To the Graduate Council:

I am submitting herewith a thesis written by Mei-Hui Wang entitled "Methods For Solving Linear Semi-Infinite Programming Problems." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Yueh-er Kuo

Yueh-er Kuo, Major Professor

We have read this thesis
and recommend its acceptance:

Don Hinton

Accepted for the Council:

Associate Vice Chancellor and
Dean of The Graduate School

METHODS FOR SOLVING LINEAR SEMI-INFINITE PROGRAMMING PROBLEMS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Mei-Hui Wang

May 1996

ACKNOWLEDGMENTS

The author is extremely grateful to Dr. Yueh-er Kuo for her generous help and guidance in preparing this thesis. Appreciation is also expressed to Dr. Don Hinton and Dr. Xiaobing Feng for their time reading this thesis.

The author would also like to thank her family for their love and support as well as her good buddies Tina, Tony, and her English teachers at Laurel Church.

ABSTRACT

This thesis discusses a class of linear semi-infinite programming problems with finite number of variables and infinitely many constraints over a compact metric space. The "adding constraint method" for solving linear semi-infinite programming problems is introduced in Chapter I. The "perturbation method" for solving regular linear programming problems is introduced in Chapter II. Based on these two methods, the "perturbation method" for solving linear semi-infinite programming problems is proposed with a proof for the convergence of the "perturbation algorithm". Some numerical examples are also included in this thesis.

PREFACE

Semi-infinite programming deals with the problem of minimizing a function $f(x_1, \cdots, x_n)$ of variables $x_1, \cdots, x_n$, where $x_1, \cdots, x_n$, are required to meet an inequality $g(x_1, \cdots, x_n; t) \geq b(t)$ for every t in some set T. In case of a finite set T, this is a usual optimization problem with a finite number of inequality constraints on the variables $x_i, \cdots, x_n$, but infinite sets T occur quite naturally in many applications. For instance, in controlling air-pollution in some region T, let $x_1, \cdots, x_n$ be the costs incident to provisions for reducing the emission of some substance. Then the total cost $f(x_1, \cdots, x_n) = \sum_{j=1}^{n} x_j$ is to be minimized subject to the restriction that the concentration $g(x_1, \cdots, x_n; t)$ remains above some given standard $b(t)$ for every $t \in T$.

Some constraints should be considered in dealing with these applications. The constraints of the problems depend on the time or the space coordinates and therefore, may be formulated as semi-infinite programming problems. Another question is whether a semi-infinite model should be used in practice instead of a discrete one. The solution of the latter is near to that of the first if sufficient points are taken into account. In fact, if it were true that generally a discrete problem could be treated more easily and efficiently, then it would not be worthwhile to consider semi-infinite programming problems from a practical point of view. However, there are two aspects which make the problem interesting in practice :

1. A model with constraints depending on some parameter t (time, space, etc.) given by means of one inequality $g(x_1, \cdots, x_n, t) \geq b(t)$ rather than a big

number of unrelated inequalities $g_i(x_1, \cdots, x_n) \geq b_i$ is preferable with respect to storage requirements as well as simplicity of handling. Even in cases where the second description is given initially (for instance due to measurement restrictions) it may be advantageous to transform the data by some approximation procedure into the first form.

2. In general, at the optimal point, the inequalities are binding (i.e., equality holds) only in a finite number of points $t_1, t_2, \cdots, t_r$, with $r \geq n$ as a rule. In many cases, especially if T is a subset of Euclidian space $R^m, m \geq 2$, this leads to a much higher efficiency of continuous methods, because it is sufficient to consider only a small number of points in each iteration and, moreover, the points for the next step can be computed very efficiently by using derivatives with respect to t for instance.

This paper discusses the linear semi-infinite programming problems. Chapter I provides an "adding constraint" method for solving linear semi-infinite programming problems. Basically, it constructs a sequence of finite linear programming problems whose solutions converge to an optimal solution of a linear semi-infinite programming problem. Chapter II describes the "perturbation" method for solving finite linear programming problems. The basic idea of the method is to incorporate a barrier function into the linear objective function such that the minimizer of the subproblem stays in the interior of the original feasible domain. By parameterizing the barrier function, the corresponding minimizers form a path that leads to the optimal solution of the linear program. Following the path and gradually reducing the parameter, the perturbation algorithm works. Chapter III outlines a new method to solve linear semi-infinite programming problems

by combining the concepts of the "adding constraint method" for solving linear semi-infinite programming problems and the "perturbation method" for solving standard linear programming problems. This method is to find some suitable approximate solutions for the finite linear programming problems in the "adding constraint" method and the corresponding approximate solutions also form a path that approaches the optimal solution of the linear semi-infinite programming problems. Moreover, a proof for the sequence of the approximate solutions converging to the optimal solution of the linear semi-infinite programming problem is also provided in Chapter III to show that this method is reasonable in theory. Some numerical examples are given in Chapter IV to demonstrate this method. From the numerical data, it can be shown that the method suggests a very efficient way to solve linear semi-infinite programming problems.

TABLE OF CONTENTS

CHAPTER PAGE

CHAPTER PAGE

CHAPTER PAGE

LIST OF TABLES

TABLE PAGE

CHAPTER I

THE ADDING CONSTRAINT METHOD FOR SOLVING LINEAR SEMI-INFINITE PROGRAMMING PROBLEMS AND ITS APPLICATIONS

1.1 Statement of the Problem and Basic Definitions

Consider the following linear semi-infinite programming problem $(LSIP)$ which has n non-negative variables and an infinite number of inequality constraints:

$$(LSIP) \qquad \begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{n} c_j x_j. \\ &\text{Subject to} \quad \sum_{j=1}^{n} x_j a_j(t) \geq b(t), \forall t \in T, \qquad (1.1) \\ &\qquad\qquad x_j \geq 0, j = 1, \cdots, n, \qquad (1.2) \end{aligned}$$

where T is a compact metric space with an infinite cardinality, $a_j(t)(j = 1, \cdots, n)$ and $b(t)$ are real valued continuous functions defined on T.

To illustrate, consider the following problem :

$$\begin{aligned} &\textit{Minimize} \quad x_1 + \frac{x_2}{2}. \\ &\text{Subject to} \quad x_1 + t x_2 \geq \sin t, \forall t \in [0, 1], \qquad (1.3) \\ &\qquad\qquad x_1, x_2 \geq 0, \qquad (1.4) \end{aligned}$$

which has 2 non-negative variables and an infinite number of inequality constraints. $T = [0, 1]$ is a compact set with an infinite cardinality, $a_1(t) = 1, a_2(t) = t$, and $b(t) = \sin t$ are real valued continuous function defined on $[0, 1]$. It is an example of linear semi-infinite programming problems.

1.2 The Adding Constraint Method

Some basic concepts and methods for solving $LSIP$ can be found in [1,6,8,9]. One direct way of solving $LSIP$ is to discretize T into a finite number of grid points $\{t_1, t_2, \cdots, t_m\}$ and then an approximation can be obtained by solving a regular linear programming problem with n non-negative variables and m inequality constraints obtained by evaluating (1.1) at $\{t_1, t_2, \cdots, t_m\}$. Usually, a case with more grid points results in better approximation at the cost of solving a larger size linear program. Also, the choice of grid points has to be taken into consideration of the behavior of the functions $a_j(t)$, $j = 1, 2, \cdots, n$ and $b(t)$.

To avoid having too many constraints at one time, an "adding constraint" (or "cutting plane") method is introduced in [6,8,7]. Basically, it constructs a sequence of finite linear programming problems whose solutions converge to an optimal solution of $LSIP$. Before describing this method, the following problem LP_k, which is a linear programming problem with k explicit constraints can be defined as follows :

$$(LP_k) \qquad \begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{n} c_j x_j. \\ &\text{Subject to} \quad \sum_{j=1}^{n} x_j a_j(t_i) \geq b(t_i), i = 1, \cdots, k, \qquad (1.5) \\ &\qquad\qquad x_j \geq 0, j = 1, \cdots, n. \qquad (1.6) \end{aligned}$$

The "adding constraint" method works according to the following scheme :

Algorithm 1.2.1

Step 1. *Set* $k = 1$, *choose any* $t_1 \in T$ *and set* $T_1 = \{t_1\}$.

Step 2. *Solve* LP_k *with an optimal solution* $x(k)$.

Step 3. *Find a minimizer* t_{k+1} *of* $G_k(t)$ *over* T *and calculate* $G_k(t_{k+1})$, *where*

$$G_k(t) = \sum_{j=1}^{n} x_j(k)a_j(t) - b(t).$$

Step 4. *If* $G_k(t_{k+1}) \geq 0$ *then* **Stop!** $x(k)$ *is the optimal solution of LSIP.*

Otherwise, set $T_{k+1} = T_k \cup \{t_{k+1}\}$, *increment* $k \longleftarrow k+1$ *and go to Step 2.*

Note that the feasible domain of LP_k contains the feasible domain of $LSIP$, the minimum value of $LSIP$ must not be less than the minimum value of LP_k. Moreover, $x(k)$ is the optimal value of LP_k, and if

$$\sum_{j=1}^{n} x_j(k)a_j(t_{k+1}) - b(t_{k+1}) \geq 0,$$

then $x(k)$ also belongs to the feasible domain of $LSIP$. $x(k)$ must be an optimal solution of $LSIP$. Therefore, if

$$\sum_{j=1}^{n} x_j(k)a_j(t_{k+1}) - b(t_{k+1}) \geq 0,$$

then the iterations can be stopped and $x(k)$ is an optimal solution of $LSIP$. Otherwise, set $T_{k+1} = T_k \sqcup \{t_{k+1}\}$ to continue this iterative process.

Example 1.2.2 *Solve the following linear semi-infinite programming problem by using the "adding constraint method".*

$$(LSIP) \qquad \begin{aligned} &\textit{Minimize} && x_1. \\ &\textit{Subject to} && x_1 \geq \sin t, \forall t \in [0,1], && (1.7) \\ & && x_1 \geq 0. && (1.8) \end{aligned}$$

Solution: This problem can be solved by the following steps which are based on Algorithm 1.2.1.

Step 1. Set $k = 1$, choose $t_1 = 0 \in [0,1]$ and set $T_1 = \{t_1\} = \{0\}$.

Step 2. Solve LP_1 which is defined as follows :

$$(LP_1) \qquad \begin{aligned} &\textit{Minimize} && x_1. \\ &\text{Subject to} && x_1 \geq \sin t_1 = \sin 0 = 0, && (1.9) \\ & && x_1 \geq 0. && (1.10) \end{aligned}$$

Since both of the two constraints (1.9) and (1.10) are $x_1 \geq 0$, the minimum value of x_1 is 0. This implies the optimal solution $x(1) = 0$.

Step 3. Find a minimizer t_2 of $G_1(t)$ over $[0,1]$ and calculate $G_1(t_2)$, where

$G_1(t) = x(1) - \sin t = 0 - \sin t = -\sin t$.

$-\sin t$ is a decreasing function on $[0,1]$, so the minimizer t_2 of $G_1(t)$ is 1 and $G_1(t_2) = G_1(1) = -\sin 1$.

Step 4. Since $G_1(t_2) = -\sin 1 < 0$, set $T_2 = \{ t_1, t_2 \} = \{ 0, 1 \}$ and then go back to step 2 to solve LP_2.

