A Probabilistic Algorithm for the Solution of Homogeneous Linear Inequalities

by

Mina Asgari Moslehabadi

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Mathematics and Statistics
in Partial Fulfillment for the Requirements of
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2010

A PROBABILISTIC ALGORITHM FOR

THE SOLUTION OF HOMOGENEOUS LINEAR INEQUALITIES

by

Mina Asgari Moslehabadi

APPROVED BY:

Y. P. Aneja
Department of Odette School of Business

T. Traynor
Department of Mathematics and Statistics

R. J. Caron, Supervisor
Department of Mathematics and Statistics

A. Alfakih, Chair of Defense
Department of Mathematics and Statistics

May, 2010

Author's Declaration of Originality

In hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referring practices.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

This thesis presents a probabilistic algorithm for the solution of system of homogeneous linear inequality constraints. In fact, the proposed method simultaneously provides information required for constraint analysis and, if the feasible region is not empty, with probability one, will find a feasible solution. In [**1**] Caron and Traynor explored the relationship between the constraint analysis problem and a certain set covering problem proposed by Boneh [**2**]. They provided the framework that showed the connection between minimal representations, irreducible infeasible systems, minimal infeasibility sets, as well as other attributes of preprocessing of mathematical programs. In [**3**] 2010 Caron et. al. showed the application of the constraint analysis methodology to linear matrix inequality constraints. This thesis builds on those results to develop a method specific to a system of homogeneous linear inequalities. Much of this thesis is devoted to the development of a hit and run sampling methodology.

Dedication

To my parents

and my husband Kevin

whose love and support

have made everything I do possible.

Acknowledgements

I would like to thank my advisor Dr. Richard J. Caron who helped me in all stages of this research. I also appreciate the contribution of Dr. Tim Traynor in this Project. I offer my thanks to Dr. Yash P. Aneja as a member of my committee. I also would express my gratitude to Mr. Alireza Nikookar for his guidance in my project. Finally, I wish to express my utmost gratitude to my husband Dr. Kevin Abbasi for his support.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1. Statement of Thesis and Outline

We are given an $(m \times n)$ real matrix A and n-vector c. Let $\mathbf{0}$ be the zero vector of appropriate dimension. We are concerned with the two regions

$$\mathcal{R}_1(A) = \{\, x \in \mathbb{R}^n \,|\, Ax \leq \mathbf{0}\} \quad \text{and} \tag{1.1.1}$$

$$\mathcal{R}_2(A, c) = \{\, x \in \mathbb{R}^n \,|\, c^\top x < 0,\ Ax \leq \mathbf{0}\} \tag{1.1.2}$$

each defined by homogeneous linear inequalities. If we take

$$A^\top = [a_1, a_2, \ldots, a_m]$$

we have

$$\mathcal{R}_1(A) = \{\, x \in \mathbb{R}^n \,|\, a_i^\top x \leq 0,\ i \in I\} \quad \text{and} \tag{1.1.3}$$

$$\mathcal{R}_2(A, c) = \{\, x \in \mathbb{R}^n \,|\, c^\top x < 0,\ \ a_i^\top x \leq 0,\ i \in I\} \tag{1.1.4}$$

where $I = \{1, 2, \ldots, m\}$. For each region, we will present a probabilistic algorithm to either determine that it is empty or provide a solution. Our thesis is that our probabilistic method will be a tool for the solutions of these systems.

In the remainder of the first chapter, we will explain the importance of this problem in optimization, beginning with its obvious connection to Farkas' theorem of the alternative. Chapter 2 will begin with a description of the probabilistic hit-and-run algorithm [**4, 5, 6**], followed by a description of the constraint analysis methodology given by Caron and Traynor [**1**], based on the work of Boneh [**2**]. In chapter 3, we present our specialization of the Caron-Traynor method to (1.1.3-1.1.4) and provide numerical evidence of its effectiveness. Conclusions are in chapter 4.

1.2. Farkas' Theorem of the Alternative

We will state Farkas' theorem of the alternative [**7**] and show its connection to our thesis, and then give a proof. First, we provide the definitions and results that will be used in the proof.

DEFINITION 1.2.1. *The cone spanned by the columns of $A^\top$ is the set of all non-negative linear combination of the columns of $A^\top$ and is given by*

$$K(A^\top) = \{\, x \in \mathbb{R}^n \,|\, x = A^\top \alpha,\ \alpha \geq 0\}.$$

DEFINITION 1.2.2. *A set $H \subseteq \mathbb{R}^n$ is convex if and only if for every x_1 and x_2 in H, and $\lambda \in [0,1]$, $\lambda x_1 + (1-\lambda)x_2 \in H$.*

DEFINITION 1.2.3. *The point x is a convex combination of $x_1, \ldots, x_k$ if $x = \sum_{i=1}^{k} \lambda_i x_i$, $\lambda_i \geq 0$ and $\sum_{i=1}^{k} \lambda_i = 1$.*

DEFINITION 1.2.4. *A set $H \subseteq \mathbb{R}^n$ is closed if for any arbitrary sequence $\{x_j\}$ in H with $\{x_j\} \to \bar{x}$, $\bar{x}$ is in H.*

DEFINITION 1.2.5. *A set $H \subseteq \mathbb{R}^n$ defined by $H = \{x \in \mathbb{R}^n \,|\, h^\top x = \alpha\}$, where $h \in \mathbb{R}^n$ with $h \neq \mathbf{0}$ and $\alpha \in \mathbb{R}$, is a hyperplane.*

DEFINITION 1.2.6. *A point x_1 of a set H is a boundary point of H if every neighborhood of the point x_1 contains points both in H and in its complement.*

THEOREM 1.2.7. *Suppose that $D \subseteq \mathbb{R}^n$ is a nonempty closed, convex set. If $\hat{x} \notin D$ then there is a separating hyperplane $h^\top x = \alpha$ such that $h^\top y < \alpha < h^\top \hat{x}$ for every $y \in D$* [**8**].

Proof. Let $f(y) = \|\hat{x} - y\|$. We choose a point y^* such that

$$f(y^*) = \min_{y \in D} f(y).$$

To see that this minimizer exists, let y_0 be any point of D, and let $r = f(y_0)$. Then $D \bigcap B(\hat{x}, r)$ is a closed bounded set. Since f is a continuous function, it has a minimum value on $D \bigcap B(\hat{x}, r)$ at some y^*. For $y \notin B(\hat{x}, r)$ we have $f(y) \geq f(y_0) \geq f(y^*)$. So y^* is the required minimizer. We define $h = \hat{x} - y^*$. Since $h \neq \mathbf{0}$ then $h^\top(\hat{x} - y^*) > 0$ then $h^\top \hat{x} > h^\top y^*$. For any point $y \in D$, let $\lambda \in (0, 1)$, then by convexity of D we have $\lambda y + (1 - \lambda)y^* \in D$. Since y^* is the minimum point of $f(y)$ we have

$$\begin{aligned}
(\hat{x} - y^*)^\top(\hat{x} - y^*) &\leq [(\hat{x} - \lambda y - (1 - \lambda)y^*)]^\top[(\hat{x} - \lambda y - (1 - \lambda)y^*)] \\
&= [\hat{x} - y^* + \lambda(y^* - y)]^\top[\hat{x} - y^* + \lambda(y^* - y)] \\
&= (\hat{x} - y^*)^\top(\hat{x} - y^*) + 2\lambda(\hat{x} - y^*)^\top(y^* - y) + (\lambda)^2(y^* - y)^\top(y^* - y).
\end{aligned}$$

If we rearrange terms in above inequality, we will get the following inequality

$$0 \leq 2\lambda(\hat{x} - y^*)^\top(y^* - y) + (\lambda)^2(y^* - y)^\top(y^* - y)$$

by dividing both sides with 2λ and taking the limit as $\lambda \to 0$, we have $(\hat{x} - y^*)^\top(y^* - y) \geq 0$, then $h^\top y^* \geq h^\top y$. If we choose any α between $h^\top \hat{x}$ and $h^\top y^*$ we have $h^\top y < \alpha < h^\top \hat{x}$ for every $y \in D$[**9**].

□

Theorem 1.2.8. *(Caratheodory's theorem for cones* [**10**] *). Any nonzero $x \in K(A^\top)$ can be written as a nonnegative linear combination of linearly independent columns of $A^\top$.*

Proof. Suppose that x is a nonzero vector in $K(A^\top)$ and also suppose that t is the smallest cardinality of the subsets of columns of $A^\top$. Then $x = \Sigma_{i=1}^{i=t} \alpha_i a_i$ where $\alpha_i \geq 0$. Then it remains to show that all of $\{a_1, \ldots, a_t\}$ are linearly independent. Suppose otherwise. Then there exists μ_i, not all of them zero, such that $\Sigma_{i=1}^{i=t} \mu_i a_i = 0$.

Now consider the following linear combination

$$\Sigma_{i=1}^{i=t}(\alpha_i - \gamma\mu_i)a_i \tag{1.2.1}$$

where

$$\gamma = \min\{\alpha_i/\mu_i, \mu_i > 0\}.$$

Hence, $\alpha_i - \gamma\mu_i \geq 0$. If k is the value of i such that $\gamma = \alpha_i/\mu_i$, then $\alpha_k - \gamma\mu_k = 0$, so that we can represent x as linear combination of fewer than t columns in $K(A^\top)$, a contradiction. $\square$

THEOREM 1.2.9. *$K(A^\top)$ is a closed, convex set.*

PROOF. This proof can be found in [**11**]. First we suppose that the columns of $A^\top$ are linearly independent. Now let $\{x_j\}$ be any arbitrary sequence in $K(A^\top)$ with $\{x_j\} \to \bar{x}$. We will show that $\bar{x} \in K(A^\top)$. For each j we have $x_j = A^\top\alpha_j$ for some $\alpha_j \geq 0$, from which we obtain $Ax_j = A(A^\top\alpha_j) = (AA^\top)\alpha_j$. Since the columns of $A^\top$ are linearly independent, then $AA^\top$ is invertible and $\alpha_j = (AA^\top)^{-1}Ax_j$. Since $\{x_j\} \to \bar{x}$, $\alpha_j \to \bar{\alpha} = (AA^\top)^{-1}A\bar{x}$. Since $\alpha_j \geq 0$ then we have $\bar{\alpha} \geq 0$, so $\bar{x} \in K(A^\top)$. Now, we drop the assumption of linear independence columns of matrix $A^\top$. Let $P(I)$ be the power set of I. Let $A_J^\top$, $J \in P(I)$ be the matrix whose columns are the columns of $A^\top$ indexed by J. Let $\hat{P}(I) \subseteq P(I)$ be such that $A_J^\top$ has full rank if and only if $J \in \hat{P}(I)$. By Caratheodory's theorem 1.2.8

$$K(A^\top) = \bigcup_{J \in \hat{P}(I)} K(A_J^\top)$$

since each $K(A_J^\top)$, $J \in \hat{P}(I)$ is closed so $K(A^\top)$ is closed.

For convexity, suppose that $x_1 \in K(A^\top)$, $x_2 \in K(A^\top)$, $\lambda \in [0,1]$, define $\alpha_1 \geq 0$, define $\alpha_2 \geq 0$ such that $x_1 = A^\top\alpha_1$ and $x_2 = A^\top\alpha_2$. Then we have $\lambda x_1 + (1-\lambda)x_2 = \lambda A^\top\alpha_1 + (1-\lambda)A^\top\alpha_2 = A^\top(\lambda\alpha_1 + (1-\lambda)\alpha_2) \in K(A^\top)$. $\square$

And now, Farkas' theorem.

THEOREM 1.2.10. *Either there is a solution to $\mathcal{R}_2(A,c)$ or a solution to $\mathcal{S}_2(A,c)$ but never both, where*

$$\mathcal{S}_2(A,c) = \{\, y \in \mathbb{R}^m \mid A^\top y = -c,\ y \geq \mathbf{0}\}.$$

PROOF. We first show that we can not have a solution to both $\mathcal{R}_2(A,c)$ and $\mathcal{S}_2(A,c)$. Suppose otherwise. Let $\hat{x} \in \mathcal{R}_2(A,c)$ and $\hat{y} \in \mathcal{S}_2(A,c)$. Post-multiplication of $(A^\top \hat{y})^\top = -c^\top$ by $(-\hat{x})$ gives $(-\hat{y}^\top A\hat{x}) = c^\top \hat{x}$. Now, $c^\top \hat{x} < 0$; and, since $\hat{y} \geq \mathbf{0}$ and $A\hat{x} \leq \mathbf{0}$ we have $(-\hat{y}^\top A\hat{x}) \geq 0$. A contradiction.

