms specifically developed for efficiently approximating least squares solutions of inconsistent systems of linear inequalities, this problem being formulated as follows: determine $x^* \in I\!R^n$ such that

$$\frac{1}{2} \parallel (Ax - b)_+ \parallel^2 = \min! \tag{2}$$

where

$$((Ax - b)_+)_i = \max\{(A_i x - b_i), 0\}, i = 1, \dots, m$$

Having $x^0 \in I\!R^n$ an initial datum, at each iteration, $k = 0, 1, \dots$, the algorithm **H** consists of three steps:

Step 1. Find $I_k = I(x^k) = \{i | A_i^T x^k \geq b_i\}$ and compute $d^k \in I\!R^n$ as the (unique) minimal norm solution of the linear equalities least squares problem

$$\| A_{I_k} d - (b_{I_k} - A_{I_k} x^k) \| = \min! \tag{3}$$

Step 2. Compute $\lambda^k \in I\!R$ as the smallest minimizer of the function

$$\theta(\lambda) = f(x^k + \lambda d^k), \lambda \in I\!R. \tag{4}$$

Step 3. Set $x^{k+1} = x^k + \lambda^k d^k$

The descent direction d^k in **Step 1** is computed by Han using the singular value decomposition (SVD).

Chapter 4 is dedicated to algorithms Generalized Han (**GH**), Regularized Han (**RH**) and Modified Han (**MH**), which we call of type Han because they are designed with the same structure as the one of the Han's algorithm. The Han-type algorithms differ at Step 1, the calculation of the descent direction: in algorithms **H**, **GH** and **RH** the direction d^k is computed by direct methods (SVD or QR decomposition with column pivoting), whereas the algorithm **MH** proposes to approximate it in an iterative way, which is basically justified especially for large problems. The algorithm **GH** and its properties, which arise from the study of a particular case presented by R. Bramley and B. Winnicka in [4, 5], is introduced in section 4.1. In section 4.2 is analysed the algorithm **RH** proposed by K. Yang in [27]. The analyses made in sections 4.1 and 4.2 were published by the author of this thesis in [19]. Section 4.3 is dedicated to algorithm **MH** proposed by the author C. Şerban et al, the results of this section being published in [7] and [20].

Chapter 5 includes an optimized version of the algorithm **MH**, and the theoretical analysis of an unified approach to all four Han-type algorithm presented in Chapters 3 and 4. Thus, we introduce the general form of a Han-Type algorithm, the algorithm **TH**, and give the necessary conditions that the direction d^k must satisfy in order that the algorithm **TH** to be classified as Han-type. The results of this chapter have been the subject of the paper [19].

In Chapter 6 we present the numerical experiments performed with the four algorithms on two types of problems: linear separability problems and classical and maritime transportation problems, showing that the solutions computed by algorithm **MH** are reliable. The two types of constrained optimization problem are equivalent to systems of inequalities. If the optimization problems are inconsistent, the corresponding systems of inequalities will also be inconsistent, so we may solve them by algorithms **H, GH, RH** or **MH**. The study in this section has been the subject of works [6], [21].

In closing, I wish to thank those with whom I collaborated in developing this thesis. First, I want to express all my gratitude to my scientific coordinator, Professor Constantin Popa, for the numerous discussions which helped substantially in clarifying important aspects of the current research direction, for all the books and articles offered for study and for the patience and willingness always offered. I also want to thank Professor Doina Carp, for all her support granted, and also my thesis committee members for their guidance through this process. Last but not least, I would especially like to thank my family for the continuous love, support and constant encouragement offered throughout the doctoral studies.

Chapter 1

Preliminaries

In this section we will present the main results related to the least squares formulation of inconsistent systems of linear equalities and inequalities. We will first introduce some notations, which will be used throughout the paper: $\langle \cdot, \cdot \rangle, \| \cdot \|$ will be used for the Euclidean scalar product and norm on some space $I\!R^n$, defined by

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i, \forall x = (x_i)_{i=1,\ldots,n}, y = (y_i)_{i=1,\ldots,n} \in I\!R^n \tag{1.1}$$

$$\| x \| = (\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}} \tag{1.2}$$

We also know the Cauchy-Buniakowski-Schwarz (C-B-S) inequality

$$|\langle x, y \rangle| \leq \| x \| \cdot \| y \|, \tag{1.3}$$

with equality if and only if the vectors x, y are collinear (i.e. $y = \alpha x$ for some $\alpha \in I\!R$).
A^T will denote the $n \times m$ transpose of the $m \times n$ matrix A and A^+ its Moore-Penrose pseudoinverse. By $\| A \|_2 = \sqrt{\rho\left(A^T A\right)}$ we will denote the spectral norm of A, where $\rho(B)$ is the spectral radius of the square matrix B. By A_i, A^j we will denote the i-th row and j-th column of A, respectively and by $\mathcal{N}(A), \mathcal{R}(A)$ the vector subspaces defined by

$$\mathcal{N}(A) = \{x \in I\!R^n, Ax = 0\}, \;\; \mathcal{R}(A) = \{y \in I\!R^m, y = Ax, x \in I\!R^n\}. \tag{1.4}$$

1.1 Systems of linear equalities

In this section we will present the main results concerning the least squares solutions of systems of linear equalities. Let A be an $m \times n$ matrix and $b \in I\!R^m$. We consider the system of linear equations

$$Ax^* = b \tag{1.5}$$

and denote by $S(A;b)$ the set of all its solutions. We shall call the system 1.5 consistent if $S(A;b) \neq \emptyset$, and inconsistent if $S(A;b) = \emptyset$. If the system (1.5) is inconsistent, we will say that $x^* \in I\!R^n$ is a *least squares solution* for it if

$$\frac{1}{2} \parallel Ax^* - b \parallel^2 = \min\{\frac{1}{2} \parallel Az - b \parallel^2, z \in I\!R^n\}, \tag{1.6}$$

and will denote by $LSS(A;b)$ the set of all such vectors.

Proposition 1. *([18],Prop.19) (i) The set $LSS(A;b)$ is convex, nonempty and $LSS(A;b) = \mathcal{N}(\mathcal{A}) + x_{LS}$, i.e for any $x^* \in LSS(A;b)$ we have*

$$x^* = P_{\mathcal{N}(\mathcal{A})}(x^*) + x_{LS}. \tag{1.7}$$

where $x_{LS} \in LSS(A;b)$ is given by

$$x_{LS} = A^+ b \tag{1.8}$$

If $n \leq m$ si $rank(A) = n$ then $LSS(A;b) = \{x_{LS}\}$.
(ii)([18],Prop.19) x_{LS} is the unique minimal norm solution, i.e.

$$\parallel x_{LS} \parallel \;\leq\; \parallel x^* \parallel, \; \forall x^* \in LSS(A;b). \tag{1.9}$$

(iii)([18],Prop.21 - The normal equation) We have the equivalence

$$x \in LSS(A;b) \Leftrightarrow A^T A x = A^T b \tag{1.10}$$

1.2 Systems of linear inequalities

Let now A be an $m \times n$ real matrix and $b \in I\!R^m$. We consider the system

$$Ax^* \leq b, \tag{1.11}$$

in which the inequalities are component-wise, i.e. $\sum_{j=1}^n A_{ij} x_j^* \;\leq\; b_i, \; \forall i = 1, \ldots, m$. In the consistent case of (1.11) many classes of efficient iterative solvers have been designed for its numerical solution (see for a good overview the monograph [9]). In the inconsistent case, it results that at least for one index i the corresponding inequality is violated, i.e. the set $I(x) \subset \{1, \ldots, m\}$, defined by

$$I(x) = \{i \in \{1, \ldots, m\}, \; \langle A_i, x \rangle \;\geq\; b_i\} \tag{1.12}$$

is nonempty for any $x \in I\!R^n$. Moreover, let us suppose that the set $I(x) = \{i_1, \ldots, i_p\}$ is ordered such that $i_1 < i_2 < \cdots < i_p$; then, $A_{I(x)}, b_{I(x)}$ will denote the submatrix of A with the rows $A_{i_1}, \ldots, A_{i_p}$ and the subvector of b with components $b_{i_1}, \ldots, b_{i_p}$, respectively. For any vector