Step 2. Set $k = 2$ and solve LP_2 which is defined as follows:

$$(LP_2) \qquad \begin{aligned} &\textit{Minimize} && x_1. \\ &\text{Subject to} && x_1 \geq \sin t_1 = 0, && (1.11) \\ &&& x_1 \geq \sin t_2 = \sin 1, && (1.12) \\ &&& x_1 \geq 0. && (1.13) \end{aligned}$$

The feasible domain of $LP_k = \{x_1 : x_1 \geq \sin 1\}$ is formed by the constraints (1.11), (1.12) and (1.13), so the the minimum value of x_1 is $\sin 1$. This implies the optimal solution $x(2)$ of LP_2 is $\sin 1$.

Step 3. Find the minimizer t_3 of $G_2(t) = x(2) - \sin t = \sin 1 - \sin t$ on $[0, 1]$ and calculate $G_2(t_3)$.

$G_2(t)$ is a decreasing function on $[0, 1]$, so the minimizer t_3 of $G_2(t)$ is 1 and $G_2(t_3) = \sin 1 - \sin t_3 = \sin 1 - \sin 1 = 0$.

Step 4. Since $G_2(t_3) \geq 0$, stop here and conclude that $x(2) = \sin 1$ is the optimal solution of LSIP.

The proof of $\{x(1), x(2), \cdots, x(k), \cdots\}$ converging to an optimal solution of $LSIP$ can be found in [6]. A refined version which allows to drop some redundant points in T_k is referred to [10].

1.3 The Application of the Adding Constraint Method for Solving Linear Semi-Infinite Programming Problems

With recent advances in the perturbation method for solving linear programming problems which will be introduced in Chapter II, a generalized barrier perturbation problem LP_{μ_k}:

$$(LP_{\mu_k}) \qquad \begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{n} c_j x_j + \mu_k \phi_k(x). \\ &\text{Subject to} \quad \sum_{j=1}^{n} x_j a_j(t_i) \geq b(t_i), i = 1, \cdots, k, \qquad (1.14) \\ &\qquad\qquad x_j \geq 0, j = 1, \cdots, n, \qquad (1.15) \end{aligned}$$

is defined to replace LP_k. $\mu_k > 0$ is a barrier perturbation parameter and $\phi_k(x)$ is a $GBLP$ function for solving LP_k. (Note that $GBLP$ function will be defined in Chapter II.) Under some appropriate assumptions, the later chapter will show that the sequence $\{x(\mu_k) : x(\mu_k)$ is the approximate solution of LP_{μ_k}, for $k = 1, 2, \cdots\}$ converges to an optimal solution of $LSIP$ by carefully choosing T_k and μ_k.

Note that LP_{μ_k} and LP_k share the same feasible domain. When the domain is a bounded set in R^n, LP_{μ_k} becomes LP_k as μ_k approaches to zero [2,3,5]. This means LP_{μ_k} can be viewed as a "relaxation" of LP_k and $x(\mu_k)$ is an "approximate" optimal solution of LP_k. That the sequence $\{x(\mu_k) : x(\mu_k)$ is the approximate solution of LP_{μ_k}, for $k = 1, 2, \cdots\}$ converges to an optimal solution of $LSIP$ will be shown in Chapter III.

CHAPTER II

THE PERTURBATION METHOD FOR SOLVING LINEAR PROGRAMMING PROBLEMS

This chapter will introduce a perturbation method to solve regular linear programming problems. Some notations, assumptions, and preliminaries are described in Section 2.1. An " ideal interior trajectory " is characterized in Section 2.2. This trajectory is well-defined, continuous, and leads to the optimum when the barrier parameter approaches zero. A good moving direction is assembled and an appropriate step size to stay in a close neighborhood of the ideal interior trajectory is found in Section 2.3. The concept of scaling matrix is adopted in Section 2.4 to determine the step size for a general perturbation algorithm. A measure is defined to be a criterion of closeness to the " ideal trajectory " in Section 2.5. A general perturbation algorithm for solving linear programming problems is proposed in Section 2.6. The sufficient condition is provided in Section 2.7 for the proof that Algorithm 2.6.1 generates a sequence which converges to the optimal solution of the linear programming problem.

Based on this concepts of "the perturbation" method for solving linear programming problems and the "adding constraint" method for solving linear semi-infinite programming problems, the "perturbation method" for solving lin-

ear semi-infinite programming problems will be developed in Chapter III.

2.1 Notations, Assumption and Preliminaries

Consider the linear programming problem P in its primal standard form

$$(P) \qquad \begin{aligned} &\textit{Minimize} \quad c^T x. \\ &\text{Subject to} \quad Ax = b, \qquad (2.1) \\ &\qquad\qquad x \geq 0, \qquad (2.2) \end{aligned}$$

where A is an $m \times n$ matrix, c and x are vectors of length n, and b is a vector of length m. Throughout this paper, the following notations and definitions will be used.

Definition 2.1.1 *If x denotes a vector $(x_1, x_2, \cdots, x_n)^T$, then the following notations are defined :*

1. *The corresponding capital letter X denotes the $n \times n$ diagonal matrix with the components of x on the diagonal, this means*

$$X = diag(x_1, x_2, \cdots, x_n) = \begin{bmatrix} x_1 & 0 & \cdots & 0 & 0 \\ 0 & x_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \vdots & x_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & x_n \end{bmatrix}.$$

2. *$x > 0$ if and only if $x_j > 0$, for $j = 1, \cdots, n$.*

3. *$x \geq 0$ if and only if $x_j \geq 0$, for $j = 1, \cdots, n$.*

4. *$x \neq 0$ if and only if there exists $j \in \{1, \cdots, n\}$ such that $x_j \neq 0$.*

5. $x = 0$ *if and only if* $x_j = 0$, *for* $j = 1, \cdots, n$.

Definition 2.1.2 *If* x *is a* n *dimensional vector* $(x_1, x_2, \cdots, x_n)^T$ *and* A *is a* $m \times n$ *matrix, then the following notations are defined :*

1.

$$(\| x \|_2 = \sum_{j=1}^{n} |x_j|^2)^{\frac{1}{2}} \text{ is called the vector two norm of } x.$$

2.

$$\| x \|_\infty = \max_j |x_j| \text{ is called the vector sup norm of } x.$$

3.

$$\| A \| = Sup_{x \neq 0} \frac{\| Ax \|}{\| x \|} \text{ is called the norm of } A. \text{ Particularly,}$$

$$\| A \|_2 = Sup_{x \neq 0} \frac{\| Ax \|_2}{\| x \|_2} \text{ is called the two norm of } A.$$

Definiton 2.1.3 (The gradient of a vector)

Let S *be a non-empty set in* R^n, *and let* $f : S \longmapsto R$, *then* f *is said to be differentiable at* $\bar{x} \in int\ S$ *if there exists a vector* $\nabla f(\bar{x})$, *called the gradient vector, and a function* $\alpha : R^n \longmapsto R$ *such that*

$$f(x) = f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) + \| x - \bar{x} \|_2 \, \alpha(\bar{x}; x - \bar{x}), \text{ for each } x \in S,$$

where $lim_{x \mapsto \bar{x}} \alpha(\bar{x}; x - \bar{x}) = 0$. *Note that if* f *is differentiable at* $\bar{x}$, *then there could only be one gradient vector, and this vector is given by* $\nabla f(\bar{x}) = (\frac{\partial f(\bar{x})}{\partial x_1}, \cdots, \frac{\partial f(\bar{x})}{\partial x_n})^T$, *where* $\frac{\partial f(\bar{x})}{\partial x_i}$ *is the partial derivative of* f *with respect to* x_i *at* $\bar{x}$.

Definition 2.1.4 $\phi(x)$ *is called a Generalized Barrier Function for linear programming problems* $(GBLP)$, *if it satisfies the following conditions :*

1. *$\phi(x) : W \mapsto \bar{R}$ is proper, strictly convex and differentiable, where $\bar{R} = R \cup \{\infty\} \cup \{-\infty\}$ is the extended real line.*

2. *If $\{x^k\} \subset W_0$ is an infinite sequence converging to x with $x_i = 0$ for at least one i, then*

$$\lim_{k \to \infty} (\nabla \phi(x^k))_i = -\infty.$$

3. *The effective domain of $\phi(x)$ contains W_0. Namely, $\phi(x)$ takes finite value at least for $x \in W_0$.*

Several simple observations can be made here :

1. Since $\phi(x)$ is strictly convex and differentiable, it must be continuously differentiable on W. In other words, $\nabla \phi(x)$ is continuous on W.

2. $\phi(x)$ is finite on W_0 and may or may not assume the value of $+\infty$ when $x_i = 0$ for some i.

3. Any positive linear combination of $GBLP$ functions is still a $GBLP$ function. So, mixed barrier functions may be applied for linear programming problems.

4. The functions

$$-\sum_{i=1}^{n} \ln x_i, \ \sum_{i=1}^{n} x_i \ln x_i, \ \sum_{i=1}^{n} \frac{1}{x_i^r}, \text{ for } r > 0,$$

are all $GBLP$ functions.

Note that the smoothness conditions are imposed in the definition because functions which are not differentiable or one point discontinuous on the boundary must be ruled out.

Definition 2.1.5 *Define an augmented primal problem P_μ associated with a generalized barrier function $\phi(x)$ as follows:*

$$(P_\mu) \qquad \begin{aligned} &\textit{Minimize} && c^T x + \mu\phi(x). \\ &\textit{Subject to} && Ax = b, \qquad (2.3) \\ & && x \geq 0, \qquad (2.4) \end{aligned}$$

where $\phi(x)$ is the GBLP function.

This paper considers the standard linear programming problem P which satisfies the following assumptions :

Assumption 2.1.6

1. *The interior feasible domain $W_0 = \{x \in R^n : Ax = b, x > 0\}$ is nonempty.*
2. *The feasible domain $W = \{x \in R^n : Ax = b, x \geq 0\}$ is bounded.*
3. *A has full row rank. In other words, rank of A = row rank of $A = m$.*
4. *P has a unique optimal solution.*

2.2 The Ideal Interior Trajectory

For any given $\mu > 0$, problem P_μ has a unique optimal solution on W_0. Moreover, when μ approaches zero, the curve formed by all the minimizers is continuous in μ and converges to the unique optimal solution x^* of P. These properties can be shown by the following Theorems.

Theorem 2.2.1 *For $\mu > 0$ problem P_μ has a unique optimal solution $x^*(\mu) \in W_0$.*

Proof : Since $\phi(x)$ is strictly convex on W, $c^T x + \mu\phi(x)$ is strictly convex too. Moreover, the feasible domain W is convex and bounded. This assures that P_μ has a unique optimal solution $x^*(\mu)$ on W. Furthermore, this solution has to satisfy the following $Karush - Kuhn - Tucker$ $(K - K - T)$ conditions :

$$\begin{cases} c + \mu \bigtriangledown \phi(x) - A^T y - \lambda = 0, \\ Ax = b, \\ x^T \lambda = 0, \\ x \geq 0, \lambda \geq 0, \end{cases} \tag{2.5}$$

where $y \in R^m$ and $\lambda \in R^n$ are Lagrangian vectors.

Note that the Karush-Kuhn-Tucker $(K - K - T)$ conditions is introduced in Appendix A.