Now suppose that $\mathcal{S}_2(A,c) = \emptyset$ so that $-c \notin K(A^\top)$. We will show that $\mathcal{R}_2(A,c) \neq \emptyset$. Since $K(A^\top)$ is closed and convex, theorem 1.2.7 guarantees a separating hyperplane $h^\top x = \alpha$ with $h^\top y < \alpha < h^\top(-c)$ for every $y \in K(A^\top)$. Since $\mathbf{0} \in K(A^\top)$ we have $0 < \alpha < -h^\top c$ so that $c^\top h < 0$. It remains to show that $Ah \leq \mathbf{0}$, that is, $h^\top a_i \leq 0$ for all $1 \leq i \leq m$. Suppose $a_k^\top h > 0$ for some index k. By definition, $\lambda a_k \in K(A^\top)$ for all $\lambda \geq 0$. Thus, for all $\lambda \geq 0$, $\lambda h^\top a_k < \alpha$. But, this only holds for $\lambda < \alpha/(h^\top a_k)$, a contradiction. Thus, $Ah \leq \mathbf{0}$ and $h \in \mathcal{R}_2(A,c)$. □

The connection of Farkas' theorem to our thesis is clear in that it involves $\mathcal{R}_2(A,c)$, a system for which we will propose a probabilistic algorithm. The importance of our thesis is, therefore, connected to the importance of Farkas' theorem. We provide four specific examples.

1.2.1 Linear Programming.

Consider the linear programming problem (LP)

$$\begin{aligned} &\text{minimize} && c^\top x \\ &\text{subject to} && Ax \leq b \end{aligned} \tag{1.2.2}$$

where A and c are defined in 1.1.2 and b is m-vector. We denote the feasible region of the LP by

$$\mathcal{R} = \{x \in \mathbb{R}^n \mid Ax \leq b\}. \tag{1.2.3}$$

We present some definitions and standard results.

DEFINITION 1.2.11. *The point $\hat{x}$ is an extreme point of $\mathcal{R}$, if it is impossible to represent $\hat{x}$ as a proper convex combination of two other distinct points in $\mathcal{R}$.*

DEFINITION 1.2.12. *The point $\hat{x}$ is a degenerate extreme point of $\mathcal{R}$, if it is an extreme point and if the number of constraints such that $a_i^\top \hat{x} = b_i$, is strictly greater than n.*

In the next lemma we show that solutions to $\mathcal{R}_2(A, c)$ are directions of unboundedness for the LP (1.2.2).

LEMMA 1.2.13. *If $\mathcal{R} \neq \emptyset$ and if $\hat{s} \in \mathcal{R}_2(A, c)$ then the LP (1.2.2) is unbounded from below in the direction $\hat{s}$.*

PROOF. Let $\hat{x} \in \mathcal{R}$ and $\hat{s} \in \mathcal{R}_2(A, c)$. We have $A(\hat{x} + \sigma\hat{s}) = A\hat{x} + \sigma A\hat{s} \leq b$ so $\hat{x} + \sigma\hat{s} \in \mathcal{R}$ for all $\sigma \geq 0$. We have $c^\top(\hat{x} + \sigma\hat{s}) = c^\top\hat{x} + \sigma c^\top\hat{s} \to -\infty$ as $\sigma \to +\infty$ since $c^\top\hat{s} < 0$. Thus, the LP is unbounded from below in the direction $\hat{s}$. □

DEFINITION 1.2.14. *The vector $\hat{s}$ is a descent direction for the LP (1.2.2) at $\hat{x}$ if for $\sigma > 0$ we have $c^\top(\hat{x} + \sigma\hat{s}) < c^\top\hat{x}$.*

LEMMA 1.2.15. *Let $\hat{x} \in \mathcal{R}$. We can partition $A^\top = [A_1^\top, A_2^\top]$ and $b^\top = [b_1^\top, b_2^\top]$ so that $A_1\hat{x} = b_1$ and $A_2\hat{x} < b_2$. If there is an $\hat{s} \in \mathcal{R}_2(A_1, c)$, then $\hat{s} \neq \mathbf{0}$ is a direction of descent for the LP (1.2.2). Otherwise $\hat{x}$ is an optimal solution for the LP (1.2.2).*

PROOF. Suppose that $\hat{s} \in \mathcal{R}_2(A_1, c)$. Since $c^\top\hat{s} < 0$, then $c^\top(\hat{x} + \sigma\hat{s}) < c^\top\hat{x}$ for all $\sigma > 0$. And by definition 1.2.14 $\hat{s} \neq \mathbf{0}$ is a descent direction for the LP (1.2.2). For second part if $\mathcal{R}_2(A_1, c) = \emptyset$ we will show that $c^\top(\hat{x} - x) \leq 0$ for all $x \in \mathcal{R}$. Since $\mathcal{R}_2(A_1, c) = \emptyset$, then from Farkas' theorem 1.2.10 there is a $y \in \mathcal{S}_2(A_1, c)$ and we have

$$\begin{aligned}
c^\top(\hat{x} - x) &= -y^\top A_1(\hat{x} - x) && (y \in \mathcal{S}_2(A_1, c)) \\
&= -y^\top A_1\hat{x} + y^\top A_1 x && \\
&= -y^\top b_1 + y^\top A_1 x && (A_1\hat{x} = b_1) \\
&= y^\top(A_1 x - b_1) && \\
&\leq 0. && (x \in \mathcal{R} \text{ and } y \in \mathcal{S}_2(A_1, c))
\end{aligned}$$

$\square$

If, in lemma 1.2.15, $\hat{x}$ is a degenerate extreme point, then an algorithm for the determination of a solution to $\mathcal{R}_2(A, c)$ provides an alternative to standard anti-cycling rules [**12**]. In fact, if there is no descent direction at the extreme point $\hat{x}$ then $\hat{x}$ will be an optimal solution for the LP (1.2.2).

Lemmas 1.2.13 and 1.2.15 lead to the following, well known, set of necessary and sufficient conditions for linear programming.

THEOREM 1.2.16. *The point $\hat{x}$ is an optimal solution for LP (1.2.2) if and only if together with some $\hat{y}$ it satisfies the Kurush Kuhn-Tucker conditions*

(1) *Primal Feasibility:* $Ax \leq b$

(2) *Dual Feasibility:* $A^\top y = -c$, $y \geq \mathbf{0}$

(3) *Complementary Slackness:* $y^\top (Ax - b) = 0$

PROOF. Suppose that $\hat{x}$ and $\hat{y}$ satisfy the Kurush Kuhn-Tucker conditions and consider any $x \in \mathcal{R}$. Since $\hat{x} \in \mathcal{R}$ from the primal feasibility condition, we need only show that $c^\top (\hat{x} - x) \leq 0$ for all $x \in \mathcal{R}$. We have

$$
\begin{aligned}
c^\top (\hat{x} - x) &= -\hat{y}^\top A(\hat{x} - x) && \text{(from dual feasibility)} \\
&= -\hat{y}^\top A\hat{x} + \hat{y}^\top Ax \\
&= -\hat{y}^\top b + \hat{y}^\top Ax && \text{(from complementary Slackness)} \\
&= \hat{y}^\top (Ax - b) && (x \in \mathcal{R} \text{ and dual feasibility}) \\
&\leq 0.
\end{aligned}
$$

For the converse, assume that $\hat{x}$ is optimal for LP (1.2.2). Thus, $\hat{x} \in \mathcal{R}$ and primal feasibility is satisfied. Define A_1, A_2, b_1 and b_2 as in lemma 1.2.15, then by optimality of $\hat{x}$, $\mathcal{R}_2(A_1, c) = \emptyset$. Thus, from Farkas' theorem 1.2.10 there is a $\hat{y}_1 \in \mathcal{S}_2(A_1, c)$. We define $\hat{y}_2 = \mathbf{0}$ and set $\hat{y}^\top = [\hat{y}_1^\top, \hat{y}_2^\top]$ so that $\hat{y}_1^\top (A_1\hat{x} - b_1) = 0$ and $\hat{y}_2^\top (A_2\hat{x} - b_2) = 0$ (complementary slackness); and $-c = A_1^\top \hat{y}_1 + A_2^\top \hat{y}_2$ and $\hat{y} \geq \mathbf{0}$ (dual feasibility). Thus, $\hat{x}$ and $\hat{y}$ satisfy the necessary and sufficient conditions. $\square$

1.2.2 Redundancy and Degeneracy.

The importance of the detection and removal of redundancy in linear programming has been well established, for example, consider [**13, 14**]. We consider set of constraints

$$C(I) = \{a_i^\top x \leq b_i \,|\, i \in I\}$$

DEFINITION 1.2.17. *The k-th constraint is redundant in $C(I)$, if $\mathcal{R} = \mathcal{R}_k$, where $\mathcal{R}_k = \{\, x \in \mathbb{R}^n \,|\, a_i^\top x \leq b_i,\ i \in I \setminus \{k\}\}$. In other words, a redundant constraint is one that can be removed without changing the feasible region, which might not be empty.*

DEFINITION 1.2.18. *The k-th constraint is necessary in $C(I)$, if $\mathcal{R} \neq \mathcal{R}_k$. In other words, a necessary constraint is the one where its removal will change the feasible region, possibly from empty to nonempty.*

Suppose that $\hat{x}$ is a degenerate extreme point. Define the index set $A(\hat{x}) = \{\, i \in I \mid a_i^\top \hat{x} = b_i \,\}$ and let $\hat{A}$ be the matrix whose rows are the $a_i^\top$, $i \in A(\hat{x})$.

THEOREM 1.2.19. *The k-th constraint, where $k \in A(\hat{x})$, is redundant in $C(A(\hat{x}))$ if and only if $\mathcal{R}_2(\hat{A}, -a_k) = \emptyset$.*

PROOF. See theorem 4.1 in Caron et. al. [**15**] □

Thus, the algorithm to be proposed to solve homogeneous linear systems can be applied to the problem of linear programming redundancy.

1.2.3 Unboundedness of Quadratically Constrained Quadratic Programming (QCQP).

DEFINITION 1.2.20. *The symmetric real matrix B of order $(n \times n)$ is positive semidefinite if $x^\top Bx \geq 0$ for all $x \in \mathbb{R}^n$. The matrix B is positive definite if $x^\top Bx > 0$ for all nonzero $x \in \mathbb{R}^n$.*

Consider the quadratically constrained quadratic programming (QCQP)

$$\begin{aligned} &\text{minimize} && Q_0(x) = a_0^\top x + \tfrac{1}{2}x^\top B_0 x \\ &\text{subject to} && Q_i(x) = a_i^\top x + \tfrac{1}{2}x^\top B_i x \le b_i,\ i \in I \end{aligned} \tag{1.2.4}$$

where B_0 and all B_i are positive semidefinite matrices of order $(n \times n)$. Also a_0 and all a_i are n vectors, and all b_i are scalars. We denote the feasible region of the (QCQP) by

$$\mathcal{R}_Q = \{x \in \mathbb{R}^n \mid Q_i(x) \le b_i,\ i \in I\}. \tag{1.2.5}$$

DEFINITION 1.2.21. *The QCQP (1.2.22) is unbounded from below if for any $M \in \mathbb{R}$ there exists an $\hat{x} \in \mathcal{R}_Q$ with $Q_0(\hat{x}) < M$.*

Let's define $I_0 = \{0, 1, \ldots, m\}$. Consider the following theorem from [**16**] or [**17**].