$y \in I\!R^m$ we define

$$(y_+)_i = \max\{y_i, 0\} = \begin{cases} y_i, & y_i \geq 0 \\ 0, & y_i < 0. \end{cases} \tag{1.13}$$

and the convex sets by $C_i = \{x \in I\!R^n, \langle A_i, x \rangle \leq b_i\}, i = 1, \ldots, m$. The inconsistency of the system (1.11) is equivalent to $\cap_{i=1}^m C_i = \emptyset$, and we reformulate it as (see e.g. [12]): find $x^* \in I\!R^n$ a minimizer of

$$f(x) = \frac{1}{2} \parallel (Ax - b)_+ \parallel^2 . \tag{1.14}$$

where

$$((Ax - b)_+)_i = \max\{(A_i x - b_i), 0\} = \begin{cases} A_i x - b_i, & A_i x - b_i \geq 0 \\ 0, & A_i x - b_i < 0. \end{cases} \tag{1.15}$$

In [12] are also proved the following results.

Proposition 2. *(i) The objective function f from (1.14) is continuously differentiable and convex.*
(ii) The gradient of f is

$$\nabla f(x) = A^T (Ax - b)_+ = A_I^T (A_I x - b_I) \tag{1.16}$$

(iii)[12, 25] The gradient of f is Lipschitz continuous,

$$\|\nabla f(x) - \nabla f(y))\| \ \leq \ \|A\|_2^2 \ \|x - y\|, \forall x, y \in I\!R^n. \tag{1.17}$$

(iv) ([12],Theorem 2.1) A vector $x^* \in I\!R^n$ is a least squares solution of (1.11) if and only if

$$\nabla f(x^*) = A^T (Ax^* - b)_+ = 0, \tag{1.18}$$

i.e. the normal equation of an inconsistent system of linear inequalities.
(v) ([12], Theorem 2.3) There exists (at least) a least squares solution of (1.11).

The set of all least squares solutions of (1.11) will be denoted by $ILSS(A; b)$.

Theorem 1. *([12], Theorem 2.5) It exists a unique $z^* \in I\!R^m$ such that $x^* \in ILSS(A; b)$ if and only if $(Ax^* - b)_+ = z^*$. Moreover $z^* \geq 0, \ \ A^T z^* = 0$*

Chapter 2

Iterative projection algorithms for consistent systems of linear equalities and inequalities

In this chapter we briefly present some of the classical projection algorithms used to solve the consistent systems of linear equalities and inequalities.

2.1 Kaczmarz algorithm for consistent systems of linear equalities

We consider the consistent systems of linear equalities

$$Ax = b \tag{2.1}$$

The Kaczmarz method, based on orthogonal successive projections on hyperplanes $H_i = \{x \in \mathbb{R}^n | \langle A_i, x_i \rangle = b_i, i = 1, 2, ..., m\}$, was proposed by its author in [14]. The solution of system 2.1 is the unique intersection point of the hyperplanes H_i.

Kaczmarz Algorithm Let us consider $x^0 \in \mathbb{R}^n$ arbitrary fixed; for $k = 0, 1, \ldots$, do

$$x^{(k+1)} = P_{H_{i_k}}(x^{(k)}) = x^{(k)} + \frac{b_{i_k} - A_{i_k}^T x^{(k)}}{\|A_{i_k}\|^2} A_{i_k} \tag{2.2}$$

2.2 Cimmino algorithm for consistent systems of linear equalities

The Cimmino algorithm is an iterative method proposed by G. Cimmino in [10], in which simultaneous projections are made on all the hyperplanes $H_i = \{x \in \mathbb{R}^n | \langle A_i, x_i \rangle = b_i, i = 1, 2, ..., m\}$.

Instead of orthogonal projections, the Cimmino algorithm uses the orthogonal reflections with respect to the hyperplanes H_i.

Cimmino Algorithm (I) Let us consider $x^0 \in I\!R^n$ arbitrary fixed; for $k = 0, 1, \ldots$, do

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \frac{2}{\mu} \sum_{i=1}^{n} m_i \frac{b_i - A_i^T x^{(k)}}{\|A_i\|^2} A_i \\ &= x^{(k)} + A^T D(b - Ax^{(k)}) \end{aligned}$$

where $\{m_i\}_{i=\overline{1,n}}$ are positive quantities called *masses*, $\mu = \sum_{i=1}^{n} m_i$ and $D = \frac{2}{\mu} diag(\frac{m_i}{\|A_i\|^2})$.

2.3 Cimmino algorithm for consistent systems of linear inequalities

This algorithm has the following form:

Cimmino Algorithm (II):

$$x^{k+1} = x^k + \frac{2}{\mu_k} \sum_{i=1}^{m} m_i c_i^k A_i \tag{2.3}$$

where $m_i = \frac{\hat{m}_i}{\sum_{i=1}^{m} \hat{m}_i}$ are called generalized masses, $\{\hat{m}_i\}_{i=\overline{1,n}}$ are positive quantities called *masses*, $\sum_{i=1}^{m} m_i = 1$, $0 < m_i < 1, \forall i \in I$, $c_i^k = \min\{0, \frac{b_i - \langle A_i, x^k \rangle}{\|A_i\|^2}\}$ and

$$\mu_k = \begin{cases} \sum_{i \in I_k} m_i, \text{ daca } |I_k| \geq 2 \\ \quad 1, \text{ daca } |I_k| = 1 \end{cases}$$

with $I_k = \{i | c_i^k < 0\}$ the set of indices for which $x^k \notin L_i = \{x \in I\!R^n | \langle A_i, x \rangle \leq b_i\}$

2.4 Richardson algorithm for consistent systems of linear equalities

Let us consider the residual vector $r^k = b - Ax^k$, $I_k = \{i | \langle A_i, x \rangle > b_i\}$ and the diagonal matrix D^k defined by

$$D_{ii}^k = \begin{cases} 1, \text{ daca } i \in I_k \\ 0, \text{ daca } i \notin I_k \end{cases}$$

The Richardson algorithm has the following form:

Richardson Algorithm Let $M = (m_{ij})$ be a positive definite matrix with nonzero elements and

$x^0 \in I\!R^n$ arbitrary fixed. If $r^k \geq 0$ then STOP. Otherwise,

$$x^{k+1} = x^k + \alpha_k A^T M^k (b - Ax^k) \tag{2.4}$$

where $M^k = D^k M D^k$ and the parameters $\{\alpha_k\}$ have the property $0 < \alpha_k < \frac{2}{\rho(A^T M^k A)}$.

2.5 Lent-Censor Algorithm for consistent systems of linear inequalities

The Lent-Censor algorithm is proposed in [16] and represents an extended form of the Hildreth algorithm [13], a useful method for solving optimization problems of large dimensions. First of all, a set of relaxation parameters was introduced in Hildreth algorithm, $\{r^{(k)}\}, 0 < r^{(k)} < 2$, which does not change in any way its convergence, a fact proven in [16]. The second change made on Hildreth algorithm consists in changing the way the rows of the matrix A are considered at each iteration, thus having an *almost cyclic* control over them, less restrictive than the *cyclic* one.