From the definition of $GBLP$ function, $(\bigtriangledown\phi(x))_i$ approaches negative infinity if x_i is zero for some i. Since μ is positive, if x_i is zero for some i, $\mu(\bigtriangledown\phi(x))_i$ becomes minus infinity and hence (2.5) could not have any finite solution. Therefore, $x > 0$. This implies that $x(\mu)$ has to lie in W_o. Moreover, $x > 0$, $\lambda \geq 0$ and $x^T\lambda = 0$, so $\lambda = 0$. (2.5) can be reduced to the following system :

$$\begin{cases} c + \mu \bigtriangledown \phi(x) - A^T y = 0, \\ Ax = b, \\ x > 0. \end{cases} \tag{2.6}$$

Theorem 2.2.2 $\{x^*(\mu) : \mu > 0\}$ *characterizes an interior, continuous curve in* W_0.

Proof : That $\{x^*(\mu) : \mu > 0\}$ characterizes an interior curve in W_0 has already been shown in Theorem 2.1.2, the continuity part will be proved as follows :

Suppose the curve is discontinuous at $\bar{\mu} > 0$. Then, there exists $\epsilon > 0$ such that no matter how small $\triangle\mu$ is,

$$\| x^*(\bar{\mu} + \triangle\mu) - x^*(\bar{\mu}) \|_2 \geq 2\epsilon > 0.$$

This will lead to a contradiction that $x^*(\bar{\mu} + \triangle\mu)$ is not the minimum of problem $P_{\bar{\mu}+\triangle\mu}$. As it was mentioned before, $c^T x + \bar{\mu}\phi(x)$ is strictly convex. Since $x^*(\bar{\mu})$ is the minimum of $P_{\bar{\mu}}$, for $x \in W_0$ and $\| x^*(\bar{\mu}) - x \|_2 > \epsilon$, there exists a number $\tau > 0$ such that

$$c^T x + \bar{\mu}\phi(x) > c^T x^*(\bar{\mu}) + \bar{\mu}\phi(x^*(\bar{\mu})) + \tau.$$

In particular,

$$c^T x^*(\bar{\mu} + \triangle\mu) + \bar{\mu}\phi(x^*(\bar{\mu} + \triangle\mu)) > c^T x^*(\bar{\mu}) + \bar{\mu}\phi(x^*(\bar{\mu})) + \tau \tag{2.7}$$

no matter how small $\triangle\mu$ is.

(Case 1) :

Suppose that the effective domain of $\phi(x)$ is W. Since W is compact and $\phi(x)$ is continuous on W, $\phi(x)$ is bounded. Choose $\triangle\mu$ small enough such that

$$0 < | \triangle\mu | \| \phi(x) \|_\infty < \frac{\tau}{2}, \tag{2.8}$$

where $\| \cdot \|_\infty$ is the sup norm. Therefore

$$c^T x^*(\bar{\mu}) + (\bar{\mu} + \triangle\mu)\phi(x^*(\bar{\mu}))$$

$$= c^T x^*(\bar{\mu}) + \bar{\mu}\phi(x^*(\bar{\mu})) + \triangle\mu\phi(x^*(\bar{\mu}))$$

$$< c^T x^*(\bar{\mu}) + \bar{\mu}\phi(x^*(\bar{\mu})) + \frac{\tau}{2} \cdots\cdots\cdots\cdots\cdots\cdots \text{by(2.8)}$$

$$< c^T x^*(\bar{\mu} + \Delta\mu) + \bar{\mu}\phi(x^*(\bar{\mu} + \Delta\mu)) - \frac{\tau}{2} \cdots\cdots\cdots\cdots\cdots \text{by(2.7)}$$

$$< c^T x^*(\bar{\mu} + \Delta\mu) + (\bar{\mu} + \Delta\mu)\phi(x^*(\bar{\mu} + \Delta\mu)) \cdots\cdots\cdots\cdots \text{by(2.8)}$$

which shows $x^*(\bar{\mu} + \Delta\mu)$ is not the minimum of $P_{\bar{\mu}+\Delta\mu}$.

(Case 2) :

On the other hand, if $\phi(x)$ takes the value of $+\infty$ on the boundary such that $\phi(x^*(\bar{\mu} + \Delta\mu))$ could possibly be unbounded above for small Δ, then

$$c^T x^*(\bar{\mu}) + (\bar{\mu} + \Delta\mu)\phi(x^*(\bar{\mu})) < c^T x^*(\bar{\mu} + \Delta\mu) + (\bar{\mu} + \Delta\mu)\phi(x^*(\bar{\mu} + \Delta\mu)) \quad (2.9)$$

is automatically true since the right hand side of (2.9) is unbounded above. Otherwise, if $\phi(x^*(\bar{\mu} + \Delta\mu))$ is still bounded, then the same argument will follow by applying (2.8).

Theorem 2.2.3 *Given a decreasing positive sequence* $\{\mu_k\}$ *such that*

$$\lim_{k\to\infty} \mu_k = 0, \text{ if } \lim_{k\to\infty} x^*(\mu_k) = x^*, \text{ then } x^* \text{ is the minimum of } P.$$

This indicates that $x^*(\mu)$ *is indeed continuous on* $\mu \geq 0$.

Proof :

(Case 1) :

Suppose that the effective domain of $\phi(x)$ is W, then for any $x \in W$ and $k = 1, 2, \cdots,$

$$c^T x^*(\mu_k) + \mu_k \phi(x^*(\mu_k)) \leq c^T x + \mu_k \phi(x). \quad (2.10)$$

Since W is compact, x^* must be in W. Taking limit on both sides of (2.10) and considering the continuity of ϕ can get the result that $c^T x^* \leq c^T x$.

Moreover, $x(\mu_k) > 0$ and $Ax(\mu_k) = b$; therefore,

$$\lim_{k \mapsto \infty} x(\mu_k) = x^* \geq 0 \text{ and } Ax^* = b.$$

Hence x^* is the unique optimal solution of P.

(Case 2) :

If $\phi(x)$ takes the value of $+\infty$ on the boundary, it will lead to a contradiction. Assume that x^* is not the minimum of P, but v^* is. In this case, there exists $\mu \in W_0$ satisfying $c^T v^* < c^T \mu < c^T x^*$. Since

$$\lim_{k \mapsto \infty} \mu_k = 0,$$

there exists k large enough and $\delta > 0$ such that $c^T \mu + \mu_k \phi(\mu) < c^T x^* - \delta$.

In case $x^* \in W_0$, since it is the limit of $\{x^*(\mu_k)\}$, when k is large enough $x^*(\mu_k)$ will be in a small neighborhood of x^* on which $\phi(x^*(\mu_k))$ is bounded and takes finite values. Therefore, $c^T x^* - \delta < c^T x^*(\mu_k) + \mu_k \phi(x^*(\mu_k))$ for large enough k. But this contradicts to the fact that $x^*(\mu_k)$ is the minimum of P_{μ_k}.

On the other hand, if $x^* \in W \backslash W_0$ and $\phi(x)$ takes the value $+\infty$ at x^*, $\mu_k \phi(x^*(\mu_k))$ might not have a limit. However,

$$\liminf_{k \mapsto \infty} \{\mu_k \phi(x^*(\mu_k))\} \geq 0.$$

Therefore, for large enough k,

$$\begin{aligned} c^T x^* - \frac{\delta}{2} &< c^T x^*(\mu_k) + \liminf_{k \mapsto \infty} \{\mu_k \phi(x^*(\mu_k))\} \\ &< c^T x^*(\mu_k) + \mu_k \phi(x^*(\mu_k)) + \frac{\delta}{2}. \end{aligned}$$

Consequently, $c^T \mu + \mu_k \phi(\mu) < c^T x^* - \delta < c^T x^*(\mu_k) + \mu_k \phi(x^*(\mu_k))$ which again is a contradiction. Therefore, x^* must be the optimal solution of P.

Theorem 2.2.4 *For any positive decreasing sequence* $\{\mu_k\}$ *with*

$$\lim_{k \mapsto \infty} \mu_k = 0,$$

$\{x^*(\mu_k)\}$ *must converge. Moreover,* $\{x^*(\mu_k)\}$ *converges to the optimal solution of* P.

Proof : Since W is compact, the sequence $\{x^*(\mu_k)\}$ must have at least one limit point. For each limit point, there exists a subsequence which actually converges to it. Each limit point is indeed an optimal solution of P can be concluded by applying Theorem 2.2.3. P is assumed to have a unique optimal solution; therefore, $\{x^*(\mu_k)\}$ has one and only one limit point and thus converges to it.

Note that the strictly convex condition of the $GBLP$ function is necessary. Without this condition, there does exist an interior trajectory $\{x^*(\mu) : \mu > 0\}$ which is discontinuous at some μ.

2.3 Moving Directions

Moving direction is the soul of an iterative algorithm. Once the trajectory is defined by a $GBLP$ function, a moving direction is needed to sail along the trajectory for optimum. Theorem 2.3.1 defines an extended formula for moving directions to a more general setting.

Theorem 2.3.1 *Let* $\phi(x)$ *be a GBLP function and problem* P_μ *be defined as before :*

$$(P_\mu) \qquad \begin{aligned} \textit{Minimize} \quad & c^T x + \mu\phi(x). \\ \textit{Subject to} \quad & Ax = b, \\ & x \geq 0. \end{aligned}$$

Suppose that $\bar{x}$ is a given interior feasible solution of P_μ, then Δx is an optimal solution of P'_s, and Δx defines a moving direction at $\bar{x}$ as

$$\Delta x = -\tilde{X}[I - \tilde{X}A^T(A\tilde{X}^2A^T)^{-1}A\tilde{X}]\tilde{X}(c + \mu \bigtriangledown \phi(\bar{x})).$$

P'_s is defined as follows :

$$(P'_s) \qquad \begin{aligned} &\textit{Minimize} \quad (c + \mu \bigtriangledown \phi(\bar{x}))^T \Delta x. \\ &\textit{Subject to} \quad A\Delta x = 0, \\ &\qquad\qquad \| \tilde{X}^{-1}\Delta x \|_2^2 \leq \beta^2 < 1, \end{aligned}$$

where $\tilde{X}$ is an any positive definite symmetric matrix.

Proof : Since A has a full row rank, the null space of A is an $n - m$ dimensional subspace of R^n, which is isomorphic to R^{n-m}. Let U be an isomorphism between the null space of A and R^{n-m} such that each Δx in the null space of A can be represented in terms of h in R^{n-m}, i.e., $\Delta x = Uh$. In this way, the homogeneous constraint $A\Delta x = 0$ can be eliminated and P'_s can be converted to a problem minimizing over $h \in R^{n-m}$, i.e.,

$$\begin{aligned} &\textit{Minimize} \quad (c + \mu \bigtriangledown \phi(\bar{x}))^T Uh. \\ &\text{Subject to} \quad \| \tilde{X}^{-1}Uh \|_2^2 \leq \beta^2 < 1, \\ &\qquad\qquad h \in R^{n-m}, \end{aligned}$$

where $\Delta x = Uh$.

The associated Lagrangian becomes

$$L(h, \lambda) = (c + \mu \bigtriangledown \phi(\bar{x}))^T Uh + \lambda(\| \tilde{X}^{-1}Uh \|_2^2 - \beta^2)$$

with an associated Lagrange multiplier $\lambda \geq 0$.

$$h = \frac{-1}{2\lambda}(U^T \tilde{X}^{-2} U^{-1})^{-1} U^T (c + \mu \bigtriangledown \phi(\bar{x})),$$

and $\triangle x = Uh = \frac{-1}{2\lambda} U(U^T \tilde{X}^{-2} U^{-1})^{-1} U^T (c + \mu \bigtriangledown \phi(\bar{x}))$ can be obtained by taking $\frac{\partial L}{\partial h} = 0$ and solving for h.