THEOREM 1.2.22. *The function $Q_0(x)$ is unbounded from below along a half line in $\mathcal{R}_Q$ if and only if there exists a vector x satisfying the following conditions*

(1) $a_0^\top x < 0$

(2) $B_i x = \mathbf{0}$, $\forall i \in I_0$

(3) $Ax \le 0$

PROOF. For proof see [**17**]. □

DEFINITION 1.2.23. *The null space of the matrix B is a subspace of $\mathbb{R}^n$ and is given by $N(B) = \{x \in \mathbb{R}^n \mid Bx = 0\}$.*

DEFINITION 1.2.24. *A real-valued function f defined on any convex subset of some vector space is convex if and only if for any two points y, z in its domain and any λ in $[0, 1]$ we have:*

$$f(\lambda y + (1-\lambda)z) \le \lambda f(y) + (1-\lambda) f(z).$$

LEMMA 1.2.25. *The quadratic function $Q_0(x)$ is convex if and only if B_0 is positive semidefinite.*

Proof. For proof see lemma 1.2.5 in [**18**]. $\square$

We let $N = \bigcap_{i \in I_0} N(B_i)$. Its members are the points x satisfying condition (2) of 1.2.22. If $N = \{0\}$ then $Q_0(x)$ is bounded below. Otherwise, let V be a matrix whose column space is N and observe that all three conditions are satisfied if there exists y with $AVy \leq 0$ and $a_0^\top Vy < 0$, that is $\mathcal{R}(AV, a_0^\top V) \neq \emptyset$. Thus, if we stay in a null space of matrices then the algorithm to be proposed can be applied to determine whether or not a convex quadratic objective function is bounded from below over a feasible region defined by convex quadratic constraints.

1.2.4 Murty's Proposed Interior Point Method for Quadratic Programming (QP).

Consider the following quadratic programming (QP)

$$\begin{aligned} &\text{minimize} \quad && Q(x) = a_0^\top x + \tfrac{1}{2} x^\top B_0 x \\ &\text{subject to} \quad && Ax \leq b \end{aligned} \tag{1.2.6}$$

where B_0 is positive definite of order n and vector a_0 is defined in the QCQP (1.2.22). Suppose that $\|a_i\| = 1$. We define

$$\delta(x) = \min\{b_i - a_i^\top x : \, i \in I\}$$

so that $\delta(x)$ is the radius of the largest ball inside $\mathcal{R}$ centered at x. If $B(x, \delta)$ is the ball centred at x with radius δ then

$$T(x) = \{i \,|\, b_i - a_i^\top x = \delta(x)\}$$

is the index set of touching constraints for this ball. Murty in [**19**] presented an interior point method for the quadratic programming (1.2.6). The stated advantage of Murty's algorithm is that it does not need the inversion of large matrices as do other interior point methods. The algorithm is started with an initial interior feasible point x_0. Each iteration of the algorithm consists of two steps: a centering step and a descent step. Since the application of our proposed algorithm is in a centering step, we just show this part of Murty's algorithm. Suppose that the point x_0 is the current interior feasible solution for the QP (1.2.6). The purpose of the centering step is to find a new interior point $x \in \mathcal{R}$ to get the largest ball, $B(x, \delta)$, inside $\mathcal{R}$ where the objective value $Q(x)$ is less than or equal to $Q(x_0)$. For this reason, we need to solve the following max-min problem

$$\begin{aligned} \text{maximize} \quad & \delta \\ \text{subject to} \quad & \delta \leq b_i - a_i^\top x, \qquad \forall i \in I, \\ & Q(x) \leq Q(x_0). \end{aligned} \tag{1.2.7}$$

Let (x_r, δ_r) be the current optimal solution to (1.2.7), where $\delta_r = \delta(x_r)$. We want to find a search direction p_r and a non-negative step size α_r such that $\nabla Q(x_r)^\top p_r < 0$ and $\delta(x_r + \alpha_r p_r) \geq \delta(x_r)$. Murty called such p_r a *profitable direction.* Murty [**19**] stated that a direction p_r satisfies $\delta(x_r + \alpha p_r) > \delta(x_r)$ for positive values of α if and only if all entries in $\{a_i^\top p_r : i \in T(x_r)\}$ are of the same sign. This provides that to solve the following system to find a profitable direction p_r

$$\begin{aligned} & \nabla Q(x_r)^\top p_r < 0 \\ & a_i^\top p_r < 0, \quad \forall i \in T(x_r), \end{aligned} \tag{1.2.8}$$

Murty suggested that we only consider solutions p_r to 1.2.8 are in the set $\Gamma = \{-a_i, a_i : i \in I\}$. But in [**18**] Vasilyeva showed that Murty's suggestion to find a profitable directions does not always work.

Our proposed algorithm can be applied to modify the Murty's algorithm to find a profitable direction, i.e. a solution to 1.2.8.

CHAPTER 2

Probabilistic Hit and Run Methods (HR) with Set Covering Problem Approach

2.1. Introduction

We present the probabilistic hit-and-run HR methods with set covering problem approach in this chapter. A literature review is presented in section 2.2. There are two well-known probabilistic hit-and-run methods: a hypersphere direction HD method and a coordinate direction CD method, those are described in detail in section 2.3. We present a related set covering problem in section 2.4. Finally a method for the construction of the set covering matrix is explained in section 2.5.

2.2. Literature Review

The hit-and-run algorithm is a Markov chain sampler for generating points from general continuous distributions over a bounded open region. The basic hit-and-run algorithm is the so-called hypersphere direction HD and it was introduced by Boneh and Golan in 1979 [**4**] as way to remove redundant constraints in optimization problems. The HD algorithm was soon followed by many variants. The first was the coordinate direction CD method. This method was suggested by Telgen [**20**] as an alternative to the HD method.

Independently, in 1980 Smith, [**21**] introduced HD, as a symmetric mixing algorithm, for generating random points over the feasible region of a certain problem. In 1984, Smith [**5**] proved that if the feasible region is open and bounded then the sequence of iteration points of the HD algorithm converges to the uniform distribution over the feasible region.

In 1993, Belisle et. al. [**22**] introduced a general class of hit-and-run algorithms which included HD and CD, and showed that these algorithms, under rather weak conditions generate points asymptotically uniformly in a bounded open set.

2.3. HD Method for Linear System

Suppose that $\mathcal{R}$ is full dimensional, so that, there exists an x_0 with $Ax_0 < b$. Also suppose that $\mathcal{R}$ is bounded.

DEFINITION 2.3.1. *The interior of* $\mathcal{R}$ *is denoted by* $int(\mathcal{R})$ *where* $int(\mathcal{R}) = \{\, x \in \mathcal{R} \mid Ax < b\}$. *Any point* $x \in int(\mathcal{R})$ *is called interior point.*

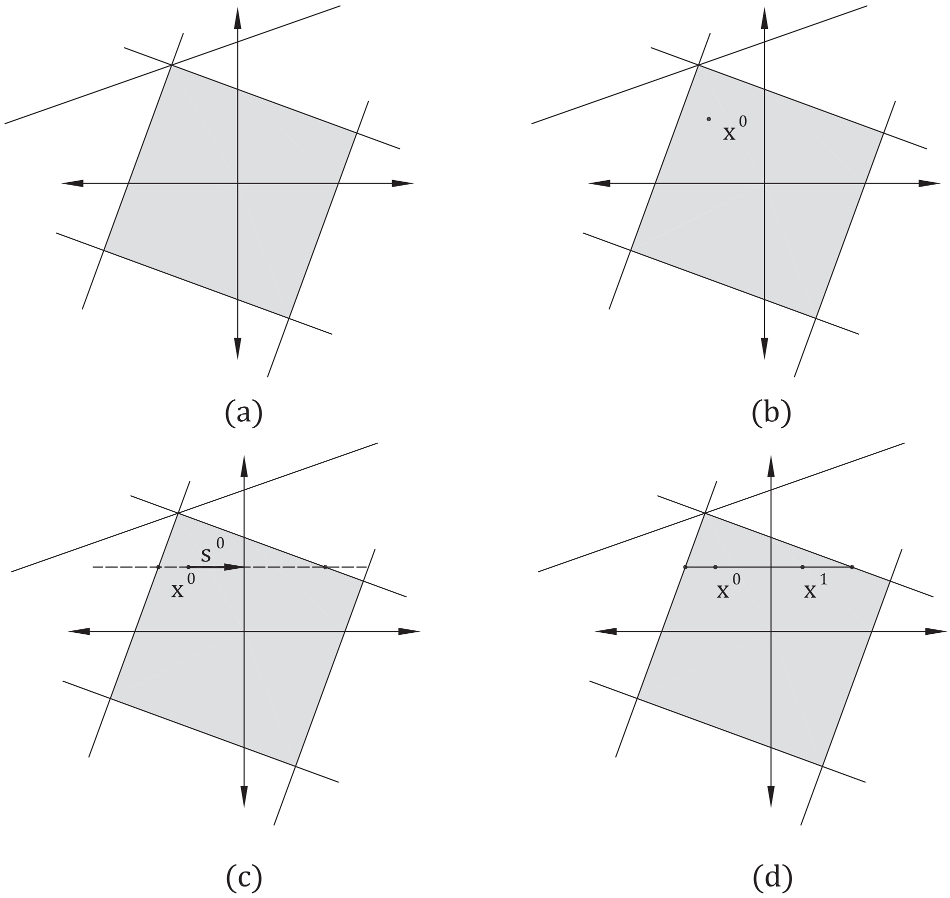
DEFINITION 2.3.2. *The region* $\mathcal{R}$ *is bounded if it is contained in a ball.*

We start an iteration of HD algorithm with a feasible interior point x^k. In the hit step, we generate a direction s^k uniformly over the surface of n-dimensional hyper sphere

$$H = \{\, x \in \mathbb{R}^n \mid \, \|x\| = 1\}.$$

In practice we generate n independent $z_i \sim N(0,1)$, and then we set $s^k = z/\|z\|$, where $z = (z_1, \ldots, z_n)^\top$. The line passing through point x^k in the direction s^k intersects the boundaries of the feasible region $\mathcal{R}$ and defines a feasible line segment in $\mathcal{R}$. Those two constraints that determine the end points of the feasible line segment are listed as a non-redundant constraints. In the run step, we generate a new interior point x^{k+1} uniformly over the feasible line segment. The point x^{k+1} will be the interior point for the next iteration. These steps will be repeated until some termination rule is satisfied, e.g. a Bayesian stopping criterion [**12**]. After termination, all constraints that have not been identified during the algorithm, are listed as redundant constraints, possibly with error.

A CD method is exactly the same with HD method except in choosing the direction. In CD method we select a direction vector s^k uniformly from the n standard coordinate directions in $\mathbb{R}^n$. To do this, we choose $\nu \sim U(0,1)$, let $\kappa = \lceil n\nu \rceil$ then we set $s^k = e_\kappa$ the κ-th coordinate direction. The Figure 2.1 illustrates a single of CD



(a) (b) (c) (d)

FIGURE 2.1. Example of CD hit-and-run iteration .

iteration. Figure 1.6 (a) shows the feasible region in $\mathbb{R}^2$. We choose x^0 as an interior feasible point in Figure 1.6 (b). We next generate a direction s^0 as shown by arrow at x^0 in Figure 1.6 (c) and, the line passing through x^0 in the direction s^0 is drawn as dashed line in Figure 1.6 (c). Finally, in Figure 1.6 (d) we see the feasible line segment and new interior feasible point x^1 and two non-redundant constraints. In the Algorithm 1 we see all detail for the CD method.