Definition 1. *Let's consider the finite set $I = \{1, 2, ..., m\}$. The sequence $\{i_k\}_{k=0}^{\infty}$ is almost cyclic on I if:*
a) $i_k \in I$, for all $k \geq 0$;
b) It exists C such as for all

$$k \geq 0, I \subseteq \{i_{k+1}, ..., i_{k+C}\}$$

A sequence almost cyclic on $\{1, 2, ..., m\}$ is cyclic if $C = m$. Assuming that $A_i \neq 0 \; \forall i \in I = \{1, 2, ..., m\}$, the Lent-Censor algorithm for the problem

$$\begin{cases} \min \frac{1}{2} \|x\| \\ Ax \leq b \end{cases} \tag{2.5}$$

has the following form:

Lent-Censor Algorithm Let us consider $z^0 \in I\!R^m_+$ and $x^0 = -A^T z^0$; for $k = 0, 1, 2, \ldots$ do

$$\begin{aligned} x^{k+1} &= x + c^k A_{i_k} \\ z^{k+1} &= z - c^k e_{i_k} \\ c^k &= \min(z^k_{i_k}, r^k \frac{b_{i_k} - \langle A_{i_k}, x^k \rangle}{\|A_{i_k}\|^2}) \end{aligned}$$

where

$$\begin{aligned} \{i_k\}_{k=0}^{\infty} \text{ almost-cyclic on } I &= \{1,2,...,m\} \\ 0 &< r^k < 2 \\ e_i &= (..0...\overset{.........i...........}{1}....0..)^T \end{aligned}$$

Let's consider x^k, z^k , $L_{i_k} = \{x \in I\!R^n | \langle A_{i_k}, x\rangle \leq b_{i_k}\}$ si $H_{i_k} = \{x \in I\!R^n | \langle A_{i_k}, x\rangle = b_{i_k}\}$ the subspace and the hyperplane, respectively, determined by i_k-th inequality of system (2.5).

- If $x^k \notin L_{i_k} \Leftrightarrow x^{k+1}$ is the orthogonal projection of x^k on $L_{i_k}, x^{k+1} \in H_{i_k}$
- If $x^k \in H_{i_k}$, then $c^k = 0,$ so $x^{k+1} = x^k$
- If $x^k \in int(L_{i_k})$ then

- $c^k = z^k_{i_k} \Rightarrow x^{k+1} = x + z^k_{i_k} A_{i_k}$, so x^{k+1} is on an orthogonal direction on H_{i_k} given by $z^k_{i_k} A_{i_k}$, or
- $c^k = \frac{b_{i_k} - \langle A_{i_k}, x^k\rangle}{\|A_{i_k}\|^2} \Rightarrow x^{k+1} = x^k + \frac{b_{i_k}t - \langle A_{i_k}, x^k\rangle}{\|A_{i_k}\|^2} A_{i_k}$ so x^{k+1} is the orthogonal projection of the vector x^k on H_{i_k}, so x^{k+1} is the minimal norm solution.

Chapter 3

Han algorithm for systems of linear inequalities

The only algorithm specifically designed for solving arbitrary systems of linear inequalities in a least squares sense was developed by S.-P. Han [12]. This algorithm requires finding the minimum norm least squares solution to systems $A_I x = b_I$, where A_I is a submatrix of A consisting of some rows of A, and has the following form:

Algorithm H. *Let* $x^0 \in I\!R^n$ *be an initial datum; for* $k = 0, 1, \dots$ *do:*

Step 1. *Find* $I_k = I(x^k) = \{i | A_i^T x^k \geq b_i\}$ *and compute* $d_{LS}^k \in I\!R^n$ *as the (unique) minimal norm solution of the linear equalities least squares problem*

$$\| A_{I_k} d - (b_{I_k} - A_{I_k} x^k) \| = \min! \qquad (3.1)$$

Step 2. *Compute* $\lambda^k \in I\!R$ *as the smallest minimizer of the function*

$$\theta(\lambda) = f(x^k + \lambda d_{LS}^k), \lambda \in I\!R. \qquad (3.2)$$

Step 3. *Set* $x^{k+1} = x^k + \lambda^k d_{LS}^k$.

According to the existence and computation of the smallest minimizer from **Step 2**, we provided in [7] the following result (which also gives us an algorithmic procedure to find it).

Proposition 3. *The smallest minimizer of the function* $\theta(\lambda)$ *from (3.2) always exists and can be efficiently and accurately computed in each iteration of the algorithm* **H**.

Theorem 2. *(i) ([12], Lemma 4.1) For the sequence* $(x^k)_{k \geq 0}$ *generated with the algorithm* **H**, *the gradient of the objective function (see (1.14)) satisfies*

$$\nabla(x^k) = A^T(Ax - b)_+ = A_I^T(A_I x^k - b_I) = -A_I^T A_I d_{LS}^k. \qquad (3.3)$$

(ii) ([12], Theorem 4.2) d^k_{LS} *is a descent direction, i.e.*

$$\langle \nabla(x^k), d^k_{LS} \rangle = - \left\| A_I d^k_{LS} \right\|^2 . \tag{3.4}$$

Theorem 3. *([12], Theorem 4.4) Let* $(x^k)_{k\geq 0}$ *be the sequence generated by algorithm* **H** *from any* $x^0 \in I\!R^n$*. Then either* $\nabla f(x^{\bar{k}}) = 0$*, for some* $\bar{k} < \infty$ *or* $\lim\limits_{k\to\infty} \nabla f(x^k) = 0.$

Theorem 4. *([12],Theorem 4.6) Let* $(x^k)_{k\geq 0}$ *be the sequence generated by algorithm* **H***,* $z^k = (Ax^k - b)_+$ *and* z^* *the unique vector from Theorem 1. Then*

$$\lim_{k\to\infty} z^k \; = \; z^*. \tag{3.5}$$

The main result related to Han's algorithm will be presented below (for proof details see the original paper [12]).

Theorem 5. *([12], Theorem 4.10) For any* $m \times n$ *matrix* A*, any right-hand side* $b \in I\!R^m$ *and any initial datum* $x^0 \in I\!R^n$*, Han's algorithm* **H** *produces a least squares solution of the system (1.11) in a finite number of steps (in exact arithmetic).*

Remark 1. *(i) It results that Han's algorithm can be (theoretically) also used for consistent systems of the form (1.11); but, in this case there are many other more efficient methods (see e.g. [9]).*
(ii) From a practical view point, because of round-off errors, Han's algorithm becomes an iterative method.
(iii) At each iterative step of the algorithm ***H****, a Singular Value Decomposition (SVD) of the matrix* A_{I_k} *needs to be computed. When the matrix is large, rare or ill-conditioned, SVD is not an efficient method in terms of computation cost, so other versions of the Han algorithm have been developed, in which the descent direction from* ***Step 1*** *is computed in a more effective way.*

Chapter 4

Han-type algorithms for systems of linear inequalities

In this chapter we present three versions of Han algorithm, together with their specific way of computing the descent direction in **Step 1**.

4.1 Generalized Han algorithm (GH)

One question that can occur related to Han's algorithm is: can we use instead of the minimal norm solution d^k_{LS} in **Step 1** any other solution of the problem 3.1 ? An answer was given in the paper [4] (see also [5]). We will present it below, a little bit more generally than in the original paper. First of all we will consider a Generalized version of Han's algorithm.