By Lemma 2 of [4], the following identity

$$\tilde{X}[I - \tilde{X}A^T(A\tilde{X}^2A^T)^{-1}A\tilde{X}]\tilde{X}x = U(U^T\tilde{X}^{-2}U^{-1})^{-1}U^Tx$$

holds for all $x \in R^n$; therefore,

$$\triangle x = -\tilde{X}[I - \tilde{X}A^T(A\tilde{X}^2A^T)^{-1}A\tilde{X}]\tilde{X}(c + \mu \bigtriangledown \phi(\bar{x})), \tag{2.11}$$

where $\frac{1}{2\lambda}$ is omitted since a scale does not affect the moving direction. This completes the proof.

Corollary 2.3.2 *:(General Newton Direction for P_μ)*
Let $\phi(x)$ be a GBLP function which is twice differentiable in the feasible domain. If choose $\tilde{X}^{-2}$ to be the Hessian matrix of $c^Tx + \mu\phi(x)$ at $\bar{x}$ and denote $\tilde{X}^{-2}$ by H_ϕ^{-2}, then $\triangle x$ obtained in Theorem 2.3.1 is the Newton direction in approximating P_μ.

Proof : As in the proof of Theorem 2.2.1, the $K - K - T$ conditions of P_μ are

$$\begin{cases} c + \mu \bigtriangledown \phi(x) - A^Ty = 0, \\ Ax = b, \\ x > 0. \end{cases} \tag{2.12}$$

Define $-\mu \bigtriangledown \phi(x) = s$, then system (2.12) becomes

$$\begin{cases} A^T y + s - c = 0, \\ Ax - b = 0, \\ \mu \bigtriangledown \phi(x) = -s, \\ x > 0. \end{cases} \tag{2.13}$$

Notice that problem P_μ is a convex program with a unique optimal solution, which is completely characterized by (2.13). The first order approximation to $\mu \bigtriangledown \phi(x) + s = 0$ yields

$$-\bar{s} - \mu \bigtriangledown \phi(\bar{x}) = [H_\phi^{-2} | I_n] \begin{pmatrix} x - \bar{x} \\ s - \bar{s} \end{pmatrix},$$

where $H_\phi^{-2} = \mu \bigtriangledown^2 \phi(\bar{x})$.

$$s = \gamma - H_\phi^{-2} x \tag{2.14}$$

can be obtained by simplifying the above equation, where $\gamma = -\mu \bigtriangledown \phi(\bar{x}) + H_\phi^{-2} \bar{x}$.

$$x = H_\phi^2(-c + \gamma + A^T y) \tag{2.15}$$

can be resulted by replacing s in (2.14) with $c - A^T y$ and representing x in terms of y.

$$y = (AH_\phi^2 A^T)^{-1}(b + AH_\phi^2 c - AH_\phi^2 \gamma) \tag{2.16}$$

can be obtained by applying A on both sides of (2.15). A moving direction

$$\triangle x = -H_\phi [I - H_\phi A^T (AH_\phi^2 A^T)^{-1} AH_\phi] H_\phi (c + \mu \bigtriangledown \phi(\bar{x})) \tag{2.17}$$

can be obtained by plugging (2.16) into (2.15). Hence, $\triangle x$ obtained in Theorem 2.3.1 is the Newton direction in approximating P_μ.

Theorem 2.3.3 *Let $\phi(x)$ be a GBLP function which is twice differentiable in the feasible domain and $\tilde{X}^{-2}$ be any positive definite symmetric matrix. Suppose that $\bar{x}$ is a current interior feasible solution of P_μ and*

$$\triangle x = -\tilde{X}[I - \tilde{X}A^T(A\tilde{X}^2A^T)^{-1}A\tilde{X}]\tilde{X}(c + \mu \bigtriangledown \phi(\bar{x})),$$

then if $\| \triangle x \|_2$ is sufficiently small, $\tilde{x} = \bar{x} + \triangle x$ minimizes the following quadratic function QP over W_0.

$$(QP) : F(x) = [c^T\bar{x} + \mu\phi(\bar{x})] + [c + \mu\bigtriangledown\phi(\bar{x})]^T(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T\tilde{X}^{-2}(x - \bar{x}). \quad (2.18)$$

Proof : Since $A\triangle x = 0$, $A\tilde{x} = A\bar{x} + A\triangle x = b$. If $\| \triangle x \|_2$ is small enough such that $\tilde{x}$ is positive, then $\tilde{x}$ is in W_0. Take Taylor's expansion of $F(x)$ at $\tilde{x}$ to prove that $\tilde{x}$ really attains minimum,

$$F(x) = F(\tilde{x}) + [(c + \mu\bigtriangledown\phi(\bar{x})) + \tilde{X}^{-2}(\tilde{x} - x)]^T(x - \tilde{x}) + \frac{1}{2}(x - \tilde{x})^T\tilde{X}^{-2}(x - \tilde{x}). \quad (2.19)$$

If plug $\triangle x = -\tilde{X}[I - \tilde{X}A^T(A\tilde{X}^2A^T)^{-1}A\tilde{X}]\tilde{X}(c + \mu\bigtriangledown\phi(\bar{x}))$ into (2.19), the second term on the right hand side becomes

$$(c + \mu \bigtriangledown \phi(\bar{x})) - \tilde{X}^{-1}[I - \tilde{X}A^T(A\tilde{X}^2A^T)^{-1}A\tilde{X}]\tilde{X}(c + \mu \bigtriangledown \phi(\bar{x}))$$

$$= A^T(A\tilde{X}^2A^T)^{-1}A\tilde{X}^2(c + \mu \bigtriangledown \phi(\bar{x})).$$

Therefore, for any feasible point x,

$$[(c + \mu \bigtriangledown \phi(\bar{x})) + \tilde{X}^{-2}\triangle x]^T(x - \tilde{x})$$

$$= [(c + \mu \bigtriangledown \phi(\bar{x}))^T\tilde{X}^2A^T(A\tilde{X}^2A^T)^{-1}A](x - \tilde{x}) = 0$$

Furthermore, $\tilde{X}^{-2}$ is positive definite, hence (2.19) becomes

$$F(x) = F(\tilde{x}) + \frac{1}{2}(x - \tilde{x})^T\tilde{X}^{-2}(x - \tilde{x}) \geq F(\tilde{x})$$

for any feasible point x in W_0. Thus, $\tilde{x}$ is the optimal solution.

Even though the quadratic approximation is good and the moving direction guarantees that one step leads to the optimal, the whole procedure is still useless. "How to choose an appropriate step size" will be considered in the next section.

In [3], Fang and Sheu constructed a primal-dual algorithm provided that a triplet $(\bar{x}, \bar{y}, \bar{s})$ is satisfying $A\bar{x} = b, A^T\bar{y} + \bar{s} = c$, where $\bar{x} > 0, \bar{s} > 0$ are given. Starting from $(\bar{x}, \bar{y}, \bar{s})$ the primal search direction not only generates a new primal feasible solution but also a dual estimate $y = (A\tilde{X}^2A^T)^{-1}A\tilde{X}^2(c + \mu \bigtriangledown \phi(\bar{x}))$. Moreover, the moving direction for dual variables is obtained as : $\triangle y = (A\tilde{X}^2A^T)^{-1}A\tilde{X}^2(\bar{s} + \mu \bigtriangledown \phi(\bar{x}))$ in [3].

2.4 Scaling and Step Size

One commonly used technique for determining a step size is to scale the problem into a transformed space. Let $\phi(x)$ be a $GBLP$ function. Consider the problem P_μ and the transformed problem $P_{\mu'}$ under the scaling matrix $\bar{X}$ formed by a current interior solution $\bar{x}$ as follows :

$$(P_\mu) \qquad \begin{aligned} &\textit{Minimize} && c^Tx + \mu\phi(x). \\ &\text{Subject to} && Ax = b, \\ & && x \geq 0, \end{aligned}$$

$$(P_{\mu'}) \qquad \begin{aligned} &\textit{Minimize} && c^T\bar{X}y + \mu\phi(\bar{X}y). \\ &\text{Subject to} && A\bar{X}y = b, \\ & && y \geq 0. \end{aligned}$$

Theorem 2.4.1 *The problems P_μ and $P_{\mu'}$ are equivalent. Moreover, the moving direction obtained from the negative projected gradient at $\bar{x}$ in P_μ is the same as that obtained at $e = (1, 1, \cdots, 1)^T$ in $P_{\mu'}$ under the transformation $x = \bar{X}y$.*

Proof : Define $T : \{x | Ax = b, x > 0\} \longrightarrow \{y | A\bar{X}y = b, y > 0\}$ such that $y = T(x) = \bar{X}^{-1}x$.

It is easy to check that if x is a feasible solution of P_μ, then $y = \bar{X}^{-1}x$ is feasible to $P_{\mu'}$ and the corresponding objective values of x in P_μ and y in $P_{\mu'}$ are equal. Furthermore, since each diagonal element of $\bar{X}$ is positive, the sign restriction is preserved under T. Therefore, the problems P_μ and $P_{\mu'}$ are equivalent.

Note that the moving direction in $P_{\mu'}$ is the orthogonal projection of negative gradient vector onto the null space of $A\bar{X}$, namely,

$$-[I - \bar{X}A^T(A\bar{X}^2A)^{-1}A\bar{X}](\frac{1}{\mu}\bar{X}c + \bar{X} \bigtriangledown \phi(\bar{X}y)). \tag{2.20}$$

Since $\bar{x}$ is mapped into $e = (1, 1, \cdots, 1)^T$ in the transformed space, the moving direction at $\bar{x}$ in P_μ should start at e in $P_{\mu'}$.

$$-[I - \bar{X}A^T(A\bar{X}^2A)^{-1}A\bar{X}](\frac{1}{\mu}\bar{X}c + \bar{X} \bigtriangledown \phi(\bar{x}))$$

which is the moving direction at e in the transformed space can be obtained by plugging in $y = e$ into (2.20).

$$\triangle x = -\bar{X}[I - \bar{X}A^T(A\bar{X}^2A)^{-1}A\bar{X}](\frac{1}{\mu}\bar{X}c + \bar{X} \bigtriangledown \phi(\bar{x}))$$

can be obtained by mapping this moving direction back to P_μ with the inverse transformation of T. It is exactly the same as the moving direction obtained di-

rectly from the "orthogonal" projection in the original space P_μ with a positive definite symmetric matrix $\bar{X}^{-1}$.

Observe that using the Euclidean norm in the transformed space is actually the same as using the inner product norm defined by the positive definite matrix $\bar{X}^{-1}$ in the original space. More specifically, this means

$$\| y \|_2 = \| \bar{X}^{-1} x \|_2 = \{ < x, x >_{\bar{X}^{-1}} \}^{\frac{1}{2}}$$

where $< x, x >_{\bar{X}^{-1}}$ is a weighted inner product defined by the quadratic form $x^T \bar{X}^{-2} x$. Theorem 2.4.1 therefore indicates that neither a transformed space nor a scale-invariant $GBLP$ function is necessary for deriving moving directions. They can be obtained directly in the space $\{Ax = b, x > 0\}$ with different norms.