THEOREM 2.3.3. *From* [**6**]*, in the CD hit and run algorithm, if*

$$u \triangleq \arg\min\{\sigma_i \,|\, \sigma_i > 0,\ i \in I\}$$

Algorithm 1 The CD hit-and-run algorithm

Given a constraint set $C(I)$ with $x^0 \in int(\mathcal{R})$, $\mathcal{R}$ is bounded

Set $k = 0,\ J = I$

Generate $\nu \sim U(0,1)$, let $\kappa = \lceil n\nu \rceil$ and, set $s^k = e_\kappa$

for $i = 1, \ldots, m$ **do**

  Determine $\sigma_i = \frac{b_i - a_i^\top x^k}{a_i^\top s^k}$, with $a_i^\top s^k \neq 0$

end for

Determine $\sigma_u = \min\{\, \sigma_i \,|\, \sigma_i > 0\}$

Determine $\sigma_r = \max\{\, \sigma_i \,|\, \sigma_i < 0\}$

If σ_r and σ_u are unique, then list constraints r and u as nonredundant

Set $J = J \setminus \{r, u\}$

Let $t \sim U(\sigma_r, \sigma_u)$ and set $x^{k+1} = x^k + ts^k$

If termination rule holds, J is output of redundant constraints and $I \setminus J$ is output of non-redundant constraints

Otherwise, set $k = k + 1$

is unique, then constraint $a_u^\top x \leq b_u$ *is non-redundant. And if*

$$r \triangleq \arg\max\{\sigma_i \,|\, \sigma_i < 0,\ i \in I\}$$

is unique, then constraint $a_r^\top x \leq b_r$ *is non-redundant.*

Proof. We just prove that $a_u^\top x \leq b_u$ is non-redundant. We need to show that for $i \neq u$. there exists a point x'' such that

$$a_u^\top x'' > b_u \tag{2.3.1}$$

$$a_i^\top x'' \leq b_i \tag{2.3.2}$$

suppose that x^k be the interior point of feasible region $\mathcal{R}$. When line $L(x^k, s^k)$ intersects the boundary of i-th constraint of $C(I)$ then the value of σ_i at the intersection point with the i-th constraint is equal to $\sigma_i = \frac{b_i - a_i^\top x^k}{a_i^\top s^k}$. It is obvious $\sigma_i > 0$ for all $a_i^\top s^k > 0$. Thus, if $x' = x^k + \sigma_u s^k$ where $\sigma_u = \frac{b_u - a_u^\top x^k}{a_u^\top s^k}$ then

$a_i^\top x' = a_i^\top x^k + (\frac{b_u - a_u^\top x^k}{a_u^\top s^k}) a_i^\top s^k < a_i^\top x^k + (\frac{b_i - a_i^\top x^k}{a_i^\top s^k}) a_i^\top s^k = b_i$, then $a_i^\top x' < b_i$. Since $a_u^\top x' = b_u$ then there exists $\epsilon > 0$ such that $x'' = x' + \epsilon s^k$ satisfies 2.3.1 and 2.3.2. So far we have shown that at the same time $x'' \notin \mathcal{R}$ while $x'' \in \mathcal{R}_u$ then from definition 1.2.18 $\mathcal{R}_u \neq \mathcal{R}$ and this implies that constraint u is non-redundant. Similarly we can prove that $a_r^\top x \leq b_r$ is non-redundant.

Suppose that constraint u is not unique. There are two cases. First one is that duplicate constraints have been hit and second one is that an intersection of constraint boundaries has been hit. The former possibility is assumed not to happen in this thesis. The latter possibility can only happen with probability 0. $\square$

2.4. The Set Covering Approach

For $i \in I$, we define $X_i = \{ x \in \mathbb{R}^n \,|\, a_i^\top x \leq 0\}$; that is, the i-th constraint is satisfied at x if and only if $x \in X_i$. For (1.1.3) we set $X_0 = \mathbb{R}^n \setminus \{\mathbf{0}\}$ and for (1.1.4) $X_0 = \{x \in \mathbb{R}^n | \ c^\top x < 0 \}$. Define the region

$$Z(I_0) = \bigcap_{i \in I_0} X_i.$$

We refer to X_i as the i-th constraint.

DEFINITION 2.4.1. *Constraint i in $C(I)$ is duplicate if there exists an index $j \in I$ such that $X_i = X_j$.*

The indexed family $\{X_i, X_i^c\}$ partitions $\mathbb{R}^n$, where X_i^c refer to the complement of X_i. If $Z(I_0)$ is not empty then $Z(I_0)$ is feasible otherwise is infeasible. If there exists a subset J of I_0 such that $Z(I_0) = Z(J)$ where

$$Z(J) = \bigcap_{i \in J} X_i,$$

then J is a reduction of I_0 and the family $\{X_i\}_{i \in J}$ is a reduction of the family $\{X_i\}_{i \in I_0}$. Also we say that the family $\{X_i\}_{i \in J}$ is irreducible if there is no suitable subsets J' of J such that $Z(J) = Z(J')$.

In this thesis, we are interested in searching for such subsets J of I_0. Because if $Z(I_0)$ is feasible then searching for J is equivalent to the detection of redundant constraints and find an irreducible feasible system. If $Z(I_0)$ is infeasible then searching for J is equivalent to find an irreducible infeasible system (IIS).

Consider the set of four homogeneous linear inequality constraints given by

$$\begin{array}{cccccc} -x_1 & + & x_2 & < & 0 & (0) \\ x_1 & + & x_2 & \leq & 0 & (1) \\ x_2 & & & \leq & 0 & (2) \\ 1/3x_1 & + & x_2 & \leq & 0 & (3) \end{array}$$

that are graphed in the Figure 2.2, then $I_0 = \{0, 1, 2, 3\}$. The hatched area shows the feasible region $Z(I_0)$. We let $J = \{0, 1\}$, since J is a subset of I_0 and $Z(I_0) = Z(J)$ then J is a reduction of I_0. Also J is an irreducible reduction and constraints 0 and 1 are identified as non-redundant and constraints 3 and 4 are detected as redundant.

Now suppose that $Z(I_0)$ is feasible then

$$Z(I_0)^c = \bigcup_{i \in I_0} X_i^c.$$

Thus $\{X_i^c, i \in I_0\}$ is a cover of $Z(I_0)^c$. Also any reduction J of I_0 that defines $Z(I_0)$ provides the reduction of the cover, that is $\{X_i^c, i \in J\}$ should cover $Z(I_0)^c$ or equivalently

$$\bigcup_{i \in J} X_i^c \supseteq Z(I_0)^c, \tag{2.4.1}$$

THEOREM 2.4.2. *From* [**23**]*, suppose that the subset J of I_0, then J is a reduction of I_0 if and only if inclusion 2.4.1 holds.*

PROOF. We always have $Z(I_0) \subseteq Z(J)$. Thus, J is a reduction of I_0 if and only if $Z(J) \subseteq Z(I_0)$. And $Z(J) \subseteq Z(I_0)$ if and only if for any x such that $x \notin Z(I_0)$, $x \notin Z(J)$. This means when x is infeasible then there exists $i \in J$ such that $a_i^\top x > 0$.

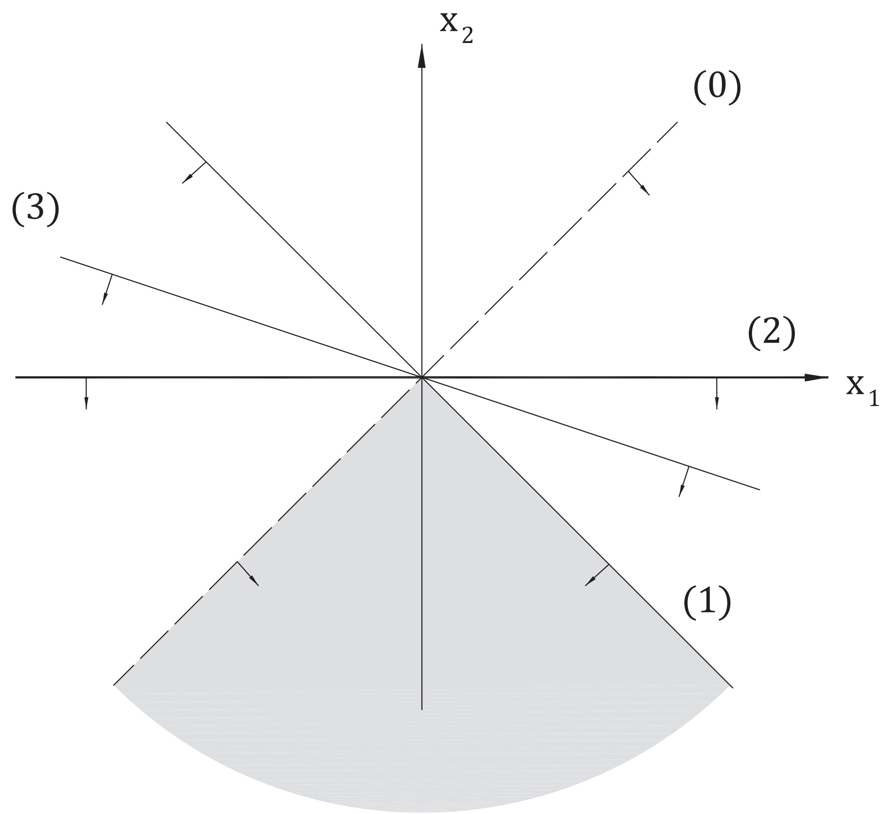


FIGURE 2.2. The example of redundant and non-redundant constraints

Let $X_i^c = \{\, x \in \mathbb{R}^n \,|\, a_i^\top x > 0,\ i \in J\}$. Then J is a reduction of I_0 if and only if the family $\{X_i^c\}_{i \in J}$ covers the set $Z(I_0)^c$ of infeasible points. $\square$

In fact inclusion 2.4.1 shows that each infeasible point in some X_i^c, $i \in J$ violates some constraints in the reduction J. If J is an irreducible reduction then all constraints in J are necessary and all constraints in $I_0 \setminus J$ are redundant.

THEOREM 2.4.3. *From Boneh in* [**2**]*, if x is an infeasible point then at least one constraint violated by x is necessary in each reduction.*

PROOF. Suppose that all constraints are violated by x are redundant. Then we can remove these redundant constraints without changing the feasible region. Because the constraints that are violated by x are removed then x is satisfied by all remaining constraints in feasible region. Hence, the point x becomes feasible which is contradiction. $\square$

COROLLARY 2.4.4. *From Caron and Traynor in* [**1**], *if x is an infeasible point and it violates only one constraint, that constraint is necessary in each reduction.*

For $x \in \mathbb{R}^n$ we define the binary word observation $\delta(x) = (\delta_1(x), \ldots, \delta_m(x))$ such that

$$\delta_i(x) = \begin{cases} 0 & \text{if } x \in X_i \\ 1 & \text{if } x \in X_i^c \end{cases}$$

Now, we suppose that J is a reduction of I. Corresponding to this reduction we can define the binary word $y = (y_1, \ldots, y_m)^\top$ such that

$$y_i = \begin{cases} 1 & \text{if } i \in J \\ 0 & \text{if } i \notin J \end{cases}$$

From theorem 2.4.2, if x is an infeasible point and J is a reduction of I then we have

$$\delta(x)y \geq 1$$

Since $\delta(x)$ and y are binary words, we can view this as a constraint for the set covering SC problem [**24**]. Let E be a set covering matrix whose rows are indexed by all possible distinct observation $\delta(x) \neq 0$, then we have the following system

$$\begin{aligned} &Ey \geq \mathbf{1} \\ &y \in \{0,1\}^I, \end{aligned} \tag{2.4.2}$$

where $\mathbf{1}$ is a vector of ones. There is a connection between all solutions in system 2.4.2 and all reductions from a theorem in [**1**].

THEOREM 2.4.5. *From* [**23**] *the set J is a reduction of I if and only if y is a feasible solution to system 2.4.2.*

PROOF. We assume that $Z(I) \neq \mathbb{R}^n$ then there is at least one infeasible point. From theorem 2.4.2, J is a reduction of I if and only if for each $x \notin Z(I)$ there exists

$k \in J$ with $a_k^\top x > 0$. But that is same as saying that whenever $\delta(x) \neq 0$ there exists $k \in J$ with $y_k = 1$ and $\delta_k(x) = 1$, then $\delta(x)y \geq 1$. Thus, J is a reduction of I if and only if for all $\delta(x) \neq 0$ we have $\delta(x)y \geq 1$; that is $Ey \geq \mathbf{1}$. In other words, the set J is a reduction of I if and only if y is a feasible solution to system 2.4.2. □

So far we have shown that any feasible solution to system 2.4.2 gives a reduction J of I. When we reduce I to J, we are looking for smaller number of constraints. Consequently, we can obtain the smallest number of constraints by solving the standard set covering SC problem

$$\begin{aligned} &\text{minimize} \quad \Sigma y = \mathbf{1}^\top \cdot y \\ &\text{subject to} \quad Ey \geq \mathbf{1},\ y \text{ is a binary word,} \end{aligned} \tag{2.4.3}$$

hence, any optimal solution to 2.4.3 corresponds to an irreducible feasible system or (IIS).

For matrix E, we know that

(1) If columns i and j are identical, then constraints i and j are duplicate.

(2) If column i is a column of zeros, then constraint i is everywhere satisfied.

(3) If column i is a column of ones, then constraint i is everywhere violated.