Algorithm GH *Let $x^0 \in I\!R^n$ be an initial datum; for $k = 0, 1, \dots$ do:*

Step 1. *Find $I_k = I(x^k)$ and compute $d^k \in I\!R^n$ an arbitrary least squares solution of the problem 3.1:*

$$\parallel A_{I_k} d - (b_{I_k} - A_{I_k} x^k) \parallel = \min!$$

Step 2. *Compute $\lambda^k \in I\!R$ as the smallest minimizer of the function*

$$\theta(\lambda) = f(x^k + \lambda d^k), \lambda \in I\!R. \tag{4.1}$$

Step 3. *Set $x^{k+1} = x^k + \lambda^k d^k$.*

The least squares solution d^k from **Step 1** can not be completely arbitrary, as mentioned in the result below.

Theorem 6. *Let us suppose that it exists a constant $C \geq 0$, independent on the iteration k, such that*

$$\parallel d^k \parallel \;\; \leq \; C \parallel d^k_{LS} \parallel, \; \forall k \geq 0. \tag{4.2}$$

*Then the same conclusion of Theorem 5 holds for the algorithm **GH**.*

The proof of this theorem starts with the remark that a general least squares solution of problem 3.1 is

$$d^k = P_{\mathcal{N}_{A_{I_k}}}(d) + d^k_{LS}$$

hence

$$A_{I_k} d^k = A_{I_k} d^k_{LS}, \ \forall k \geq 0 \tag{4.3}$$

We also have

$$\nabla f(x^k) = -A^T_{I_k} A_{I_k} d^k_{LS} = -A^T_{I_k} A_{I_k} d^k, \tag{4.4}$$

and

$$\langle \nabla f(x^k), d^k \rangle = - \parallel A_{I_k} d^k \parallel^2 \ \leq \ 0. \tag{4.5}$$

In [4, 5], R. Bramley and B. Winnicka proposed as d^k for algorithm **GH** a least squares solution of the problem 3.1 computed by a QR decompositon with column pivoting; thus, the relation (4.3) is satisfied. It is also shown, by using a general result of QR decomposition proven in [11], that the Theorem 6 holds.

4.2 Regularized Han algorithm (RH)

In [27], K. Yang comments on Han's algorithm by considering its major drawback in the fact that, in each iteration, initial objective function from (1.14) is replaced by

$$f(x^k) = \frac{1}{2} \parallel (Ax^k - b)_+ \parallel^2 = \frac{1}{2} \parallel A_{I_k} x^k - b_{I_k} \parallel^2$$

In this way, many originally satisfied constraints might be violated in the new iterative solution x^k. Regarding this aspect, he also proposed to use the complement of the set I_k, denoted by J_k and characterized by

$$J_k = J(x^k) = \{i \in \{1, \dots, m\}, A_i x^k < b_i\} = \{1, \dots, m\} \setminus I_k, \tag{4.6}$$

together with a diagonal weights matrix

$$W_k = diag(w^k_1, \dots, w^k_{q_k}), \ w^k_i \geq 0 \tag{4.7}$$

where q_k is the number of elements in the set $J_k, \forall k \geq 0$. With these ideas, Yang designed the Regularized version of Han's algorithm from below.

Algorithm RH *Let* $x^0 \in I\!R^n$ *be an initial datum; for* $k = 0, 1, \dots$ *do:*

Step 1. *Find* $I_k = I(x^k), J_k = J(x^k)$ *as before and compute* $\tilde{d}^k \in I\!R^n$ *the minimal norm solution of the (regularized) linear least squares problem*

$$\parallel A_{I_k} d - (b_{I_k} - A_{I_k} x^k) \parallel^2 + \parallel W_k A_{J_k} d \parallel^2 = \min! \tag{4.8}$$

Step 2. *Compute* $\lambda^k \in I\!R$ *as the smallest minimizer of the function*

$$\theta(\lambda) = f(x^k + \lambda \tilde{d}^k), \lambda \in I\!R. \tag{4.9}$$

Step 3. *Set* $x^{k+1} = x^k + \lambda^k \tilde{d}^k$.

The next results, proved in [27] shows properties of the algorithm **RH**. We extended the proofs of these results with elements that later helped us formulate the unified approach of Han-type algorithms (section 5.2)

Proposition 4. *(i)([27], Lemma 10.1) The descent direction* $\tilde{d}^k$ *satisfies*

$$(A_{I_k}^T A_{I_k} + A_{J_k}^T W_k^2 A_{J_k})\tilde{d}^k = A_{I_k}^T (b_{I_k} - A_{I_k} x^k) = -\nabla f(x^k) \tag{4.10}$$

and

$$\tilde{d}^k = -(A_{I_k}^T A_{I_k} + A_{J_k}^T W_k^2 A_{J_k})^+ \nabla f(x^k). \tag{4.11}$$

(ii) ([27], Lemma 9.1) The vector

$$\hat{x}^k = x^k + \tilde{d}^k \tag{4.12}$$

is a least squares solution of the regularized problem

$$\| A_{I_k} x - b_{I_k} \|^2 + \| W_k A_{J_k}(x - x^k) \|^2 = \min! \tag{4.13}$$

Remark 2. *If* $W^k = 0, \forall k \geq 0$ *then* $\tilde{d}^k = d_{LS}^k$ *and we get the algorithm* **H***.*

Proposition 5. *(i) ([27], Corollary 10.2) We have the equality*

$$\langle \nabla f(x^k), \tilde{d}^k \rangle = -(\| A_{I_k} \tilde{d}^k \|^2 + \| W_k A_{J_k} \tilde{d}^k \|^2) \ \leq \ 0. \tag{4.14}$$

(ii) ([27], Lemma 10.4) If $x^0 \in I\!R^n$ *is the initial datum in the algorithm* **RH** *then*

$$\| \nabla f(x^k) \|^2 \ \ \leq \ 2 \| A \|_2^2 \ f(x^0), \forall k \geq 0. \tag{4.15}$$

The next result show that the Theorem 3 holds for algorithm **RH** as well.

Theorem 7. *([27], Theorem 10.5) Let* $x^0 \in I\!R^n$ *be arbitrary fixed, and* $(x^k)_{k\geq 0}$ *the sequence generated by the algorithm* **RH***. Then, either it exists* $k_0 \geq 0$ *such that* $\nabla f(x^{k_0}) = 0$*, or* $\lim_{k\to\infty} \nabla f(x^k) = 0$*.*

4.3 Modified Han algorithm (MH)

In many practical applications (see e.g. [9]), the matrix A is large and sparse, and so can be the submatrices A_{I_k} in Han's algorithm iterations. Then, the use of a direct method for computing the

d_{LS}^k or d^k solutions of the problem 3.1 can be affected by round-off errors, fill-in process in A_{I_k} and a (too) big computational time. One way to overcome such a difficulty would be to approximate the "directly computed" solutions d_{LS}^k or d^k, with one obtained with an iterative solver for problems of the form 3.1. Having all these aspects in mind, in [7] we proposed a Han-type algorithm based on an iterative approximation of d_{LS}^k and we proved that the Theorems 3 and 4 also hold for the sequence $(x^k)_{k\geq 0}$ generated by algorithm **MH**.
Concerning this aspect, let **ALG** be an iterative algorithm which approximates the minimal norm solution of a linear least squares problem of the form

$$\parallel Fx - c \parallel = \min!, \tag{4.16}$$

where F is an arbitrary rectangular matrix and c an appropriate vector.