Since the $\bar{X}^{-1}$ norm varies according to the current feasible solution $\bar{x}$, it reflects how far a current point is away from boundary. Since

$$\| y \|_2 = \| \bar{X}^{-1} x \|_2 \leq \| \bar{X}^{-1} \|_2 \times \| x \|_2 < \| x \|_2 \text{ if } \| \bar{X}^{-1} \|_2 < 1;$$

therefore, under the $\bar{X}^{-1}$ norm, a step size less than one retains the interior feasibility of a new solution. The choice of a scaling matrix affects the moving direction. According to Corollary 2.3.2, only when the scaling matrix is chosen to be the Hessian matrix of $\phi(x)$, the moving direction is a Newton direction and the quadratic model QP in Theorem 2.3.3 becomes the second order Taylor's approximation. But it may cause an ellipsoid too large to be inscribed in the feasible domain. One strategy is to use $\bar{X}^{-1}$ as the scaling matrix when the Hessian matrix produces an oversize ellipsoid.

2.5 Criterion of Closeness

Section 2.2 characterizes an ideal interior trajectory which is continuous in μ and leads to the optimal solution of P. However, it takes excessive computations to attain the true minimum $x^*(\mu)$ of P_μ for each μ. Therefore, a "criterion of closeness" is needed to tell when to reduce the barrier parameter μ. The criterion of closeness can be viewed as an n-dimensional ball around $x^*(\mu)$. Once a current solution falls inside that ball, then consider it is close enough to $x^*(\mu)$ and start another iteration with a new parameter $\mu' < \mu$.

Restate the $K-K-T$ conditions of problem P_μ as follows :

$$\begin{cases} A^T y + s - c = 0, \\ Ax - b = 0, \\ \frac{1}{\mu} s = -\nabla \phi(x), \\ x > 0. \end{cases} \tag{2.21}$$

For a primal algorithm, what it has is a current primal feasible solution with a moving direction and a dual estimate. If apply the formula for the dual estimate y in previous definition and define

$$s = c - A^T y = c - A^T (A\bar{X}^2 A^T)^{-1} A\bar{X}^2 (c + \mu \nabla \phi(\bar{x})),$$

then the vector pair (y, s) automatically satisfies the first equation of (2.21). Moreover, $\bar{x}$ satisfies the second equation for primal feasibility. Therefore, if $\frac{1}{\mu}s + \nabla\phi(\bar{x})$, or equivalently, $\| \frac{1}{\mu}s + \nabla\phi(\bar{x}) \|_2$, is small, $\bar{x}$ becomes close to $x^*(\mu)$ since the $K-K-T$ conditions(2.21) has a unique solution.

Expand $\frac{1}{\mu}s + \nabla\phi(\bar{x})$ in terms of $\bar{x}$ to get

$$\frac{1}{\mu}s + \nabla\phi(\bar{x}) = \frac{1}{\mu}c - A^T(A\bar{X}^2A^T)^{-1}A\bar{X}^2(\frac{1}{\mu}c + \nabla\phi(\bar{x})) + \nabla\phi(\bar{x})$$

$$= [I - A^T(A\bar{X}^2A^T)^{-1}A\bar{X}^2](\frac{1}{\mu}c + \nabla\phi(\bar{x}))$$

$$= \bar{X}^{-1}[I - \bar{X}A^T(A\bar{X}^2A^T)^{-1}A\bar{X}]\bar{X}(\frac{1}{\mu}c + \nabla\phi(\bar{x})),$$

which shows that $\bar{X}(\frac{1}{\mu}s+\nabla\phi(\bar{x}))$ is the moving direction in the transformed space. Therefore, the criterion function of closeness can be defined as follows :

Definition 2.5.1 *Given a primal interior feasible solution $\bar{x}$ and $\mu > 0$, the "criterion function of closeness" $d(\bar{x}, \mu)$ is defined by*

$$d(\bar{x}, \mu) = \| \bar{X}(\frac{1}{\mu}s + \nabla\phi(\bar{x})) \|_2 = \| [I - \bar{X}A^T(A\bar{X}^2A^T)^{-1}A\bar{X}]\bar{X}(\frac{1}{\mu}c + \nabla\phi(\bar{x})) \|_2, \tag{2.22}$$

where $s = c - A^Ty$ and y is the dual estimate as what was defined previously.

It can be shown that $\mathrm{d}(\bar{x}, \mu)$ is the best measure at $\bar{x}$ with least square error among all dual pairs (y, s).

Theorem 2.5.2 *$d(\bar{x}, \mu)$ is the optimal solution to the following convex programming problem :*

$$(CP)\begin{cases} \text{Minimize } \| \bar{X}(\frac{1}{\mu}s + \nabla\phi(\bar{x})) \|_2 . \\ s = c - A^Ty, \\ y \in R^m. \end{cases}$$

Proof : An equivalent unconstrained problem :

$$Minimize \parallel \bar{X}[\frac{1}{\mu}(c - A^T y) + \bigtriangledown\phi(\bar{x})] \parallel_2 .$$

can be obtained by plugging $s = c - A^T y$ into the objective function. Therefore,

$$\parallel \bar{X}[\frac{1}{\mu}(c - A^T y) + \bigtriangledown\phi(\bar{x})] \parallel_2^2$$

$$=< [\frac{1}{\mu}c^T - \frac{1}{\mu}y^T A + \bigtriangledown\phi(\bar{x})^T]\bar{X}, \bar{X}[\frac{1}{\mu}(c - A^T y) + \bigtriangledown\phi(\bar{x})] >$$

$$=\parallel \bar{X}(\frac{1}{\mu}c + \bigtriangledown\phi(\bar{x})) \parallel_2^2 + \frac{1}{\mu^2}y^T(A\bar{X}^2A^T)y - \frac{2}{\mu}y^T A\bar{X}^2[\frac{1}{\mu}(c + \bigtriangledown\phi(\bar{x}))],$$

which is a function of y.

$$y = (A\bar{X}^2A^T)^{-1}A\bar{X}^2(c + \mu \bigtriangledown \phi(\bar{x})),$$

which is the optimal solution of problem CP can be obtained by taking the derivative of $\parallel \bar{X}[\frac{1}{\mu}(c - A^T y) + \bigtriangledown\phi(\bar{x})] \parallel_2^2$ with respect to y and setting it to be zero. But y is the dual estimate induced from $(\bar{x}, \mu)$; therefore, $d(\bar{x}, \mu)$ is the minimum of CP by definition.

Theorem 2.5.3 *For any $\mu > 0$, $d(x^*(\mu), \mu) = 0$. Conversely, if $d(\bar{x}, \mu) = 0$ then $\bar{x}$ must be the optimal solution of P_μ.*

Proof : This is an immediate consequence of the above discussion. Observe that $d(\bar{x}, \mu) \geq 0$ which is the minimum value of CP depending on the given primal feasible solution $\bar{x}$ and μ. Because $x^*(\mu)$ is the optimal solution of P_μ and its associated dual pair $(y^*(\mu), x^*(\mu))$ satisfies $A^T y^*(\mu) + s^*(\mu) - c = 0$ as well as $\frac{1}{\mu}s^*(\mu) = - \bigtriangledown \phi(x^*(\mu))$, $\parallel X^*(\mu)(\frac{1}{\mu}s^*(\mu) + \bigtriangledown\phi(x^*(\mu)) \parallel_2 = 0$. Hence, it must be the minimum value of CP at $x^*(\mu)$. In other words, $d(x^*(\mu), \mu) = 0$. Conversely, since P_μ has a unique solution, if $d(\bar{x}, \mu) = 0$ then $\bar{x}$ must be the optimal solution of P_μ.

Theorem 2.5.4 *For $\bar{x} \in W_0$ and $\mu > 0$, $d(\bar{x}, \mu)$ is continuous in μ.*

Proof : Let M_n be the set of all $n \times n$ real matrices and $\| \cdot \|_2$ be the matrix two norm. Define a function $h : W_0 \longmapsto M_n$ by $h(\bar{x}) = (A\bar{X}^2A^T)^{-1}$, and show that h is continuous in W_0 as follows :

For $y \in W_0$, let $Y = diag(y_1, y_2, \cdots, y_n)$, then

$$\| h(\bar{x}) - h(y) \|_2 = \| (A\bar{X}^2A^T)^{-1} - (AY^2A^T)^{-1} \|_2$$

$$= \| (A\bar{X}^2A^T)^{-1}[AY^2A^T - A\bar{X}^2A^T](AY^2A^T)^{-1} \|_2$$

$$\leq \| (A\bar{X}^2A^T)^{-1} \|_2 \| A(Y^2 - \bar{X}^2)A^T \|_2 \| (AY^2A^T)^{-1} \|_2 .$$

When y is close enough to $\bar{x}$, $(AY^2A^T)^{-1}$ is bounded and $A(Y^2 - \bar{X}^2)A^T$ tends to zero; therefore, h is continuous in W_0. Since the composite of continuous functions is still continuous, $d(\bar{x}, \mu)$ is continuous in μ from (2.22).

Notice that the function $d(\bar{x}, \mu)$ is only an index showing how close it is between a current solution $\bar{x}$ and the real minimizer $x^*(\mu)$. It is not a distance function defined on (x, μ), since it says nothing about the distance between two arbitrary feasible points. In particular, $d(x^1, \mu) < d(x^2, \mu)$ does not imply x^1 is closer to $x^*(\mu)$ than x^2 is. Since the d–function is defined by using $\bar{X}^{-1}$-norm under which the space has been distorted according to different current points and therefore can not reflect the true geometric relations in the original space. Also, $d(\bar{x}, \mu)$ is the length of a moving direction under the measure of the scaling matrix $\bar{X}$. Hence, $d(\bar{x}, \mu)$ has to be less than one to avoid primal infeasibleness.

2.6 The Generic Barrier Function Algorithm

Having all the essential components in hand, a generic barrier function algorithm can be constructed as follows :

Algorithm 2.6.1

Step 1 (Initialization and Notations)

Let $0 < \theta, \beta < 1$ *be two constants and* L *be the input length of* P, $\mu_0 > 0$ *and* x^0 *be an initial interior feasible solution of* P *such that* $d(x^0, \mu_0) \leq \beta$.

Let $f_\mu(x) = c^T x + \mu\phi(x)$.

Set $k := 0$.

Step 2 (Check Optimum)

If $\mu_k \leq e^{-L}$ *then* **Stop!** x^k *is the solution.*

Otherwise, go to next step.

Step 3 (Compute Dual Estimate)

$y^k := (AX_k^2A^T)^{-1}AX_k^2(c + \mu \bigtriangledown \phi(x^k))$,

$s^k := c - A^T y^k$,

$r^k := X_k(\frac{1}{\mu_k}s^k + \bigtriangledown\phi(x^k))$,

where $X_k = diag(x_1^k, \cdots, x_n^k)$.

Step 4 (Check Criterion of Closeness)

$d(x^k, \mu_k) := \| r^k \|_2$,

if $d(x^k, \mu_k) \leq \beta$, *then set*

$$\mu_{k+1} = (1 - \theta)\mu_k,$$

$$x^{k+1} = x^k - X_k r^k,$$

$$k = k + 1.$$

Otherwise, go to next step.

Step 5 (Compute Another Solution Closer to $\mathbf{x}^*(\mu_k)$)

$$\delta := argmin_{0<\alpha\leq 1}\{f_{\mu_k}(x^k - \alpha X_k r^k) \mid x^k - \alpha X_k r^k > 0\},$$

$$x^k = x^k - \delta X_k r^k.$$

Go to Step 3 without updating k and μ_k.