DEFINITION 2.4.6. *The matrix F is a reduction of a the set covering matrix E if F is the subset of E such that for any binary words y, $Fy \geq 1$ implies $Ey \geq 1$.*

Definition 2.4.6 implies that if we replace E in the original set covering problem by F the new SC problem has the same feasible solutions.

LEMMA 2.4.7. *From [**1**], F is a reduction of E if and only if for any $e \in E$ there is $f \in F$ such that $f \leq e$*

PROOF. For proof see Lemma 5 in [**1**]. □

From lemma 2.4.7 we can say matrix E is irreducible if and only if no two elements of E are comparable.

DEFINITION 2.4.8. *Let e and e' be two rows of the set covering matrix E. If inequality $e'^{\top}y \geq 1$ is satisfied by y^* implies that inequality $e^{\top}y \geq 1$ is also satisfied by y^* then we say e majorizes e', that is, e is redundant and it should be removed from the matrix E.*

The general framework for finding the irreducible feasible system and irreducible infeasible system by Caron and Traynor [**1**] is based on the set covering matrix E. Suppose that we have an optimal solution y with an irreducible reduction J to the problem 2.4.3. Thus, if the family $\{X_i\}_{i \in I}$ is feasible, then J provides a irreducible feasible system and, if the family is infeasible then J corresponds to an irreducible infeasible system. In practice any non-zero component of y provides the elements of J. But unfortunately, determining the matrix E for a certain set of constraints is not easy.

2.5. Collecting the Set Covering Matrix E

Suppose that X is an open bounded subset of $\mathbb{R}^n$. One way of collecting the rows of the set covering matrix E is by sampling points $x \in X$ and calculating the corresponding observation $\delta(x)$. Here we provide some theorems to ensure that all $\delta(x)$ can be sampled with non-zero probability.

DEFINITION 2.5.1. *A probability distribution is supported on the set X if for every $b \in X$, every neighborhood of b will intersect X in a set of positive probability.*

DEFINITION 2.5.2. *Let $f : X \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$, a point $\hat{x} \in X$ is a local minimum of f if there exists $\epsilon > 0$ such that for any $x \in B(\hat{x}, \epsilon)$ we have $f(\hat{x}) \leq f(x)$.*

THEOREM 2.5.3. *From [**1**], suppose $J \subseteq I$ and let $A_i = \{x \in X \mid g_i(x) \leq 0\}$ where the g_i are continuous functions. Then we put $g_J(x) = \max_{j \in J} g_j(x)$, if zero is not a local minimum of any g_J then each non-zero value of δ will be sampled with non-zero value probability under any distribution supported on X.*

Proof. For proof see theorem 6 in [**1**]. □

DEFINITION 2.5.4. *Constraint i is an implicit equality in a set of linear homogeneous inequalities, if $a_i^\top x = 0$ for all $x \in Z(I)$.*

Consider the set of three linear inequalities given by

$$\begin{array}{rcrcll} -x_1 & + & x_2 & \leq & 0 & (1) \\ x_1 & + & x_2 & \leq & 0 & (2) \\ 0.5x_1 & - & x_2 & \leq & 0 & (3) \end{array}$$

that are graphed in the Figure 2.3. Here $I = \{1, 2, 3\}$ and $Z(I) = \{\mathbf{0}\}$. For all $x \in Z(I)$ we have $a_i^\top(\mathbf{0}) = 0$, then constraint (1), constraint (2) and constraint (3) are implicit equalities.

DEFINITION 2.5.5. *Let $f : X \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$, the point $\hat{x}$ is a global minimum of function f if $f(\hat{x}) \leq f(x)$ for any $x \in X$.*

Theorem 2.5.6. *Any local minimum of a convex function on a convex set is always a global minimum* [**25**].

Proof. Let $X \subseteq \mathbb{R}^n$ be a convex set and $f : X \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$. Also suppose that $\hat{x}$ is a local minimum of f. We will show $f(\hat{x}) \leq f(z)$ for any arbitrary $z \in X$ with $z \neq \hat{x}$. Since $\hat{x}$ is a local minimum of f then by definition 2.5.2 there exists $\epsilon > 0$ such that for any $x \in B(\hat{x}, \epsilon)$ we have $f(\hat{x}) \leq f(x)$. Since X is a convex set then by definition 1.2.2 we have $\lambda z + (1-\lambda)\hat{x} \in X$ for all $\lambda \in [0, 1]$. As $\lambda \longrightarrow 0^+$ then $(\lambda z + (1-\lambda)\hat{x}) \longrightarrow \hat{x}$. Hence, we can consider λ small enough such that $[\lambda z + (1-\lambda)\hat{x}] \in B(\hat{x}, \epsilon)$. Then we have

$$\begin{array}{rcll} f(\hat{x}) & \leq & f(\lambda z + (1-\lambda)\hat{x}) & (\text{ since } \hat{x} \text{ is local minimum of } f\ ) \\ & \leq & \lambda f(z) + (1-\lambda) f(\hat{x}) & (\text{ since } f \text{ is convex function}) \end{array}$$

then by rearranging terms we have $f(\hat{x}) \leq f(z)$. □

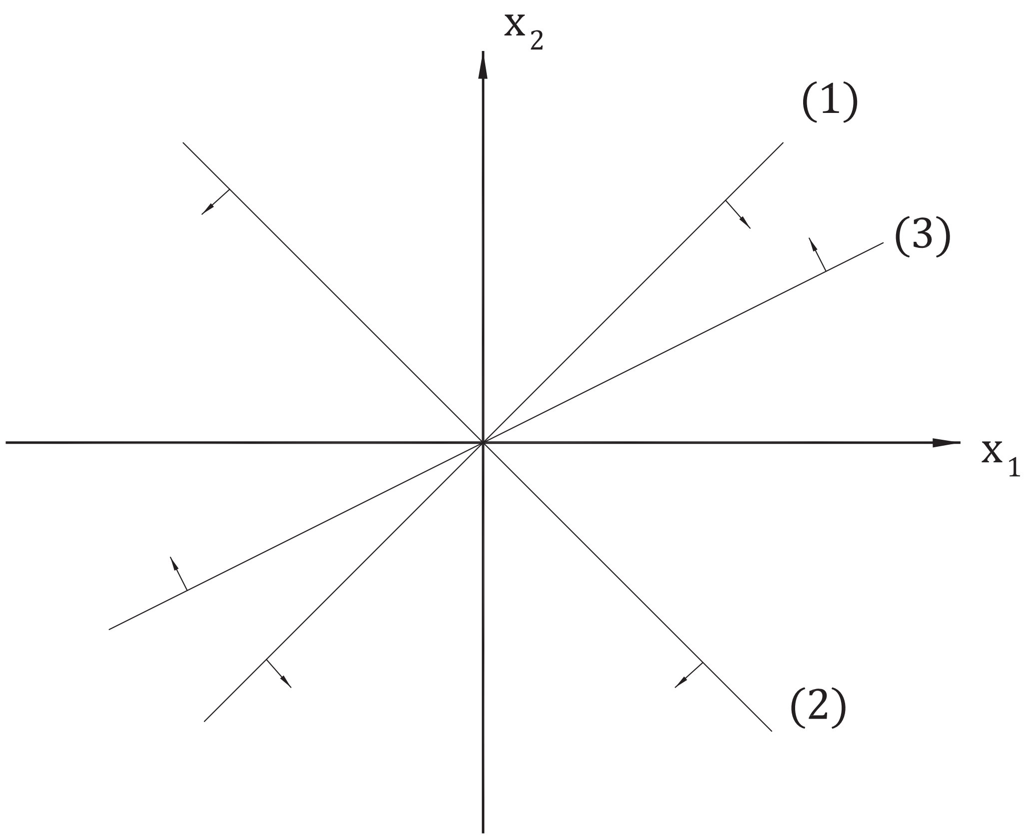


FIGURE 2.3. The example of implicit equalities.

COROLLARY 2.5.7. *From Caron and Traynor* [**1**], *if there are no implicit equalities in a set of linear inequality constraints then under any distribution supported on X, all non-zero values of δ are chosen with positive probability.*

PROOF. Suppose that $a_J^\top(x) = \max_{j \in J} a_j^\top x$ for any subset J of I. Suppose that 0 is a local minimum of $a_J^\top(x)$. Since any linear function is a convex function then, from theorem 2.5.6, 0 is a global minimum of $a_J^\top(x)$. Then $a_J^\top(x) = 0$ for all x in $Z(J)$. By definition 2.5.4 we conclude that each $a_j^\top(x)$ is an implicit equality for all x in $Z(J)$. Thus, if there are no implicit equalities then 0 is not a local minimum of any $a_J^\top(x)$, and by theorem 2.5.3 under any distribution supported on X, all non-zero values of δ are chosen with positive probability. □

2.6. Concluding Remarks

This chapter was about constraints classification. We have shown that, with probabilistic hit-and-run methods, we can detect necessary constraints in a particular set of homogeneous linear inequality system. Also we have used Boneh's set covering approach to introduce the framework for finding the irreducible feasible system and irreducible infeasible system [**1**]. Also we have explained the theoretical results for sampling points with non-zero probability. In the next chapter, we will use the connection between constraints analysis and the set covering problem by Boneh. We will present the algorithm to find a solution for the system of homogeneous linear inequality constraints.

CHAPTER 3

A Probabilistic Algorithm for the Solution of Homogeneous Linear Inequalities

3.1. Introduction

We present our proposed algorithm for the homogeneous linear inequality constraints in this chapter. This algorithm has two main steps, a sampling point step and a set covering SC problem step, these are described in detail in section 3.2. Convergence of the algorithm is explained in section 3.3. Then, we present the determination of the implicit equalities in section 3.4. Finally examples and numerical results are provided in section 3.5.

3.2. The Proposed Probabilistic Method

Prior to explaining the proposed probabilistic method, we need to introduce the new region B. Consider the following region

$$\mathcal{R}(A,c) = \{\, x \in B \,|\, c^{\top}x < 0,\ Ax \leq \mathbf{0}\} \tag{3.2.1}$$

where

$$B = \{\, x \in \mathbb{R}^n \,|\, -\mathbf{1} < x < \mathbf{1}\}$$

and $\mathbf{1}$ is n-vector of ones. The set B is an open bounded, full dimensional box in $\mathbb{R}^n$ with side length two and centred at the origin.

It is obvious that $\mathcal{R}(A,c) \subseteq \mathcal{R}_2(A,c)$. If there is a point in $\mathcal{R}_2(A,c)$ then by scaling, there is a one in $\mathcal{R}(A,c)$. Hence, we can restrict the feasible region of homogeneous linear inequality constraints in 1.1.4 within the box B. We want to introduce a method to find a minimal representation for the linear homogenous problems. It has been shown that probabilistic methods can be faster than deterministic methods to detect necessary constraints [**6**]. Then we are interested to introduce our algorithm based on probabilistic methods such as HR methods. We know that the feasible region for the linear homogenous problems are unbounded, if we want to use HR methods we need to reduce our feasible region to a bounded region like B. Sampling points uniformly in the box B is easy since all that is needed is independently generate each coordinate of the point uniformly from $(-1, 1)$. A proposed probabilistic method is presented that, with probability one, either finds a nonzero solution to (1.1.3 - 1.1.4) or determines that no such solution exists. In addition, this method collects all the information that can provide a minimal representation or an irreducible infeasible set (IIS) in the case of infeasibility. This method is based on HR methods and on the set covering paradigm. For simplicity of the proposed algorithm, we can define both systems 1.1.3 and 1.1.4 into one system such that the feasible region of this system denoted by Z and this region which may or may not be empty. Also we set $c = a_0$.

$$Z := \{x \in \mathbb{R}^n \mid x \neq \mathbf{0},\ a_0^\top x < 0,\ a_i^\top x \leq 0\}$$

If $a_0 = \mathbf{0}$ then we are finding a nonzero solution to $a_i^\top x \leq 0$; otherwise we are finding a solution to $a_i^\top x \leq 0$ with $a_0^\top x < 0$ making the condition $x \neq \mathbf{0}$ redundant.