Algorithm MH. *Let $x^0 \in I\!R^n$ be an initial datum; for $k = 0, 1, \ldots$ do:*
Step 1. *Find $I_k = I(x^k)$ and compute an approximation $d^{k,j} \in I\!R^n$ of the minimal norm solution d_{LS}^k of the problem 3.1*

$$\parallel A_{I_k} d - (b_{I_k} - A_{I_k} x^k) \parallel = \min!$$

by performing $j \geq 1$ iterations of the algorithm **ALG***, with 0 as initial approximation*
Step 2. *Compute $\lambda^{k,j} \in I\!R$ as the smallest minimizer of*

$$\theta(\lambda) = f(x^k + \lambda d^{k,j}), \lambda \in I\!R$$

Step 3. *Set $x^{k+1} = x^k + \lambda^{k,j} d^{k,j}$.*

Theorem 8. *Let us suppose that there exist an integer $J \geq 1$, independent on the iteration index k, such that*

$$\parallel A_{I_k} d^{k,J} \parallel \geq \frac{1}{2} \parallel A_{I_k} d_{LS}^k \parallel, \ \parallel A_{I_k}(d^{k,J} - d_{LS}^k) \parallel \leq \frac{1}{2} \parallel A_{I_k} d^{k,J} \parallel,$$

$$\parallel d^{k,J} - d_{LS}^k \parallel \leq \frac{1}{2} \parallel A_{I_k} d^{k,J} \parallel, \ \forall k \geq 0, \tag{4.17}$$

where d_{LS}^k is the minimal norm solution of 3.1 and $d^{k,J}$ the approximation generated after J iterations of **ALG** *in Step1 of* **MH***. Then the conclusions of Theorems 3 and 4 hold for the sequence $(x^k)_{k\geq 0}$ generated by the algorithm **MH**.*

Related to the assumption (4.17), we can provide the following result, published in [7]:

Theorem 9. *If as ALG in **Step 1** of the algorithm **MH** we use the Kaczmarz Extended (KE) algorithm from [17], with zero as initial approximation, then the inequalities from (4.17) hold with $J \geq 1$ independent on the iteration k.*

This proves that the algorithm **MH** allows resolving inconsistent system 1.11 in a least squares sense. Moreover, we give an example of an iterative algorithm, Extended Kaczmarz algorithm and we demonstrated that it can be successfully used in **Step 1** of the algorithm **MH**.

Chapter 5

The unified approach of Han-type algorithms

From the study of the algorithms **H**, **GH, RH** and **MH**, a series of elements have emerged, helping us to develop an unified approach of the Han-type algorithms. Next, we give the form of an algorithm of Type Han (**TH**), and the conditions it has to satisfy. This general algorithm will include as particular cases the four algorithms considered in the paper: **H, GH, RH** and **MH.**

5.1 An optimized version of Modified Han (MH) algorithm

The **MH** algorithm presented in section 4.3 can be optimized by reducing the number of conditions in (4.17). This change also allows us to consider it for the unified approach we aim to produce. The results of this section have been published in [19].
Next, we give the optimized version of Theorem 8.

Theorem 10. *Let us suppose that there exists an integer $J \geq 1$, and constants $C_1 \in [0,1),\ C_2 \geq 0$, all independent on the iteration index k, such that*

$$\| A_{I_k}(d^{k,J} - d^k_{LS}) \| \leq C_1 \| A_{I_k} d^{k,J} \|, \quad \| d^{k,J} - d^k_{LS} \| \leq C_2 \| A_{I_k} d^{k,J} \|, \ \ \forall k \geq 0, \tag{5.1}$$

where d^k_{LS} is the minimal norm solution of 3.1 and $d^{k,J}$ the approximation generated after J iterations of **ALG** *in* ***Step 1*** *of algorithm* **MH**. *Then the conclusions of Theorems 3 and 4 hold for the sequence $(x^k)_{k\geq 0}$ generated by the algorithm* **MH**.

For the unification procedure from section 5.2 we need the following modified version of the above theorem.

Corollary 1. *The conclusion of Theorem 10 remains true if we replace the second inequality from (5.1)*

$$\| A_{I_k}(d^{k,J} - d^k_{LS}) \| \leq C_1 \| A_{I_k} d^{k,J} \|, \quad \| d^{k,J} - d^k_{LS} \| \leq C_2 \| A_{I_k} d^{k,J} \|, \ \ \forall k \geq 0,$$

by

$$\| d^{k,J} - d^k_{LS} \| \leq \hat{C}_2 \| A_{I_k} d^k_{LS} \|, \ \ \forall k \geq 0. \tag{5.2}$$

Related to the assumption (5.2), we can provide the following result, proven in [7], which assures us that the algorithm Kaczmarz Extended (KE) can be successfully used in the calculus of the descent direction in **Step 1** of algorithm **MH.**

Theorem 11. *If as ALG in **Step 1** of the algorithm **MH** we use the Kaczmarz Extended (KE) algorithm from [17], with zero as initial approximation, then the first inequality from (5.1) and the inequality from (5.2) hold with $C_1 = \frac{1}{2}, C_2 = \frac{1}{4}$ and $J \geq 1$ independent on the iteration k.*

5.2 The unified approach

The results of this section have been published in [19]. In order to obtain an unified point of view related to the Han-type algorithms, we will introduce three general assumptions on the descent direction D^k generated in **Step 1**, and x^k from **Step 3.**

Assumption 1.

$$\langle \nabla f(x^k), D^k \rangle \leq 0, \ \ \forall\, k \geq 0. \tag{5.3}$$

Assumption 2. It exists a constant $C \geq 0$, independent on the iteration k such that

$$\| D^k \| \leq C \| \nabla f(x^k) \|, \ \ \forall\, k \geq 0. \tag{5.4}$$

Assumption 3.

$$\text{If } \lim_{k \to \infty} \langle \nabla (x^k), D^k \rangle = 0, \text{ then } \lim_{k \to \infty} \nabla f(x^k) = 0. \tag{5.5}$$

In what follows we show that the descent directions $d^k_{LS}, d^k, \tilde{d}^k$ and $d^{k,J}$ corresponding to the algorithms **H, GH, RH** and **MH**, respectively, satisfy the above assumptions.

Proposition 6. *The assumption (5.5) holds for $D^k \in \{d^k_{LS}, d^k, \tilde{d}^k, d^{k,J}\}$.*

Proposition 7. *The assumptions (5.3) and (5.4) hold for $D^k \in \{d^k_{LS}, d^k, \tilde{d}^k, d^{k,J}\}$.*

Taking into consideration all the above, we give the following form of the algorithm of Type Han (**TH**):

Algorithm TH*Let $x^0 \in I\!R^n$ be an initial datum; for $k = 0, 1, \dots$ do:*

Step 1. *Find $I_k = I(x^k)$ and compute $D^k \in I\!R^n$ a descent direction.*

Step 2. *Compute $\lambda^k \in I\!R$ as the smallest minimizer of the function*

$$\theta(\lambda) = f(x^k + \lambda D^k), \lambda \in I\!R. \tag{5.6}$$

Step 3. *Set $x^{k+1} = x^k + \lambda^k D^k$.*

The next theorem is the main result of our approach because it shows that the algorithm **TH** satisfies Theorem 3.

Theorem 12. *Let us suppose that the descent direction D^k in Step 1 of the algorithm* **TH** *satisfies $D^k \in \{d^k_{LS}, d^k, \tilde{d}^k, d^{k,J}\}$. Then, the conclusion of the Theorem 3 holds for the sequence $(x^k)_{k\geq 0}$ generated by* **TH**.