2.7 The Condition and the Proof for the Convergence of the Algorithm

Some properties on the *GBLP* functions are imposed in Condition 2.7.1 to make sure that Algorithm 2.6.1 converges to the optimal solution of problem *P*.

Let $\phi(x)$ be a *GBLP* function and $X = diag(x_1, x_2, \cdots, x_n)$ for $x \in W$.

Condition 2.7.1 *There exists $M > 0$ such that $\| X \bigtriangledown \phi(x) \|_2 \leq M$ for each $x \in W$.*

Theorem 2.7.2 *For any $\mu_k > 0$, if there is a primal feasible solution x^k with $d(x^k, \mu_k) > \beta$ then Step 5 of Algorithm 2 will eventually generate a primal feasible solution x' such that $d(x', \mu_k) \leq \beta$.*

Proof : This is a direct consequence of the descent property of the moving direction. Recall that

$$\triangle x^k = -X_k[I - X_k A^T (A X_k^2 A^T)^{-1} A X_k] X_k (\frac{c}{\mu_k} + \bigtriangledown \phi(x^k)) = -X_k r^k.$$

Notice that $d(x^k, \mu_k) = \| r^k \|_2 > \beta > 0$ and $d(x^k, \mu_k) = 0$ if and only if $\| r^k \|_2 = 0$. Consider the directional derivative of f_{μ_k} along $\triangle x^k$ and define

$$g(t) = f_{\mu_k}(x^k + t\triangle x^k), \text{ so}$$

$$\begin{aligned} g'(0) &= \frac{d}{dt} f_{\mu_k}(x^k + t\triangle x^k)|_{t=0} \\ &= (\nabla f_{\mu_k}(x^k))^T \triangle x^k \\ &= [\nabla(c^T x^k + \mu_k \phi(x^k))]^T \triangle x^k \\ &= \mu_k (\frac{c}{\mu_k} + \triangle\phi(x^k))^T \triangle x^k \\ &= -\mu_k X_k^{-2} \| \triangle x^k \|_2^2 < 0 \end{aligned}$$

$$\text{because } \| \triangle x^k \|_2^2 = (\triangle x^k)^T (\triangle x^k)$$

$$= (\frac{c}{\mu_k} + \nabla\phi(x^k))^T \{X_k[I - X_k A^T (AX_k^2 A^T)^{-1} AX_k]X_k\}\{X_k[I - X_k A^T (AX_k^2 A^T)^{-1} A$$

$$X_k]X_k\}(\frac{c}{\mu_k} + \nabla\phi(x^k))\}$$

$$= (\frac{c}{\mu_k} + \nabla\phi(x^k))^T \{X_k^2[I - X_k A^T (AX_k^2 A^T)^{-1} AX_k]X_k^2\}(\frac{c}{\mu_k} + \nabla\phi(x^k))$$

$$= (\frac{1}{\mu_k} X_k^2)(c + \mu \nabla \phi(x^k))^T \triangle x.$$

Since f_{μ_k} is continuously differentiable, $g'(t)$ is continuous at $t = 0$. Hence, there exists an interval $[-\alpha, \alpha]$ with $\alpha > 0$ such that $g'(t) < 0$ for all t in $[\alpha, \alpha]$. This implies that $g(t)$ is monotonously decreasing on $[-\alpha, \alpha]$. In other words, when starting at x^k, moving along the direction $\triangle x^k$ with a step size α, the resulting point will end up with a lower objective function value. Let

$$\delta := argmin_{0<\alpha\leq 1}\{f_{\mu_k}(x^k + \alpha\triangle x^k) | x^k + \alpha\triangle x^k > 0\},$$

$$x^k := x^k + \delta\triangle x^k,$$

and repeat the same procedure until $\| \Delta x^k \|_2 = 0$. In this way, this algorithm generates a sequence of primal feasible solutions with monotonically decreasing objective value. Since f_{μ_k} has a unique minimal solution $x^*(\mu_k)$. By the continuity of $d(x^k, \mu_k)$ in Theorem 2.5.4, the algorithm will eventually produce a primal feasible solution x' with $d(x', \mu_k) \leq \beta$ after taking several runs of Step 5.

Theorem 2.7.3 *Under Condition 2.7.1, if $\{x^k\}$ is the sequence generated by Algorithm 2.6.1 satisfying $d(x^k, \mu_k) \leq \beta$; i.e., $\| X_k(\frac{s^k}{\mu_k}) + \nabla\phi(x^k) \|_2 \leq \beta, k = 1, 2, \cdots$, then*

$$\lim_{k \mapsto \infty} X_k s^k = 0.$$

Proof : There exists such a sequence from Theorem 2.7.2. Also,

$$\| X_k \frac{s^k}{\mu_k} \|_2 - \| X_k \nabla \phi(x^k) \|_2 \leq \| X_k(\frac{s^k}{\mu_k} + \nabla\phi(x^k)) \|_2 \leq \beta.$$

Condition 2.7.1 implies that $\| X_k s^k \|_2 \leq \mu_k(\beta + M)$.

If $\mu_k \longmapsto 0$, then $\sum_{i=1}^{n}(x_i^k s_i^k)^2 \longmapsto 0$. This further implies that either $x_i^k \longmapsto 0$ or $s_i^k \longmapsto 0$ for each component i when k tends to infinity. Therefore,

$$\lim_{k \mapsto \infty} X_k s^k = 0.$$

Theorem 2.7.4 *If the sequence $\{x^k\}$ in Theorem 2.7.3 converges, it converges to an optimal solution of P.*

proof : Suppose that $\{x^k\}$ converges to a point x^*. Since $\{x^k\}$ is a sequence of primal feasible solutions, $Ax^* = b$ and $x^* \geq 0$. If the corresponding sequence $\{s^k\}$ converges to a dual feasible solution, say s^*, then Theorem 2.7.3 generates the

complementary slackness condition for s^* and x^*. Thus, x^* is an optimal solution of P.

According to Algorithm 2.6.1, s^k is defined by

$$s^k = c - A^T(AX_k^2A^T)^{-1}AX_k^2(c + \mu_k \bigtriangledown \phi(x^k)).$$

Due to the "unique optimal solution" assumption in Assumption 2.1.6, AX always has full row rank , for $x \in W$. Consequently, matrix $AX_k^2A^T$ is invertible. On the other hand, $X_k \bigtriangledown \phi(x^k)$ is uniformly bounded by Condition 2.7.1. Therefore, s^k is a well-defined continuous function in both x and μ_k.

$$\lim_{m \mapsto \infty} s^k = s^* = c - A^T(AX^{*2}A^T)^{-1}AX^{*2}c$$

can be obtained by taking limit of s^k.

Suppose that s^* is not dual feasible, i.e., at least one component of s^* is negative, say $s_j^* < 0$, then there exists a number $N_1 > 0$ and a number $\eta < 0$ such that $s_j^k \leq \eta < 0$, for $k \geq N_1$. Moreover,

$$x_j^{k+1} = x_j^k - \delta(x_j^k)^2[\frac{s_j^k}{\mu_k} + (\bigtriangledown\phi(x^k))_j].$$

However, from Theorem 2.7.3, x_j^* has to be zero and this forces the j-th component of $\bigtriangledown\phi(x^*)$ to be negative infinity by the definition of $GBLP$ functions. This indicates that there exists a number $N_2 > 0$ such that $(\bigtriangledown\phi(x^k))_j < 0$, for $k \geq N_2$. $x_j^{k+1} = x_j^k - \delta(x_j^k)^2[\frac{s_j^k}{\mu_k} + (\bigtriangledown\phi(x^k))_j] > x_j^k$, for $k \geq N$ if $N = max(N_1, N_2)$. Therefore,

$$\lim_{k \mapsto \infty} x_j^k > x_j^N > 0.$$

This contradicts to $x_j^* = 0$. Hence, s^* must be dual feasible.

Theorem 2.7.5 (Convergence)

Algorithm 2.6.1 converges to the unique optimal solution of P under Condition 2.7.1.

Proof : For each μ_k starting from μ^0, Algorithm 2.6.1 generates a sequence $\{x^k\}$ with $d(x^k, \mu_k) \leq \beta$. Since the feasible domain is compact, there exists a convergent subsequence $\{x^{km}\}$ of $\{x^k\}$. That $\{x^{km}\}$ converges to an optimal solution of P can be conclude by applying Theorem 2.7.3 and Theorem 2.7.4 to the subsequence. But problem P is assumed to have a unique optimal solution. Therefore, $\{x^k\}$ must converge to the unique optimal solution of P.

CHAPTER III

THE PERTURBATION METHOD FOR SOLVING LINEAR SEMI-INFINITE PROGRAMMING PROBLEMS

This chapter will introduce a new method to solve linear semi-infinite programming problems by combining the concepts of two methods that were introduced in Chapter I and Chapter II.

3.1 Definitions, Assumption and Preliminaries

Recall that $LSIP$ and LP_k problem which were defined in Chapter I :

$$(LSIP) \qquad \begin{aligned} &Minimize \quad \sum_{j=1}^{n} c_j x_j. \\ &\text{Subject to} \quad \sum_{j=1}^{n} x_j a_j(t) \geq b(t), \forall t \in T, \qquad (3.1) \\ &\qquad\qquad x_j \geq 0, j = 1, \cdots, n, \qquad (3.2) \end{aligned}$$

where T is a compact metric space with an infinite cardinality, $a_j(t)$, for $j = 1, \cdots, n$ and $b(t)$ are real valued continuous functions defined on T, and LP_k was defined as follows :

$$
\begin{aligned}
& \textit{Minimize} \quad \sum_{j=1}^{n} c_j x_j. \\
(LP_k) \qquad & \\
& \text{Subject to} \quad \sum_{j=1}^{n} x_j a_j(t_i) \geq b(t_i), i = 1, \cdots, k, \\
& \qquad\qquad\quad x_j \geq 0, j = 1, \cdots, n,
\end{aligned}
$$

which is equivalent to

$$
\begin{aligned}
& \textit{Minimize} \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n. \\
(LP_k) \qquad & \\
& \text{Subject to} \quad x_1 a_1(t_1) + x_2 a_2(t_1) + \cdots + x_n a_n(t_1) \geq b(t_1), \\
& \qquad\qquad\quad x_1 a_1(t_2) + x_2 a_2(t_2) + \cdots + x_n a_n(t_2) \geq b(t_2), \\
& \qquad\qquad\qquad\qquad \vdots \\
& \qquad\qquad\quad x_1 a_1(t_k) + x_2 a_2(t_k) + \cdots + x_n a_n(t_k) \geq b(t_k), \\
& \qquad\qquad\quad x_j \geq 0, \quad j = 1, \cdots, n,
\end{aligned}
$$

or

$$
\begin{aligned}
& \textit{Minimize} \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + 0 \cdot x_{n+1} + 0 \cdot x_{n+2} + \cdots + 0 \cdot x_{n+k}. \\
(LP_k) \quad & \text{Subject to} \\
& x_1 a_1(t_1) + \cdots + x_n a_n(t_1) - x_{n+1} \cdot 1 + x_{n+2} \cdot 0 + \cdots + x_{n+k} \cdot 0 = b(t_1), \\
& x_1 a_1(t_2) + \cdots + x_n a_n(t_2) + x_{n+1} \cdot 0 - x_{n+2} \cdot 1 + \cdots + x_{n+k} \cdot 0 = b(t_2), \\
& \qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots \\
& x_1 a_1(t_k) + \cdots + x_n a_n(t_k) + x_{n+1} \cdot 0 + x_{n+2} \cdot 0 + \cdots - x_{n+k} \cdot 1 = b(t_k), \\
& x_j \geq 0, \qquad j = 1, \cdots, n, n+1, \cdots, n+k,
\end{aligned}
$$