We assume without loss of generality, that there are no duplicate constraints or implicit equalities. However if there are duplicate constraints, it can be removed by our algorithm or by deterministic methods [**26**]. Implicit equalities will be found by our algorithm and this is discussed in section 3.4. Also we assume that $a_0 \neq \mathbf{0}$ in this thesis. We start the algorithm with any nonzero point x^0 which is chosen uniformly in B. That is we choose independently $x_i^k \sim U(-1, 1)$ where $i = 1, 2, \ldots, n$ and x_i^k refer to the i-th component of x^k.

3.2.1 Sampling Points.

There are three main parts: a hit part, a run part and cleaning the set covering matrix part. In the hit part we need follow from step (0) to step (5), and for a run part we need follow step (6). Finally for cleaning the set covering matrix we need follow step (7).

Step (0): Initialization

In this step at first we set N as the iteration limit, we set the iteration counter $k = 0$, and we let $M^k = \{ i \mid a_i^\top x^k < 0,\ i \in I_0 \}$. Since $a_0 \neq \mathbf{0}$ we have $m+1$ inequalities. And we generate a random direction vector s^k as we did in Algorithm 1.

Step (1): Determination of the first row of the set covering matrix

In this step we calculate the corresponding observation x^k and we set $\delta = \delta(x^k)$. If $\delta(x^k) = 0$ then we are done and x^k is a feasible solution to our problem, though we may choose to continue the algorithm in order to provide an analysis of the constraints set. In this case we start with $E = \emptyset$, otherwise $\delta(x^k) \neq 0$ and we set $\delta(x^k) = E$.

Step (2): Computation of the step sizes

In this step, we calculate step sizes. We know that together, x^k and s^k define the line

$$L(x^k, s^k) = \{ x \in \mathbb{R}^n \mid x = x^k + \sigma s^k,\ \sigma \in \mathbb{R}\}$$

and the intersection of this line with B is the line segment

$$\hat{L}(x^k, s^k) = \{ x \in \mathbb{R}^n \mid x = x^k + \sigma s^k,\ \sigma_{-1} \leq \sigma \leq \sigma_{m+1}\}$$

where

$$\sigma_{-1} = -1 - x_\kappa^k \quad \text{and} \quad \sigma_{m+1} = 1 - x_\kappa^k.$$

Then we calculate the intersection points of the line segment $\hat{L}(x^k, s^k)$ with all of the inequality boundaries. Since $s^k = e_\kappa$, the intersection point with the boundary of the i-th constraint, $i \in I_0$ is determined as follows:

$$\begin{aligned} a_i^\top (x^k + \sigma_i s^k) = 0 \longleftrightarrow \sigma_i &= -(a_i^\top x^k / a_i^\top e_\kappa) \\ &= -(a_i^\top x^k / a_{i\kappa}) \qquad (\text{ where } a_{i\kappa} \neq 0). \end{aligned}$$

If $a_{i\kappa} = 0$ then there is no intersection between line segment $\hat{L}(x^k, s^k)$ and the boundary of the i-th constraint.

Step (3): Selection of the suitable step size

In this step we need to choose the suitable step size σ^k uniformly in the line segment $\hat{L}(x^k, s^k)$ in order to find the point x^{k+1} for the next iteration. Since x^{k+1} must be in the box B and must satisfy the constraints in M^k, we can change the interval $[\sigma_{-1}, \sigma_{m+1}]$ and we can choose x^{k+1} uniformly in the line segment:

$$L'(x^k, s^k) = \{\, x \in \mathbb{R}^n \mid x = x^k + \sigma s^k,\ \sigma_r \leq \sigma \leq \sigma_u\},$$

where

$$\sigma_r = \max\{\sigma_{-1}, \max\{\sigma_i : \sigma_i < 0,\ i \in M^k,\ a_{i\kappa} < 0\}\}$$

and

$$\sigma_u = \min\{\sigma_{m+1}, \min\{\sigma_i : \sigma_i > 0,\ i \in M^k,\ a_{i\kappa} > 0\}\}.$$

We select $\sigma^k \sim U(\sigma_r, \sigma_u)$.

Prior to proceeding to step (4) we need to know some things about the crossing points strategy.

(a) Crossing Points Strategy

The proposed algorithm is based on the solution of the set covering problem where each row of the set covering matrix corresponds to a sample point, and is determined by the constraint satisfaction at that sampled point, therefore the sampling methodology is very important to this algorithm. Hence, we want to generate as many observations as we can for the matrix E. One way to do this is to sample along the line segment $L'(x^k, s^k)$. This is done by taking advantage of the constraint intersection points as suggested in Caron et. al. [**3**]. From the constraint intersection points, we have the increasing ordered values

$$\lambda_{\ell^-} < \lambda_{\ell^-+1} < \ldots < \lambda_{-1} < 0 < \lambda_1 < \ldots < \lambda_{\ell^+-1} < \lambda_{\ell^+}$$

where $\lambda_{\ell^+} = \sigma_{m+1} > 0$ and $\lambda_{\ell^-} = \sigma_{-1} < 0$ and every λ_ι, $\iota = \ell^- + 1, \ldots, \ell^+ - 1$, corresponds, and is equal to, one of σ values when the line segment $\hat{L}(x^k, s^k)$ crosses a constraint boundary inside the box B.

At the crossing point λ_ι we denote by i_ι the index of the constraint whose boundary is crossed, i.e, $a_{i_\iota}^\top(x^k + \sigma_\iota s^k) = 0$. When the parameter σ increases beyond λ_ι, the i_ι-th bit of $\delta(x^k + \sigma s^k)$, changes (flips) from 0 to 1 or from 1 to 0 . This help us to construct the rows of E by generating an observation for each interval of the line segment $\hat{L}(x^k, s^k)$ between the crossing points.

As we need to clean the matrix E in step (7) of the proposed algorithm. For simplicity of the final step we can use the "cleaning as we cross" strategy in [**3**]. If we want to follow this strategy, we will append the non-zero observation $\delta(x^k)$ to E, only if a bit of current observation changes from 1 to 0. We do not append the zero observation to E, as the constraint $\mathbf{0}^\top y \geq 1$ is infeasible.

We demonstrate this strategy with Figure 3.1. In Figure 3.1 we have four homogenous inequality constraints. The current iterate is x^0, the direction is s^0 and the line segment is AB. The observation in the region containing x^0 is denoted by $\delta_0 = (1, 1, 0, 0)$. When we start from x^0 and move to the right, we cross the boundaries of constraints (1) and (2). As we cross the boundary of constraint (1) the first bit of the observation δ_0 changes from 1 to 0 and we get $\delta_1 = (0, 1, 0, 0)$. Since δ_0 majorizes δ_1 we remove δ_0. As we continue in the right direction we cross the boundary of constraint (2) and we get $\sigma_2 = (0, 0, 0, 0)$ but we do not append this observation to E. As we start from x^0 and move to the left, we cross the boundaries of constraints (3) and (4). We get $\delta_3 = (1, 1, 1, 0)$ and $\delta_4 = (1, 1, 1, 1)$. Since δ_0 is majorized by δ_3 and δ_3 is majorized by δ_4 we can remove both δ_3 and δ_4. Only the observation δ_1 will be appended to E.

Step (4): Ordering the value of step sizes

In this step we sort the value of those σ_i, $i \in I_0$ which are in the interval $[\sigma_r, \sigma_u]$, in increasing order, as described in the description of the crossing point strategy.

Step (5): Collect the rows of the set covering matrix

In this step we follow the crossing point strategy to collect the rows of the set covering matrix E. We start with $\delta = \delta(x^k)$ and we do the following loop.

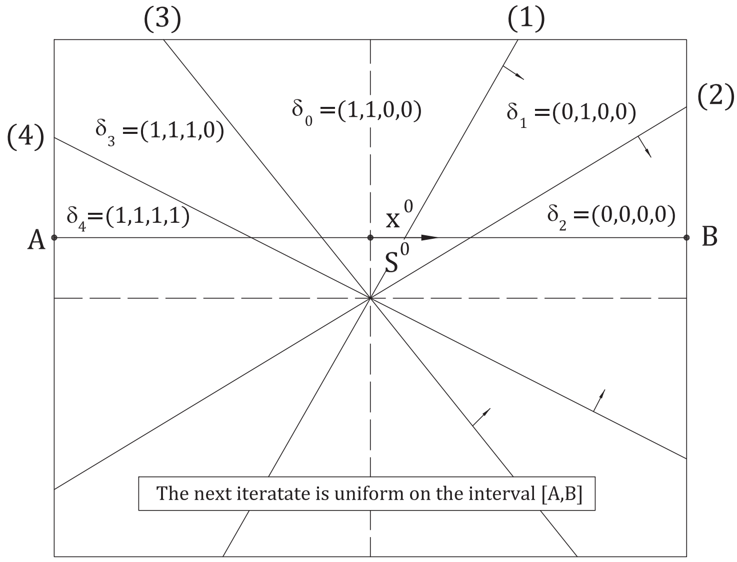


FIGURE 3.1. Crossing point strategy.

For $\iota = 1, \ldots, \ell^+$ where $\lambda_{\ell^+} = \sigma_u$

Create $\hat{\delta}$ from δ by flipping the i_ι bit of δ

If $\hat{\delta}_{i_\iota} = 0$, $\hat{\delta} \neq 0$ and $\hat{\delta}$ is not a row of E append $\hat{\delta}$ to E

If $\hat{\delta}_{i_\iota} = 0$, $\hat{\delta} \neq 0$ and δ is a row of E, remove δ from E.

Flip δ_{i_ι}

Repeat the loop for $\iota = -1, \ldots, \ell^-$ where $\lambda_{\ell^-} = \sigma_r$ starting again with $\delta = \delta(x^k)$

Step (6): Update

In the run step we update $x^{k+1} = x^k + \sigma^k s^k$, $a_i^\top x^{k+1} = a_i^\top x^k + \sigma^k a_i^\top s^k$, and M^{k+1}. Until $k \leq N$, we repeat steps (1) to (6).

Step (7): Clean the set covering matrix

In this step we need to remove those rows of the matrix E that are redundant. In this step, we use singleton rows of E, if any exist, to remove redundancies in E. For example, if a row of E has a one only in column k, then all rows with a one in column k are redundant and can be removed.

3.2.2 Set Covering.

In this step, we solve the set covering problem 2.4.3. Since any feasible solution y gives a reduction then we do not need to find an optimal solution for 2.4.3 to get a benefit. We use Chavatal's greedy algorithm [**24**] to get our solution.

(a) Greedy Algorithm

If we let $E = (e_{ij})$ where E is the set covering matrix of order $(r \times m)$ such that

$$e_{ij} = \begin{cases} 1 & \text{if } i \in p_j \\ 0 & \text{otherwise} \end{cases}$$

so m columns of E are the incidence vectors of $p_1, \ldots, p_m$. In this algorithm we consider the finite sets $p_1, \ldots, p_m$ and we denote $\bigcup(p_j : 1 \leq j \leq m)$ by I such that $I = \{1, 2, \ldots, r\}$, $J = \{1, 2, \ldots, m\}$ and we need to find a subset J^* of J such that $\bigcup(p_j : j \in J^*) = I$. Then J^* in Chavatal's greedy algorithm is called a cover for the problem 2.4.3. In fact any elements of J^* provides the non-zero component of y.

Algorithm 2 The Greedy Algorithm for the Set Covering Problem

Given finite sets $p_1, p_2, \ldots, p_m$

Set $J^* = \emptyset$

If $p_j = \emptyset$ for all $j \in J$ then stop, and J^* is a cover

Otherwise find the index k maximizing $|p_j|$, $j \in J$, Where $|p_j|$ denotes the cardinality of p_j

Set $J^* = J^* \bigcup \{k\}$

set $p_j = p_j \setminus p_k$, where the set $p_j \setminus p_k$ is the set p_j with the elements of the set p_k being removed.

EXAMPLE 1. Find a feasible solution for the set covering problem 2.4.3 such that matrix E is defined as following:

$$E = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

We have $p_1 = \{4\}$, $p_2 = \{1, 2\}$, $p_3 = \{1\}$, $p_4 = \{3\}$, $p_5 = \emptyset$, $p_6 = \emptyset$ and $p_7 = \{2, 3, 4\}$. Since matrix E has four rows and seven columns then we set $I = \{1, 2, 3, 4\}$ and $J = \{1, 2, 3, 4, 5, 6, 7\}$, and we start the Algorithm 2.