We proved in this section that algorithms **H, GH, RH** and **MH** satisfy Theorem 3. Algorithms **H, GH** and **MH** have in addition the properties mentioned in Theorem 4, whereas **H, GH** also those from Theorem 5. Work is in progress in order to prove such results for the algorithms **RH** and **MH**, and to extend the unification results from this paper to cover also these results. Till then, the successful experiments from the papers [7] and [20] are encouraging us for these theoretical developments.

Chapter 6

Applications

To test the Han-type algorithms discussed in this paper, we have implemented each of the four algorithms **H, GH, RH** and **MH** in Matlab R2010a, using the built-in Matlab functions as *pinv* and *qr* to compute the descent direction for algorithms **H, GH, RH**, whereas for **MH** we used the Kaczmarz Extended (KE) algorithm from [17] as **ALG** in Step 1. In order to solve the problem 4.8 of **RH** algorithm, we set $w_k = 10^{-3}, \forall k$, this value being determined experimentally as the most suitable value of the weights. For all algorithms, we computed λ^k, the smallest minimizer of the convex function θ, as in Proposition 3. We also made use of the build-in Matlab implementation of Simplex algorithm, $linprog$. All runs with respect to the four algorithms are started with the initial datum $x_0 = (0, \ldots, 0)^T$ and are terminated if at the current iteration x^k satisfy $A^T(Ax^k - b)_+ \parallel \ \leq \ 10^{-15}$. The results of our numerical experiments have been published in [6, 7, 21].

6.1 Linear separability problems

6.1.1 Linear separability

Lemma 1. *([2])Two sets $\mathcal{A} = \{A^1, A^2, \ldots, A^m\} \subseteq I\!R^n$, $\mathcal{B} = \{B^1, B^2, \ldots, B^k\} \subseteq I\!R^n$ are linearly separable if and only if there exist $w \in I\!R^n$ and $\gamma \in I\!R$ such that*

$$\langle w, A^i\rangle - \gamma \geq 1 \text{ and } \langle w, B^j\rangle - \gamma \leq -1, \forall i \in 1, \ldots, m, \forall j \in 1, \ldots, k$$

The point sets $\mathcal{A}$ and $\mathcal{B}$ represented by the matrices $A \in R^{m\times n}$ and $B \in R^{k\times n}$, respectively, are linearly separable if it exists $v \in R^n$ such that

$$\min_{1\leq i\leq m} \langle A_i, v\rangle > \max_{1\leq i\leq k} \langle B_i, v\rangle, \tag{6.1}$$

which is equivalent with: there exist $w \in R^n$ and $\gamma \in R$ such that (see also Lemma 1)

$$Aw \geq e_m\gamma + e_m, Bw \leq e_k\gamma - e_k, e_i \in I\!R^i, e_i = (1, \ldots, 1) \tag{6.2}$$

In [1], the linear separability problem in $I\!R^n$ is formulated as a problem which minimizes the points of $\mathcal{A}$ and $\mathcal{B}$ incorrectly classified by the separating hyperplane $\omega x = \gamma$:

$$\min_{w,\gamma} \frac{1}{m} \parallel (-Aw + e\gamma + e)_+ \parallel_1 + \frac{1}{k} \parallel (Bw - e\gamma + e)_+ \parallel_1, \tag{6.3}$$

where $\parallel z \parallel_1 = \sum_{i=1}^{n} |z_i|, z \in I\!R^n$. According to Lemma 2.2 in [1], the optimization problem 6.3 will always generate a separating hyperplane $wx = \gamma$ for linearly separable sets $\mathcal{A}$ and $\mathcal{B}$; as for linearly inseparable sets, an optimal separating hyperplane will be generated such that it minimizes the number of points in $\mathcal{A}$ which lie in $\{x, \langle w, x \rangle < \gamma + 1\}$ and the number of points in $\mathcal{B}$ which lie in $\{x, \langle w, x \rangle > \gamma - 1\}$.
In [4, 5], the optimization problem 6.3 is expressed as a linear system of inequalities

$$Gw \leq g \tag{6.4}$$

which my be solved in a least-squares sense:

$$min_{\tilde{w}} \parallel (G\tilde{w} - g)_+ \parallel^2 \tag{6.5}$$

where $G = \begin{pmatrix} -A & e_m \\ B & -e_k \end{pmatrix}, \tilde{\omega} = (\omega^T, \gamma)^T$ si $g = \begin{pmatrix} -e_m \\ -e_k \end{pmatrix}$.
Next, we shall present computational comparisons on several linear separability problems, following the papers [4] and [5]. All tests were performed on finite sets of points, linearly inseparable. The measure of error used for both sets of problems was given by counting the number of incorrectly classified points (ICP): $A\omega > \gamma e_m$ and $B\omega \leq \gamma e_k$. The results of these tests have been published in [7, 19].

6.1.2 Numerical experiments

Two-dimensional problems. The first set of linear separability problems addressed is the two-dimensional one, and consists of two cases: the first is a smaller square inside the unit square (Square problem), and the second are two triangles that overlap (Triangle problem). For the Square problem, a total of two hundred data points were used, randomly distributed inside the two squares, two-thirds of them being inside the unit square. As for the Triangle problem, each triangle had one hundred data points, also randomly generated. To produce all these uniformly distributed data points, we used the build-in Matlab random number generator, *rand*. Table 6.1 presents the results obtained for the Square and Triangle problems.

Problem	**ICP**	**ICP %**
$Square$	13	6.5
$Triangle$	27	13.36

Table 6.1: The number of incorrectly classified points (**ICP**) for the Triangle and the Square problems. Algorithms: **H**, **GH**, **RH** and **MH**

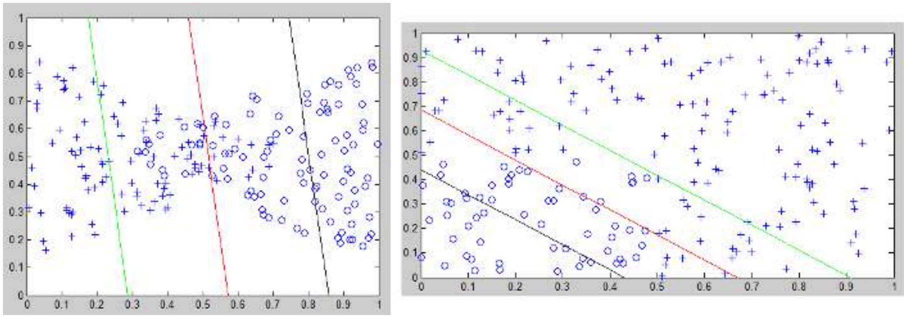

Figure 6.1: Triangle problem (left); Square problem (right) A:+, B : ○

Database tests. The second set of problems contains two real world databases: the Wisconsin Breast Cancer Database (WBC) and the Cleveland Heart Disease Database (CHD) (see [4]). Both files were downloaded from [28]. Each data point in the WBC and CHD databases has 9, respectively, 13 components, corresponding to experimental measurements. After we discarded the data points that had missing measurements, the WBC database was comprised of 551 data points of which 346 were benign (set $\mathcal{A}$) and 205 were malignant (set $\mathcal{B}$). The CHD database remained with 297 data points of which 137 were negative (set $\mathcal{A}$) and 160 were positive (set $\mathcal{B}$). As in [4], each set was divided randomly into a training set with 67% of the data points and a testing set with the remaining 33%. This division was carried out by a random permutation of integers $i \in [1, |S|], S \in \{\mathcal{A}, \mathcal{B}\}$ generated with the build-in Matlab function, *randperm*. The main idea was to apply the solvers on the training set, and then to test the resulting hyperplane on the testing set.