i.e.,

$$\textit{Minimize} \quad c_1x_1 + \cdots + c_nx_n + 0 \cdot x_{n+1} + \cdots + 0 \cdot x_{n+k}.$$

$$\text{Subject to} \quad \begin{bmatrix} a_1(t_1) & \cdots & a_n(t_1) & -1 & 0 & \cdots & 0 \\ a_1(t_2) & \cdots & a_n(t_2) & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1(t_k) & \cdots & a_n(t_k) & 0 & 0 & \cdots & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n+k} \end{bmatrix} = \begin{bmatrix} b(t_1) \\ b(t_2) \\ \vdots \\ b(t_k) \end{bmatrix},$$

$$x_j \geq 0, \qquad j = 1, 2, \cdots, n+k.$$

Therefore, LP_k can be converted into a standard linear programming form SLP_k as follows :

$$(SLP_k) \qquad \begin{aligned} &\textit{Minimize} && c(k)^T x. \\ &\text{Subject to} && A(k)x = B(k), && (3.3) \\ & && x_j \geq 0, j = 1, 2, \cdots, n+k, && (3.4) \end{aligned}$$

where

$$A(k) = \begin{bmatrix} a_1(t_1) & \cdots & a_n(t_1) & -1 & 0 & \cdots & 0 \\ a_1(t_2) & \cdots & a_n(t_2) & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1(t_k) & \cdots & a_n(t_k) & 0 & 0 & \cdots & -1 \end{bmatrix}$$

is a $k \times n+k$ dimensional matrix, $B(k) = [b(t_1), b(t_2), \cdots, b(t_k)]^T$ is a k dimensional column vector, $c(k) = [c_1, \cdots, c_n, 0, \cdots, 0]^T$ and $x = [x_1, \cdots, x_n, x_{n+1}, \cdots, x_{n+k}]^T$ are $n+k$ dimensional column vectors.

Note that $A(k)$ and $A(k+1)$ are related in the following way

$$A(k+1) = \begin{bmatrix} & & & & & & 0 \\ & A(k) & & & & & 0 \\ & & & & & & \vdots \\ a_1(t_{k+1}) & \cdots & a_n(t_{k+1}) & 0 & 0 & \cdots & -1 \end{bmatrix},$$

and $A(k)$ has a full row rank k for all k.

Assume that $LSIP$ and SLP_k satisfy the following assumptions:

Assumption 3.1.1

1. **Interior Point Assumption :**

   LSIP has an interior feasible solution relative to (3.2). *Since the feasible domain of* $LSIP$ *is contained in the feasible domain of* SLP_k, *for* $k = 1, 2\cdots$, SLP_k *also has an interior feasible solution, for* $k = 1, 2, \cdots$.

2. **Bounded Feasible Domain Assumption :**

   SLP_k *has a feasible domain* $W_k = \{x \in R^{n+k} : \sum_{j=1}^{n} x_j a_j(t_i) - x_{n+i} = b(t_i),$ $i = 1, 2, \cdots, k,$ *and* $x_j \geq 0,\ j = 1, \cdots, n\}$ *which is a bounded subset of* R^{n+k}, *for* $k = 1, 2, \cdots$.

3. SLP_k *has a unique optimal solution, for* $k = 1, 2, \cdots$.

4. $A(k)$ *has a full row rank* k *which can be seen by the form of* $A(k)$.

Note that SLP_k satisfies Assumption 2.1.6 if it satisfies Assumption 3.1.1.

In order to develop a convergent algorithm for solving $LSIP$, it is necessary to have the following condition :

Condition 3.1.2 *There exists* $M_k > 0$ *such that* $\| X \bigtriangledown \phi_k(x) \|_2 \leq M_k$ *for each* $x \in W_k$, *and* $k = 1, \cdots$, *where* $\phi_k(x)$ *is the GBLP function for solving* SLP_k.

Let $\phi_k(x)$ satisfy Condition 3.1.2. A new method for solving $LSIP$ is developed by the concept of solving SLP_k approximately with the "perturbation method" instead of solving SLP_k exactly in the "adding constraint" method for solving $LSIP$. Note that "solving SLP_k approximately" is actually "solving SLP_{μ_k} approximately", and SLP_{μ_k} is a "relaxed" problem of SLP_k.

3.2 The New Algorithm for Solving Linear Semi-Infinite Programming problems

Algorithm 3.2.1

Step 1.(Initialization and Notations)

Given $0 < \delta_1, \delta_2, \delta_3, \beta_1 < 1$, *where* $\delta_1, \delta_2, \delta_3, \beta_1$ *are all constants.*

Given $\epsilon > 0$ *be a sufficiently small number and given* $\epsilon_1, \mu_1 > 0$, $t_1 \in T$.

Set $T_1 := \{t_1\}$.

Set $k := 1$.

Step 2. *Find an initial interior feasible solution* x^k *of* SLP_k *such that*

$d(x^k, \mu_k) \leq \beta_k$. *Let*

$$f_{\mu_k}(x) = \sum_{j=1}^{n+k} c_j x_j + \mu_k \phi_k(x) = c(k)^T x + \mu_k \phi_k(x),$$

where $\phi_k(x)$ *is a GBLP function which satisfies Condition 3.1.2.*

Step 3. *Find the minimizer* t_{k+1} *of* $G_{k+1}(t)$ *which is given by*

$$G_{k+1}(t) = \sum_{j=1}^{n} a_j(t)x_j^k - b(t).$$

Calculate

$$G_{k+1}(t_{k+1}) = \sum_{j=1}^{n} a_j(t_{k+1})x_j^k - b(t_{k+1}).$$

Step 4. (Check Feasibility)

If $G_{k+1}(t_{k+1}) < -\epsilon_k$,

let $T_{k+1} = T_k \cup \{t_{k+1}\}$,

$\epsilon_{k+1} = (1-\delta_2)\epsilon_k$,

$\mu_{k+1} = (1-\delta_1)\mu_k$,

$\beta_{k+1} = (1-\delta_3)\beta_k$. *Reset* $k \longleftarrow k+1$, *go to Step 2.*

Otherwise, go to Step 5.

Step 5. (Check Optimum)

Check μ_k, *if* $\mu_k \leq \epsilon$ *then* **Stop !**

$(x_1^k, x_2^k, \cdots, x_n^k)$ *is an optimal solution of LSIP, where* $(x_1^k, x_2^k, \cdots, x_n^k)$

denotes the first n component of x^k, *and* x^k *is a* $n+k$ *dimensional vector.*

Otherwise, go to Step 6.

Step 6. (Compute Dual Estimate of SLP_k **)**

$y_k := [A(k)X_k^2A(k)^T]^{-1}A(k)X_k^2(c + \mu_k \bigtriangledown \phi_k(x^k))$,

$s_k := c - A(k)^T y_k$,

$r_k := X_k(\frac{1}{\mu_k}s_k + \bigtriangledown\phi_k(x^k))$,

$X_k := diag(x_1^k, x_2^k, \cdots, x_{n+k}^k)$.

Step 7. (Check Criterion of Closeness)

Let $d(x^k, \mu_k) := \| r_k \|_2$.

If $d(x^k, \mu_k) \leq \beta_k$ *then set*

$$x^k = x^k - X_k r_k = x^k + \Delta x^k,$$

$$\mu_k = (1 - \delta_1)\mu_k,$$

$$\epsilon_k = (1 - \delta_2)\epsilon_k,$$

go to Step 3.

Otherwise, go to Step 8.

Step 8.(Compute Another Solution Closer to the Optimal Solution of SLP_k**)**

$\delta := argmin_{0<\alpha\leq 1}\{f_{\mu_k}(x^k - \alpha X_k r_k) \mid x^k - \alpha X_k r_k > 0\}$,

$x^k = x^k - \delta X_k r_k$.

Go to Step 6 without updating k *and* μ_k.

Note that Step 4 checks $G_{k+1}(t_{k+1}) < -\epsilon_k$ or $G_{k+1}(t_{k+1}) \geq -\epsilon_k$ instead of checking $G_{k+1}(t_{k+1}) < 0$ or $G_{k+1}(t_{k+1}) \geq 0$. This method is called "relaxation" method in numerical practice.

3.3 The Results of the Convergence for Algorithm 3.2.1

Denote the sequence generated by Algorithm 3.2.1 be $\{x(\mu_k^i) : i = 1, 2, \cdots, k = 1, 2, \cdots\}$, the sequence or a subsequence of $\{x(\mu_k^i) : i = 1, 2, \cdots, k = 1, 2, \cdots\}$, converges to the optimal solution of $LSIP$. This can be shown by the following Theorems :

Theorem 3.3.1 *Under Assumption 3.1.1, the sequence* $\{x(\mu_k^i) : i = 1, 2, \cdots\}$ *found in Algorithm 3.2.1 converges to the optimal solution* x^k *of* SLP_k *when* $\{\mu_k^i : i = 1, 2, \cdots\}$ *is a sequence of positive numbers which decreases to zero.*

Proof : Since Algorithm 3.2.1 selects $x(\mu_k^i)$ satisfying $d(x(\mu_k^i), \mu_k^i) \leq \beta_k$ and $\phi_k(x)$ satisfying Condition 3.1.2 (or Condition 2.7.1); therefore, $\{x(\mu_k^i) : i = 1, 2 \cdots\}$ converges to x^k when $\{\mu_k^i : i = 1, 2, \cdots\}$ is a sequence of positive numbers which decreases to zero by Theorem 2.7.5.

Under the Interior Point Assumption on $LSIP$, the Bounded Feasible Domain Assumption on SLP_k, and the assumption that SLP_k has a unique optimal solution for every k, the approach developed in Algorithm 3.2.1 either terminates in finite steps or generates an infinite sequence $\{x(\mu_k), k = 1, 2, \cdots\}$.

In Algorithm 3.2.1, ϵ_k, for $k = 1, \cdots$, μ_k, for $k = 1, \cdots$, are defined and the relation between ϵ_k, μ_k $k = 2, \cdots$ can be seen as follows :

$$\begin{cases} \epsilon_{k+1} = (1 - \delta_2)\epsilon_k, & if \quad \mu_{k+1} = (1 - \delta_1)\mu_k, \\ \epsilon_k = (1 - \delta_2)\epsilon_k, & if \quad \mu_k = (1 - \delta_1)\mu_k. \end{cases} \tag{3.5}$$

Theorem 3.3.2 *If Algorithm 3.2.1 terminates in finite steps when* $k = k^*$, *then* x^{k^*} *is the optimal solution of LSIP.*

Proof : In this case, the algorithm stops when μ_{k^*} is sufficiently small and x^{k^*} satisfies

$$Minimum_{t \in T} \sum_{j=1}^{n} a_j(t) x_j^{k^*} - b(t) \geq -\epsilon_{k^*}.$$

Therefore, when Algorithm 3.2.1 terminates in finite steps, i.e., μ_{k^*} is sufficiently small, ϵ_{k^*} is sufficiently small can be concluded because of the relation between ϵ_{k^*} and μ_{k^*} in (3.5). Hence,

$$Minimum_{t\in T}\sum_{j=1}^{n} a_j(t)x_j^k - b(t) \geq -\epsilon_k \longrightarrow 0 \text{ as } k \text{ large enough.}$$

i.e.,

$$\sum_{j=1}^{n} a_j(t)x_j^{k^*} - b(t) \geq 0.$$

This means x^{k^*} is feasible to $LSIP$.