Iteration (0)

We set $J^* = \emptyset$, since all sets p_j for all $j \in J$ are not empty then we find the maximum index k such that set p_k has a maximum cardinality, that is

$p_k = \max\{|p_1|, |p_2|, |p_3|, |p_4|, |p_7|\} = \max\{1, 2, 1, 1, 3\} = 3$. Then $k = 7$ and we set $J^* = \{7\}$ also, we calculate new p_1, p_2, p_3, p_4 and p_7 such that

$$p_1 = p_1 \setminus p_7 = \emptyset$$

$$p_2 = p_2 \setminus p_7 = \{1\}$$

$$p_3 = p_3 \setminus p_7 = \{1\}$$

$$p_4 = p_4 \setminus p_7 = \emptyset$$

$$p_7 = p_7 \setminus p_7 = \emptyset$$

Iteration (1)

We have $J^* = \{7\}$ since sets p_2 and p_3 are not empty then we set $p_k = \max\{|p_2|, |p_3|\} = \max\{1, 1\} = 1$ then we have either index $k = 2$ or index $k = 3$. If we take the index $k = 2$ then we set $J^* = \{7, 2\}$ and new p_2, p_3 are calculated as follows:

$$p_2 = p_2 \setminus p_2 = \emptyset$$

$$p_3 = p_3 \setminus p_2 = \emptyset$$

Iteration (2)

We have $J^* = \{7, 2\}$ since all sets $p_j = \emptyset$ for all $j \in J$ we stop, then $J^* = \{7, 2\}$ is a cover for I in the Example 1, it is obvious that $p_7 \bigcup p_2 = I$. Then $y = (0, 1, 0, 0, 0, 0, 1)$ is a feasible solution for the Example 1. Similarly if we take index $k = 3$ then $J^* = \{7, 3\}$ is another cover for I and $y = (0, 0, 1, 0, 0, 0, 1)$ is an other feasible solution for the Example 1. □

3.3. Convergence of the Algorithm

In this part we show that our variation of hit and run method guaranties a feasible solution if one exists. We define M^k as the set of all strict inequalities satisfied at x^k.

$$M^k = \{\ i \mid a_i^\top x^k < 0,\ i \in I_0\ \}.$$

We define C^k be the intersection of those inequality regions containing x^k. From [**3**] we have the following theorem.

THEOREM 3.3.1. *If $Z \neq \emptyset$, the proposed algorithm will, with probability one, find a point $x \in Z$.*

PROOF. Suppose that the feasible region $Z \neq \emptyset$. The initial point x^0 will be chosen uniformly in the bounding box B. Let C^0 be the intersection of those inequality regions containing x^0, and let M^0 be the set of all strict inequalities satisfied at x^0. When we start the algorithm then point x^1 will be found in C^1 with corresponding M^1. The choice of σ^0 ensure that $M^0 \subseteq M^1$. Suppose we continue the algorithm until we find a point x^k in a region C^k with set M^k, such that $M^{k-1} \subseteq M^k$. Since our proposed algorithm is uniform in C^k, if the region C^k is not feasible then point x^{k+1} will be found in C^{k+1} with corresponding M^{k+1}, $M^k \subseteq M^{k+1}$. Finally all of $m + 1$ constraints will be used in the algorithm, then the feasible point will be found. □

Algorithm 3 Algorithm for the solution of homogeneous linear inequalities

Given a matrix A such that $A^\top = [a_0, \ldots, a_m]$, and a point $x^0 \in B$. Set $k = 0$

while $k \leq N$ **do**

Select an integer κ uniformly from $\{1, \ldots, n\}$ and set s^k to be the κ-th coordinate vector.

Set $M^k = \{ \ i \mid a_i^\top x^k < 0, \ i \in I_0 \ \}$

Determine intersection points σ_{-1}, σ_{m+1} and all σ_i, $i \in I_0$

Determine σ_r and σ_u

Select $\sigma^k \sim U(\sigma_r, \sigma_u)$

From the intersection points list and order all σ_i which are in the interval $[\sigma_r, \sigma_u]$

Let $\delta = \delta(x^k)$ and

for $\iota = 1, \ldots, \ell^+$ **do**

Create $\hat{\delta}$ from δ by flipping the i_ι bit of δ

If $\hat{\delta}_{i_\iota} = 0$, $\hat{\delta} \neq 0$ and $\hat{\delta}$ is not a row of E append $\hat{\delta}$ to E

If $\hat{\delta}_{i_\iota} = 0$, $\hat{\delta} \neq 0$ and δ is a row of E, remove δ from E.

Flip δ_{i_ι}

end for

Repeat the loop for $\iota = -1, \ldots, \ell^-$ starting again with $\delta = \delta(x^k)$

Replace x^k with $x^k + \sigma^k s^k$

Set k to $k + 1$

end while

Clean the matrix E

Solve the set covering problem to find y. For any nonzero component y_j of y, list the j-th constraint as necessary. Otherwise list the j-th constraint as redundant.

In the proposed algorithm, after feasibility is found we continue the algorithm, which would be equivalent to Telgen's CD algorithm, until N iterations are complete. We then solve the SC problem.

3.4. Determination of Implicit Equality

The proposed algorithm is designed for homogeneous linear inequality system without implicit equalities. In this section we explain how this algorithm can identify the implicit equalities in our problem. In step (2), suppose that $\sigma_i^k = \sigma_j^k$. Then either constraints i and j have the same boundaries or the line segment $\hat{L}(x^k, s^k)$ hits the intersection point of these two boundaries, which has probability zero. Thus, constraints i and j might, together, be implicit equalities. This can be checked algebraically.

3.5. Examples and Numerical Results

We implemented the algorithm using MATLAB 7.2 educational software. We discuss an implementation of the algorithm and analyze the performance of this implementation in this section. All examples were run on a computer with Pentium 4 processor (2.2 GHz, 3GB RAM). At first we solve very simple examples graphically, then we test them using our software in order to verify the results.

Consider a set of four homogeneous linear inequality constraints given by

$$\begin{array}{ccccccc} x_1 & + & x_2 & \leq & 0 & (1) \\ x_1 & - & x_2 & \leq & 0 & (2) \\ x_1 & & & \leq & 0 & (3) \\ x_2 & & & < & 0. & (4) \end{array}$$

The constraints graphed in Figure 3.2. The shaded area of Figure 3.2 is the set of solutions for this problem. From Figure 3.2 it is obvious that constraints (2) and (4) are necessary and constraints (1) and (3) are redundant.

Now we use the algorithm to solve the problem and we expect to get the same results, we start the algorithm with $x_0 = [0.9562, 0.2810]^\top$ which is chosen randomly in B.

Iteration (0)

Step (0): $N = 4$, $s^0 = [1, 0]^\top$, $M^0 = \{\}$.

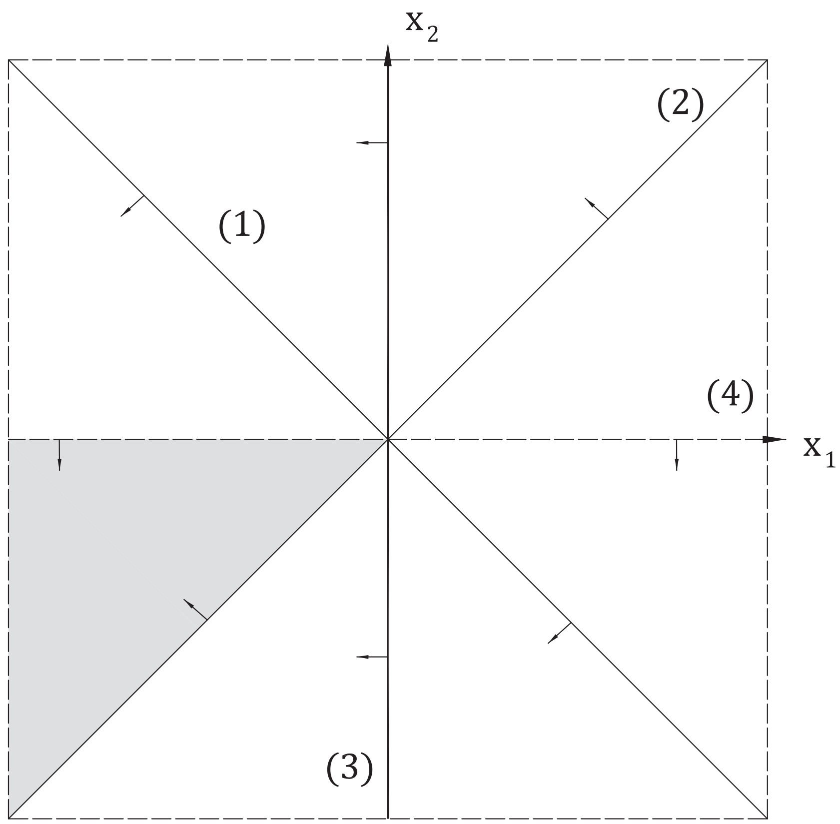


FIGURE 3.2. The example of linear homogeneous inequalities.

Step (1): $\delta(x^0) = (1, 1, 1, 1)$, and

$$E = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}.$$

Step (2): $\sigma_{-1} = -1.9562$, $\sigma_1 = -1.2372$, $\sigma_2 = -0.6752$, $\sigma_3 = -0.9562$, $\sigma_4 = 0.0438$,

and $\sigma_r = -1.9562$, $\sigma_u = 0.0438$.

Step (3): $\sigma^0 = -0.5935$.

Step (4): $-1.9562 < -1.2372 < -0.9562 < -0.6752 < 0 < 0.0438$.

Step (5): For $\iota = 2$ we have $\hat{\delta} = (1, 0, 1, 1)$ and after clean as cross we have

$$E = \begin{pmatrix} 1 & 0 & 1 & 1 \end{pmatrix}.$$

For $\iota = 3$ we have $\hat{\delta} = (1,0,0,1)$ and after clean as cross we have

$$E = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}.$$

For $\iota = 1$ we have $\hat{\delta} = (0,0,0,1)$ and after clean as cross we have

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}.$$

Step (6): $x^1 = [0.9562, 0.2810]^\top + (-0.5935)[1,0]^\top = [0.3627, 0.2810]^\top$.

Iteration (1)

Step (0): $s^1 = [0,1]^\top$, $M^1 = \{\}$.

Step (1): $\delta(x^1) = (1,1,1,1)$, and

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Step (2): $\sigma_{-1} = -1.2810$, $\sigma_0 = -0.2810$, $\sigma_1 = -0.6437$, $\sigma_2 = 0.0817$, $\sigma_4 = 0.7190$, and $\sigma_r = -1.2810$, $\sigma_u = 0.7190$.

Step (3): $\sigma^1 = -1.0116$.

Step (4): $-1.2810 < -0.6437 < -0.2810 < 0 < 0.0817 < 0.7190$.

Step (5): For $\iota = 2$ we have $\hat{\delta} = (1,0,1,1)$ and

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix},$$

and after clean as we cross, we have

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}.$$

For $\iota = 0$ we have $\hat{\delta} = (1,1,1,0)$ and

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

For $\iota = 1$ we have $\hat{\delta} = (0, 1, 1, 0)$ and

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix},$$

and after clean as we cross, we have

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

Step (6): $x^2 = [0.3627, 0.2810]^\top + (-1.0116)[0, 1]^\top = [0.3627, -0.7306]^\top$.

Iteration (2)

Step (0): $s^2 = [1, 0]^\top$, $M^2 = \{1, 4\}$.

Step (1): $\delta(x^2) = (0, 1, 1, 0)$, and

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

Step (2): $\sigma_{-1} = -1.3627$, $\sigma_1 = 0.3679$, $\sigma_2 = -1.0932$, $\sigma_3 = -0.3627$, $\sigma_4 = 0.6373$, and $\sigma_r = -1.3627$, $\sigma_u = 0.3679$.

Step (3): $\sigma^2 = -0.9089$.

Step (4): $-1.3627 < -1.0932 < -0.3627 < 0 < 0.3679$.

Step (5): For $\iota = 3$ we have $\hat{\delta} = (0, 1, 0, 0)$ and

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

and after clean as we cross, we have

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

For $\iota = 2$ we have $\hat{\delta} = (0, 0, 0, 0)$.

Step (6): $x^3 = [0.3627, -0.7306]^\top + (-0.9089)[1, 0]^\top = [-0.5462, -0.7306]^\top$.