For each database, we randomly partitioned the data sets and ran the solvers ten times and computed the incorrectly classified points $ICP_i^S, i = 1, \dots, 10$; then we obtained the average and the percentage of the results: $MICP^S = \frac{1}{10} \sum_{i=1}^{10} ICP_i^S$ and $ICP\% = \frac{MICP^S}{|S|} * 100$.

Baza de date	**ICP% training**	**ICP% test**
WBC	3.30	3.50
CHD	15.30	16.19

Table 6.2: ICP% for databases WBC and CHD. Algorithms: **H**, **GH**, **RH** and **MH**

6.2 Unbalanced and inconsistent classical and maritime transportation problems

The results of this section have been published in [6, 21].
The classical transportation problem involves sources $(S_i)_{i\in\{1,\ldots,n\}}$, where supplies $(s_i)_i = 1, \ldots, n$ of some goods are available, and destinations $(D_j)_{j\in\{1,\ldots,m\}}$, where some demands $(d_j)_{j=1,\ldots,m}$ are requested ([15]). If we denote by $x_{ij}, i = 1, \ldots, n, j = 1, \ldots, m$ the number of units transported from source S_i to destination D_j, we get the following mathematical model of the (classical) transportation problem:

$$\begin{aligned} \text{min} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} \geq d_j; \sum_{j=1}^{m} x_{ij} = s_i; x_{ij} \geq 0 \quad i = 1, \ldots, n, j = 1, \ldots m \end{aligned} \tag{6.6}$$

We will consider in this paper the unbalanced case, i.e. $\sum_{i=1}^{n} s_i < \sum_{j=1}^{m} d_j$ for which the linear program (6.6) becomes inconsistent (i.e. the set of feasible solutions is empty).
The maritime container transportation is a transportation problem subject to the following additional hypothesis: (**M1**) the unit of cargo (container) has different capacities, so the cost of a unit of transport is different for a particular route. The most common unit used in cargo transportation is TEU (Twenty-foot Equivalent Unit), corresponding to a 20-foot-long (6.1 m) intermodal container; (**M2**) the destinations become in this case warehouses with specific dimensions; hence, the number of containers transported to a given warehouse will be restricted by these dimensions.
We analysed in our numerical experiments the following two variants of the unbalanced and inconsistent transportation problem described in Table 5 from [6], for which we supposed that there exist 7 sources, $S_i, i = 1, \ldots, 7$, and 7 destinations, $D_j, j = 1, \ldots, 7$. In the case of maritime transportation, the destinations are warehouses and we will assume that all the warehouses are rectangular buildings, with known length (L), width (W) and height(H) (Table 3 in [6]). Thus, the containers will be stored on superposed rows in each warehouse. Here are the problems considered:
Problem P1 - a 2-index unbalanced and inconsistent transportation problem; (x_{ij}) represents the quantity of commodity shipped from source S_i to destination D_j. The mathematical model of this problem is the following (see inequalities from (6.6)):

$$\sum_{i=1}^{7} x_{ij} \geq d_j, j = 1, \ldots, 7; \sum_{j=1}^{7} x_{ij} = s_i, i = 1, \ldots, 7 \tag{6.7}$$

Problem P2 - a 2-index inconsistent maritime container transportation problem; (x_{ij}) is the number of containers of $Type(i)$ shipped from S_i to warehouse D_j and we assume that each source S_i provides only one category of containers (see Table 2 in [6]). The mathematical model of this problem is the following:

$$\sum_{i=1}^{7} l_i x_{ij} \leq T(j), j = 1, \dots, 7 \tag{6.8}$$

$$\sum_{i=1}^{7} T_{w_i} x_{ij} \geq d_j, j = 1, \dots, 7; T_{w_i} \sum_{j=1}^{7} x_{ij} = s_i, i = 1, \dots, 7. \tag{6.9}$$

where (6.8) represents the storage restrictions imposed by the dimensions of each warehouse, (6.9) are the inequalities from (6.6), l is the length of each type of container (see Table 2, column 2 in [6]), and T_{w_i} is the TEU's equivalent weights of the container types (see Table 2, last column in [6]).

Problem P3 - a 3-index inconsistent maritime container transportation problem; (x_{ij}) is the number of containers of $Type(i)$ shipped from S_i to warehouse D_j and we assume that each source S_i provides all types of containers $Type(l), l = 1, \dots, 7$ (see Table 2 in [6]). The mathematical model of this problem is the following:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{l=1}^{7} c_{ijl} x_{ijl} = \min \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{l=1}^{7} c_{ij} q_l x_{ijl} \tag{6.10}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{l=1}^{7} x_{ijl} \geq d_j, j = 1, \dots, m; \sum_{j=1}^{m} \sum_{l=1}^{7} x_{ijl} = s_i, i = 1, \dots, n$$

$$x_{ijl} \geq 0, i = 1, \dots, n, j = 1, \dots m, l = 1, \dots, 7$$

where: c_{ijl} is the transportation cost of one container of $Type(l)$ from source S_i to warehouse D_j; c_{ij} is the transportation cost of one TEU (one unit of commodity) from source S_i to warehouse D_j; $q = (q_1, q_2, \dots, q_7)$ holds the TEU's equivalent weights of the container types considered (see last column of Table 2 in [6]).

After several steps of processing (see [6]), we get the following form of the linear programs **P1, P2** si **P3**:

$$\min\langle c, y\rangle \ \text{ s.t. } \ By \geq d, \ \ y \geq 0 \tag{6.11}$$

and its dual

$$\max\langle d, u\rangle \ \text{ s.t. } \ B^T u \leq c, \ \ u \geq 0 \tag{6.12}$$

where $B : m \times n, c, y \in I\!R^n, d, u \in I\!R^m$. We will denote by $\mathcal{P}, \mathcal{D}$ the set of feasible solutions of the primal (6.11) and its dual. The following result, mentioned (without proof) by Han in his original paper [12], gives us the possibility to express in an equivalent way the primal-dual pair of linear programs. We included its proof in [6].

Proposition 8. *Let us suppose that both problems have feasible solutions, i.e.* $\mathcal{P} \neq \emptyset, \mathcal{D} \neq \emptyset$*.*

Then the following assumptions are equivalent:

(i) $\hat{y} \in \mathcal{P}, \hat{u} \in \mathcal{D}$ are optimal solutions for problems (6.11) and (6.12), respectively; (ii) the vector $x = [\hat{y}^T, \hat{u}^T]^T \in I\!R^{m \times n}$ is a solution of the system of linear inequalities

$$Ax \; \leq \; b, \tag{6.13}$$

where

$$A = \begin{bmatrix} c^T & -d^T \\ -B & 0 \\ 0 & B^T \\ -I & 0 \\ 0 & -I \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ -d \\ c \\ 0 \\ 0 \end{bmatrix} \tag{6.14}$$

It holds that solving a pair of *feasible* primal-dual linear programs is equivalent with solving a certain system of linear inequalities. There are two more possible cases that can occur beside the feasible one: one of the problem has feasible solutions and the other does not, and both problems do not have feasible solutions. In these cases, as Han himself mentioned in [12], the system (6.13) - (6.14) provides a *kind of least squares solution* for one or both linear programs, respectively. Such a situation is considered for the inconsistent problems **P1, P2** and **P3** described above, the systems of linear inequalities to which these problems are equivalent being inconsistent too. The problems were solved with $linprog$ Matlab implementation of Simplex algorithm, whereas the two associated systems with **MH** algorithm. The problems **P1, P2** and **P3** being inconsistent, the Simplex algorithm failed to solve them, returning instead a result that minimizes *the worst case constraint violation* (see [26]). The tables from Fig. 6.2 show the cost solution of the problems considered.

Algoritm	cost	$\Vert (Ax-b)_+ \Vert$
MH	15336	38.7529
Simplex	31235*	145.0241

Algoritm	cost	$\Vert (\hat{A}x-\hat{b})_+ \Vert$
MH	15336	38.7529
Simplex	29070*	145.0626

Algoritm	cost	$\Vert (\hat{A}x-\hat{b})_+ \Vert$
MH	15336	38.7529
Simplex	31235*	146.0675

Figure 6.2: The transportation cost of problem **P1** (left), **P2** (center), **P3** (right)

where * denotes that the Simplex algorithm failed to solve the problem, returning instead a result that minimizes *the worst case constraint violation* (see [26]).

The study presented in this paper will be continued in the future, especially because systems of linear inequalities often appear in applications that target, for example, reconstruction of the CT images, signal processing, etc. Because these problems are usually large, one may consider to parallelize different stages of the algorithms, for example by applying parallel gradient solver in **Step 1**. Furthermore, we will consider to implement these algorithms in a Grid environment, to use all the computing power offered by this platform. The author of this paper has made certain steps in this direction through the works [22], [23] and [24] which describe the modelling of some algorithms applied on satellite images as Grid services. Since the Grid environment enables satellite image processing on its true size, which is usually hundreds of MB, the results are more accurate that those obtained in classical ways, when the image is truncated.

Bibliography

[1] K. Bennett, O.L. Mangasarian, *Robust linear programming discrimination of two linearly inseparable sets*, Optimization Methods and Software, pp. 23-34, 1992

[2] R.A. Bosch, J.A. Smith, *Separating Hyperplanes and the Authorship of the Disputed Federalist Papers*, American Mathematical Monthly, Volume 105, Number 7, 601-608, 1998

[3] S. P. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004

[4] R. Bramley, B. Winnicka, *Solving linear inequalities in a least squares sense*, SIAM J. Sci. Comp., 17(1), pp. 275-286, 1996

[5] R. Bramley, B. Winnicka, *Solving linear inequalities in a least squares sense*, Technical Report TR 396, School of Informatics and Computing, Indiana University - Bloomington, USA, 1994

[6] D. Carp, C.Popa, C. Serban, *Modifed Han algorithm for maritime containers transportation problem*, ROMAI J., v.10, no.1, 11-23, 2014

[7] D. Carp, C. Popa, C. Serban, *Modified Han algorithm for inconsistent linear inequalities*, Carpathian J. Math., 31 (2015), no. 1, 37-44, 2015

[8] Y. Censor, To. Elfving, G.T. Herman, *Regularized Least-Squares Solution of Linear Inequalities*, Technical Report no. MIPG97, University of Pennsylvania, 1985

[9] Y. Censor, A.Z. Stavros, *Parallel optimization: theory, algorithms and applications*, Numer. Math. and Sci. Comp. Series, Oxford Univ. Press, New York, 1997

[10] G. Cimmino, *Calcolo approssiomatto per le soluzioni dei sistemi di equazioni lineari*, Ric.Sci. Prog. Tech. Econom. Naz., pp. 326-333, 1938

[11] C. Golub, V. Pereysa, *Differentiation of pseudoinverse, separable nonlinear least squares problems and other tales*, Generalized Inverses and Applications, Academic Press, New York, 303-323, 1976

[12] S. -P. Han, *Least squares solution of linear inequalities*, Tech. Rep. TR-2141, Mathematics Research Center, University of Wisconsin - Madison, 1980

[13] C. Hildreth, *A quadratic programming procedure*, Naval Res. Logist. Quart. 4, pp.79-85, 1957

[14] S. Kaczmarz, *Approximate solution of systems of linear equations*, Int. J. of Control. 57(6), pp. 1269-1271, 1993

[15] T.C. Koopmans, M. Beckmann, *Assignment problems and location of economic activities*, Econometrica, Vol.25, No.1, pp. 53-76, 1957

[16] A. Lent, Y. Censor, *Extensions of Hildreth's Row-Action Method For Quadratic Programming*", SIAM Control and Optimization, Vol. 8, No.4, 1980

[17] C. Popa, *Extensions of block-projections methods with relaxation parameters to inconsistent and rank-defficient least-squares problems,B I T*, 38(1), pp. 151-176, 1998

[18] C. Popa, *Projection algorithms - classical results and developments. Applications to image reconstruction*, Lambert Academic Publishing - AV Akademikerverlag GmbH & Co. KG, Saarbrücken, Germany, 2012

[19] C. Popa, C. Serban, *Han-type algorithms for inconsistent systems of linear inequalities - a unified approach*, Applied Mathematics and Computation, Volume 246, DOI: 10.1016/j.amc.2014.08.018, 247-256, 2014

[20] C. Popa, D. Carp, C.Serban, *Iterative solution of inconsistent systems of linear inequalities*, Proceedings of 84th Annual Meeting of the International Association of Applied Mathematics and Mechanics (GAMM) March 18-22, Proceedings of Applied Mathematics and Mechanics (PAMM), 13, 407 - 408 (2013) / DOI 10.1002/pamm.201310199, 2013

[21] C. Popa, D. Carp, C. Serban, *Cost Optimization of a 3-index Maritime Container Transportation Problem using the Modified Han Algorithm*, Analele Universitatii Ovidius din Constanta - Seria Stiinte Economice - Vol.XIV, nr. 1/2014, http://stec.univ-ovidius.ro/html/anale/RO, 2014

[22] C. (Gherghina) Serban, C. Maftei, *Computational Grids and Remote Sensing-based Analysis of Thermal Environment of a Geographic Area*, Proceedings of the 9th RoEduNet IEEE International Conference 2010, iunie Sibiu Romania, ISSN 2068-1046, pp. 346-352, 2010

[23] C. (Gherghina) Serban, *A Web Services Composition with BPEL and Satellite Image Processing for urban and river bed changes assessment*, Annals of "Ovidius" University of Constanta, Series Mathematics Vol. 17(3), ISSN 1224-1784, e-ISSN 1844-0835, 131-148, 2009

[24] C. (Gherghina) Serban, E. Isbasoiu, *Grid Services and Satellite Image Processing for urban and river bed changes assessment*, Proceedings of the 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing - SYNASC 2008, IEEE Computer Press, ISBN 978-0-7695-3523-4, 400-404, 2008

[25] J.H. Spoonamore, *Least square methods for solving systems of inequalities with application to an assignment problem*, Doctoral Dissertation, University of Illinois at Urbana-Champaign Champaign, USA,1992

[26] R.J. Vanderbei, *Linear programming. Foundations and extensions*, Int. Series in Oper. Res. and Manag. Sciences, vol. 37, Springer US, 2001

[27] K. Yang, *New iterative methods for linear inequalities*, Techical Report 90-6, Department of Industrial and Operations Engineering, University of Michigan, USA, 1990

[28] University of California-Irvine Repository of Machine Learning Databases and Domain Theories, ftp://ftp.ics.uci.edu/pub/machine-learning-databases/

[29] MATLAB (Matrix Laboratory), http://www.mathworks.com/products/matlab/