Iteration (3)

Step (0): $s^3 = [1, 0]^\top$, $M^3 = \{1, 4, 3\}$.

Step (1): $\delta(x^3) = (0, 1, 0, 0)$, and

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Step (2): $\sigma_{-1} = -0.4538$, $\sigma_1 = 1.2768$, $\sigma_2 = -0.1843$, $\sigma_3 = 0.5462$, $\sigma_4 = 1.5462$, and $\sigma_r = -0.4538$, $\sigma_u = 0.5462$.

Step (3): $\sigma^3 = -0.3844$.

Step (4): $-0.4538 < -0.1843 < 0 < 0.5462$.

Step (5): For $\iota = 2$ we have $\hat{\delta} = (0, 0, 0, 0)$.

Step (6): $x^4 = [-0.5462, -0.7306]^\top + (-0.3844)[1, 0]^\top = [-0.9307, -0.7306]^\top$.

Iteration (4)

Step (0): $s^4 = [1, 0]^\top$, $M^4 = \{1, 4, 3, 2\}$.

Step (1): $\delta(x^4) = (0, 0, 0, 0)$ then point x^4 is a feasible solution. Also we see that $M^0 \subseteq M^1 \subseteq M^2 \subseteq M^3 \subseteq M^4$ and $M^4 = I$. As we get the correct results then there is no need to complete this iteration.

step (7): We clean the matrix E to get

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

To complete the algorithm, it remains to solve the set covering problem.

For this step we use the Greedy algorithm to get $y = (0, 1, 0, 1)$, indicating that constraints 2 and 4 are necessary.

For constraint analysis the numerical results from the algorithm are corrected. However the correct result can not always be expected from the algorithm since this method is probabilistic. If we start the algorithm for the first example with a different initial point, say $x_0 = [0.3951, -0.3077]^\top$ then we would have the following steps.

Iteration (0)

Step (0): $N = 4$, $s^0 = [1, 0]^\top$, $M^0 = \{4\}$.

Step (1): $\delta(x^0) = (1, 1, 1, 0)$, and

$$E = \begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix}.$$

Step (2): $\sigma_{-1} = -1.3950$, $\sigma_1 = -0.0873$, $\sigma_2 = -0.7027$, $\sigma_3 = -0.3950$, $\sigma_4 = 0.6050$, and $\sigma_r = -1.3950$, $\sigma_u = 0.6050$.

Step (3): $\sigma^0 = 0.4727$.

Step (4): $-1.3950 < -0.7027 < -0.3950 < -0.0873 < 0 < 0.6050$.

Step (5): For $\iota = 1$ we have $\hat{\delta} = (0, 1, 1, 0)$ and

$$E = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix},$$

and after clean as we cross, we have

$$E = \begin{pmatrix} 0 & 1 & 1 & 0 \end{pmatrix}.$$

For $\iota = 3$ we have $\hat{\delta} = (0, 1, 0, 0)$ and

$$E = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

and after clean as we cross, we have

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}.$$

For $\iota = 2$ we have $\hat{\delta} = (0, 0, 0, 0)$.

Step (6): $x^1 = [0.3951, -0.3077]^\top + (0.4727)[1, 0]^\top = [0.8678, -0.3077]^\top$.

Iteration (1)

Step (0): $s^1 = [0, 1]^\top$, $M^1 = \{4\}$.

Step (1): $\delta(x^1) = (1, 1, 1, 0)$, and

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Step (2): $\sigma_{-1} = -0.6923$, $\sigma_0 = 0.3077$, $\sigma_1 = -0.5601$, $\sigma_2 = 1.1755$, $\sigma_4 = 1.3077$, and $\sigma_r = -0.6923$, $\sigma_u = 0.3077$.

Step (3): $\sigma^1 = -0.1589$.

Step (4): $-0.6923 < -0.5601 < 0 < 0.3077$.

Step (5): For $\iota = 1$ we have $\hat{\delta} = (0, 1, 1, 0)$ and

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix},$$

and after clean as we cross, we have

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

Step (6): $x^2 = [0.8678, -0.3077]^\top + (-0.1589)[0, 1]^\top = [0.8678, -0.4666]^\top$.

Iteration (2)

Step (0): $s^2 = [1, 0]^\top$, $M^2 = \{4\}$.

Step (1): $\delta(x^0) = (1, 1, 1, 0)$, and

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Step (2): $\sigma_{-1} = -1.8678$, $\sigma_1 = -0.4012$, $\sigma_2 = -1.3343$, $\sigma_3 = -0.8678$, $\sigma_4 = 0.1322$,

and $\sigma_r = -1.8678$, $\sigma_u = 0.1322$.

Step (3): $\sigma^2 = -1.6202$.

Step (4): $-1.8678 < -1.3343 < -0.8678 < -0.4012 < 0 < 0.1322$.

Step (5): For $\iota = 1$ we have $\hat{\delta} = (0, 1, 1, 0)$ and

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix},$$

and after clean as we cross, we have

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

For $\iota = 3$ we have $\hat{\delta} = (0, 1, 0, 0)$ and

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

and after clean as we cross, we have

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

For $\iota = 2$ we have $\hat{\delta} = (0, 0, 0, 0)$.

Step (6): $x^3 = [0.8678, -0.4666]^\top + (-1.6202)[1, 0]^\top = [-0.7524, -0.4666]^\top$.

Iteration (4)

Step (0): $s^4 = [1, 0]^\top$, $M^4 = \{4, 1, 2, 3\}$.

Step (1): $\delta(x^1) = (0, 0, 0, 0)$, and x^4 is a feasible solution, for constraint analysis we continue Telgen's CD algorithm. Then we have $\sigma_r = -0.2476$, $\sigma_u = 0.2858$. Since $u = 2$ then constraint (2) is necessary.

Step (7):

$$E = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}.$$

In set covering step we get $y = (0, 1, 0, 0)$, giving that constraint (2) is necessary. We have now shown one example with two different results. In this example the four constraints partition B into 8 full dimensional regions. If the algorithm had recorded all of the possible observations from these regions then the set covering matrix E would have been complete and the algorithm would identify all necessary constraints. Otherwise the set covering matrix E will have missing rows and the algorithm may not identify all necessary constraints.

We tested the algorithm on 12 feasible examples that were randomly generated. The examples and results are described in table 1. The number of constraints is m which consists of $m - 1$ inequalities and one strict inequality. The columns give the example number, the number of variables n, the number of constraints m, the number of identified necessary constraints $\hat{m}$, the number of iterations to get feasibility F and the number of iterations limit N.

Table 1. Description of the Examples and Numerical Results

EX.	n	m	$\hat{m}$	F	N
Ex.1	5	5	5	22	1000
Ex.2	5	10	8	3	1000
Ex.3	5	50	16	31	1000
Ex.4	10	10	10	18	2000
Ex.5	10	20	19	118	2000
Ex.6	10	100	27	639	2000
Ex.7	50	50	32	1340	8000
Ex.8	50	100	63	2857	8000
Ex.9	50	500	114	29541	50000
Ex.10	100	100	88	10000	20000
Ex.11	100	200	122	20221	30000
Ex.12	100	1000	890	44552	60000

CHAPTER 4

Conclusion

In this chapter, conclusions are presented based on the proposed algorithm and numerical investigations in chapter 3. The following items shows the major contributions of this thesis work:

• We developed a method for implementing Boneh's set covering problem to find a feasible solution to the homogeneous linear inequality systems. The applications of this work is described in detail in chapter 1.

• We analyzed the set of homogeneous linear inequality constraints as redundant and necessary by our method. This work is applicable for linear programs. The removal of redundant constraints can improve the performance of interior point algorithms. This can be increased the size of the problems which has to solve.

• The advantage of the proposed algorithm is that we do not need the interior feasible point by comparison to other basic hit and run methods.

• The numerical results show that the proposed algorithm can solve the problems with varying sizes and dimensions.

Bibliography

[1] R. J. Caron, T. Traynor, A general framework for the analysis of sets of constraints, in: S. Neogy, R. Bapat, A. Das, T. Parthasarathy (Eds.), Mathematical Programming and Game theory for Decision Making, World Scientific, 2007, pp. 33–45.

[2] A. Boneh, Identification of redundancy by a set-covering equivalence, in: J. Brans (Ed.), Operational Research '84, Elsevier Science Publishers B.V.(North Holland), Amsterdam, 1984, pp. 407–422.

[3] R. J. Caron, S. Jibrin, T. Traynor, Contraint analysis of linear matrix inequalities, INFORMS Journal of Computing 22 (2010) 144–15 3.

[4] A. Boneh, A. Golan, Constraints redundancy and feasible region boundedness by a random feasible point generator (RFPG), Paper presented at EURO III, Amsterdam.

[5] R. L. Smith, Efficient monte carlo procedures for generating points uniformly distributed over bounded regions, Operations Research 32 (1984) 1296–1308.

[6] H. Berbee, C. Boender, A. H. Rinnooy Kan, H. Romeijn, C. Scheffer, R. L. Smith, J. Telgen, Hit-and-run algorithms for the identification of nonredundant linear equalities, Mathematical Programming 37 (1987) 184–207.

[7] J. Farkas, Über die theorie der einfachen ungleichungen, Journal für die Reine und Angewandte Mathematik 124 (1902) 1 – 24.

[8] M. Weitzman, An economics proof of the supporting hyperplane theorem, Economics Letters 68 (2000) 1–6.

[9] W. Diewert, Chapter 3: Convex sets and concave functions (2008).
URL `http://faculty.arts.ubc.ca/ediewert/581ch3.pdf`

[10] D. Bertsekas, A. Nedic, A.Ozdaglar, Convex Analysis and Optimzation, Athena Scientific, 2003.

[11] D. Bertsekas, Convex optimization theory, chapter 2 exercises and solutions: Extended version, massachusetts institute of technology.
URL `http://www.athenasc.com/convexdualitysol2.pdf`

[12] R. Bland, New finite pivoting rules for the simplex method, Mathematics of Operations Research 2 (1977) 103–107.

[13] E. Andersen, K. Andersen, Presolving in linear programming, Mathematical Programming 71 (1995) 221–245.

[14] J. T. Mark H. Karwan, Vahid Lotfi, S. Zionts (Eds.), Redundancy in mathematical programming, Springer-Verlag, Berlin, 1983.

[15] R. J. Caron, J. F. McDonald, C. M. Ponic, A degenerate extreme point strategy for the classification of linear inequalities as redundant or necessary, Journal of Optimization Theory and Applications 62 (1989) 225–237.

[16] R. Caron, W. Obuchowska, An algorithm to determine boundedness of quadratically constrained convex quadratic programmes, European Journal of Operational Research 80 (1995) 431–438.

[17] R. Caron, W. Obuchowska, Unboundedness of a convex quadratic function subject to concave and convex quadratic constraints, European Journal of Operational Research 63 (1992) 114–123.

[18] V. Vasilyeva, An analysis of murty's proposeded interior point method for quadratic programming, Master's thesis, University of Windsor (2008).

[19] K. Murty, A new practically efficient interior point method for quadratic programming.

[20] J. Telgen, The CD algorithm, private communication (1979).

[21] Monte Carlo procedures for generating random feasible solutions to mathematical programs.

[22] C. Bélisle, H. Romeijn, R. Smith, Hit-and-run algorithms for generating multivariate distributions, Mathematics of Operations Research 18 (1993) 255–266.

[23] J. Feng, Redundancy in nonlinear systems: a set covering approach, Master's thesis, University of Windsor (1999).

[24] V. Chvatal, A greedy heuristic for the set-covering problem, Mathematics of Operations Research 4 (1979) 233–235.

[25] S. Cheng, local minimum of convex function is necessarily global (2006).
URL `http://planetmath.org/encyclopedia/ExtremalValueOfConvexconcaveFunctions.html`

[26] J. A. Tomlin, J. F. Welch, Finding duplicate rows in a linear programming model, Operations Research Letters 5 (1986) 7–11.

Vita Auctoris

Mrs. Mina Asgari Moslehabadi was born in Tehran, Iran. She got her B.Sc. degree in Pure Mathematics at the Teaching and Training University of Tehran. She expects to graduate with a Master of Science degree in Mathematics at the University of Windsor in spring 2010.