From the result of Theorem 3.3.1, x^{k^*} is the optimal solution of SLP_{k^*}; Moreover, x^{k^*} is feasible to $LSIP$, x^{k^*} must be the optimal solution of $LSIP$ because the feasible domain of SLP_{k^*} contains the feasible domain of $LSIP$.

Theorem 3.3.3 *In Algorithm 3.1, if the approach does not terminate in finite steps; or say, it generates an infinite sequence* $\{x(\mu_k) : k = 1, 2, \cdots\}$, *then there exists a subsequence* $\{x(\mu_{k_i}) : i = 1, 2, \cdots\}$ *of* $\{x(\mu_k) : k = 1, 2, \cdots\}$ *and the first n component* $\{(x(\mu_{k_i})_1, x(\mu_{k_i})_2, \cdots, x(\mu_{k_i})_n) : i = 1, 2, \cdots\}$ *of* $\{x(\mu_{k_i}) : i = 1, \cdots, \}$ *converges to the optimal solution* x^* *of* $LSIP$.

Proof: In this case, however $\mu_{k+1} = (1-\delta_1)\mu_k \leq (1-\delta_1)^k\mu_1$ for a given $0 < \delta_1 < 1$ by (3.5). Hence, $\{\mu_k : k = 1, 2, \cdots\}$ decreases to zero.

1. The feasible domain W_k of SLP_k(or SLP_{μ_k}) is contained in W_{k-1} for $k = 2, 3, \cdots$ because the constraints of SLP_{k-1} are also constraints of SLP_k. Under the Bounded Feasible Domain Assumption, there exists a subsequence $\{\mu_{k_i}\}$ of $\{\mu_k\}$ such that $\{(x(\mu_{k_i})_1, x(\mu_{k_i})_2, \cdots, x(\mu_{k_i})_n)\}$ converges to a point, say, x^*.

2. T is compact, a subsequence $\{\mu_{m_i}\}$ of $\{\mu_{k_i}\}$; or consequently,

   $\{(x(\mu_{m_i})_1, \cdots, x(\mu_{m_i})_n)\}$ subsequence of $\{(x(\mu_{k_i})_1, \cdots, x(\mu_{k_i})_n)\}$ can be chosen such that $\{t_{m_i+1}\}$ converges to a limit point $t^* \in T$.

3. Now x^* is the optimal solution of $LSIP$ can be shown as follows :

   (1) $x(\mu_{k_i}) > 0$ and $\{(x(\mu_{k_i})_1, \cdots, x(\mu_{k_i})_n)\}$ converges to x^*, so $x^* \geq 0$.

   (2) Define

   $$G^*(t) = \sum_{j=1}^{n} a_j(t)x_j^* - b(t) \text{ on T.}$$

   Since x^* is the limit point of the sequence $\{(x(\mu_{k_i})_1, \cdots, x(\mu_{k_i})_n)\}$,

   $$G^*(t_i) = \sum_{j=1}^{n} a_j(t_i)x_j^* - b(t_i) \geq 0, \quad for \quad i = 1, 2, \cdots. \tag{3.6}$$

   by the definition of x^*.

   Let $\bar{t} \in T$ be a minimizer of $G^*(t)$ over T. From the way of choosing t_{k+1} in Step 3,

   $$G_{m_i+1}(\bar{t}) \geq G_{m_i+1}(t_{m_i+1}) \tag{3.7}$$

   can be obtained because t_{m_i+1} is the minimizer of $G_{m_i+1}(t)$. Let m_i approach to ∞, then

   $$G^*(\bar{t}) \geq G^*(t^*) \geq 0$$

   can be obtained by (3.6) and (3.7). Thus x^* is a feasible solution of $LSIP$.

   (3) Since $d(x(\mu_{m_i}), \mu_{m_i}) \leq \beta_{m_i}$ which approaches to zero as m_i approaches to infinity, $x(\mu_{m_i})$ approaches to the optimal solution $x_{m_i}^*$ of $SLP_{\mu_{m_i}}$ as m_i large enough.

(4) Suppose x^* is not the optimal solution of $LSIP$, under the assumption that $LSIP$ has an optimal solution, say $\bar{x} = (\bar{x_1}, \bar{x_2}, \cdots, \bar{x_n})^T$. Then there exists a constant $w > 0$ such that

$$\sum_{j=1}^{n} c_j x_j^* - \sum_{j=1}^{n} c_j \bar{x_j} > w. \tag{3.8}$$

Since $\bar{x}$ is feasible to $LSIP$, it is feasible to $SLP_{\mu_{m_i}}$ for each m_i; therefore,

$$\sum_{j=1}^{n} c_j \bar{x_j} + \mu_{m_i} \phi_{m_i}(\bar{x}) \geq \sum_{j=1}^{n} c_j (x_{m_i}^*)_j + \mu_{m_i} \phi_{m_i}(x_{m_i}^*). \tag{3.9}$$

It is obvious to see that

$$c^T(x(\mu_{m_i})_1, \cdots, x(\mu_{m_i})_n) - c^T\bar{x} = c^T(x(\mu_{m_i})_1, \cdots, x(\mu_{m_i})_n) - c^T x^* + c^T x^* - c^T \bar{x}. \tag{3.10}$$

Since $\{(x(\mu_{m_i})_1, \cdots, x(\mu_{m_i})_n)\}$ converges to x^*, given a small number γ with $w > \gamma > 0$, there exists a positive number N_2 such that

$$-\gamma < c^T(x(\mu_{m_i})_1, \cdots, x(\mu_{m_i})_n) - c^T x^* < \gamma \quad for \quad m_i \geq N_2. \tag{3.11}$$

Moreover, $x(\mu_{m_i})$ approaches to $x_{m_i}^*$ as m_i approaches to infinity can be concluded from the discussion in (3). So, if given $\epsilon = \frac{1}{2}(w - \gamma)$, then there exists a positive number N_1 such that for $m_i \geq N_1$,

$$\sum_{j=1}^{n} c_j (x_{m_i}^*)_j + \mu_{m_i} \phi_{m_i}(x_{m_i}^*) \geq [\sum_{j=1}^{n} c_j x(\mu_{m_i})_j + \mu_{m_i} \phi_{m_i}(x(\mu_{m_i}))] - \epsilon. \tag{3.12}$$

Hence, when $m_i > N_1$,

$$\sum_{j=1}^{n} c_j \bar{x_j} + \mu_{m_i} \phi_{m_i}(\bar{x}) \geq [\sum_{j=1}^{n} c_j x(\mu_{m_i})_j + \mu_{m_i} \phi_{m_i}(x(\mu_{m_i}))] - \epsilon \tag{3.13}$$

can be obtained by (3.9) and (3.12). Moreover, the inequality (3.13) is equivalent to

$$\sum_{j=1}^{n} c_j x(\mu_{m_i})_j - \sum_{j=1}^{n} c_j \bar{x}_j \leq \mu_{m_i}[\phi_{m_i}(\bar{x}) - \phi_{m_i}(x(\mu_{m_i}))] + \epsilon \qquad (3.14)$$

whenever $m_i \geq N_1$ and $\epsilon = \frac{1}{2}(w - \gamma)$. Therefore,

when $m_i \geq \text{Max } (N_1, N_2)$,

$$c^T(x(\mu_{m_i})_1, \cdots, x(\mu_{m_i})_n) - c^T\bar{x} > w - \gamma > 0 \qquad (3.15)$$

can be deduced by (3.8),(3.10),and (3.11). Also,

$$0 < w-\gamma < c^T(x(\mu_{m_i})_1, \cdots, x(\mu_{m_i})_n) - c^T\bar{x} \leq \mu_{m_i}[\phi_{m_i}(\bar{x}) - \phi_{m_i}(x(\mu_{m_i}))]$$

$$+\frac{1}{2}(w - \gamma) \qquad (3.16)$$

can be obtained by (3.14) and (3.15).

Note that $\mu_{k+1} \leq (1 - \delta_1)^k \mu_1, k = 1, 2, \cdots$; therefore, μ_{m_i} reduces to zero as m_i approaches to infinity. By the Bounded Feasible Domain Assumption and the definition of $GBLP$ function, the right sub-term of (3.16) :

$$\mu_{m_i}[\phi_{m_i}(\bar{x}) - \phi_{m_i}(x(\mu_{m_i}))]$$

approaches to zero as m_i approaches to infinity. Therefore, $0 < w-\gamma \leq \frac{1}{2}(w - \gamma)$. This causes a contradiction. Thus x^* must be the optimal solution of $LSIP$.

3.4 Improvement of Algorithm 3.2.1

The purpose of this section is trying to make Algorithm 3.2.1 easier. Two problems "how to find an initial interior feasible solution of SLP_{k+1} whenever

$G_{k+1}(t_{k+1}) < -\epsilon_k < 0$," and "how to get an interior feasible solution of SLP_{k+1} satisfying $d(x^{k+1}\mu_{k+1}) \leq \beta_{k+1}$" in step 2 of Algorithm 3.2.1 can be solved by the following discussion :

3.4.1 Find An Initial Interior Feasible Solution of SLP_{k+1} Whenever $G_{k+1}(t_{k+1}) < -\epsilon_k < 0$

Define

$$J(x) = \sum_{j=1}^{n} a_j(t_{k+1})x_j - b(t_{k+1}).$$

Suppose $x^k = (x_1^k, x_2^k, \cdots, x_{n+k}^k)^T$ is an interior feasible solution of SLP_k such that $J(x^k) < 0$ and $x^{k+1} = (x_1^k + \Delta x_1^k, x_2^k + \Delta x_2^k, \cdots, x_{n+k}^k + \Delta x_{n+k}^k, x_{n+k+1}^{k+1})$ is an interior feasible solution of SLP_{k+1} :

$$(SLP_{k+1}) \qquad \begin{aligned} &\textit{Minimize} \quad c(k+1)^T x. \\ &\text{subject to} \quad A(k+1)x = B(k+1), \qquad (3.17) \\ &\qquad\qquad x \geq 0. \qquad (3.18) \end{aligned}$$

the constraints of SLP_{k+1} can be rewritten as follows :

$$\begin{cases} \begin{bmatrix} & & & & 0 \\ & A(k) & & & 0 \\ & & & & \vdots \\ a_1(t_{k+1}) & \cdots & a_n(t_{k+1}) & 0\ 0\ \cdots & -1 \end{bmatrix} \begin{pmatrix} x_1^k + \Delta x_1^k \\ \vdots \\ x_{n+k}^k + \Delta x_{n+k}^k \\ x_{n+k+1}^{k+1} \end{pmatrix} = \begin{pmatrix} B(k) \\ b(t_{k+1}) \end{pmatrix}, \\ x_j^k + \Delta x_j^k > 0, \text{ for } j = 1, 2, \cdots, n+k, \\ x_{n+k+1}^{k+1} = \sum_{j=1}^{n} a_j(t_{k+1})(x_j^k + \Delta x_j^k) - b(t_{k+1}) > 0. \end{cases} \qquad (3.19)$$

Reduce (3.19) to be

$$\begin{cases} A(k)(x^k + \Delta x^k) = B(k), \\ x_{n+k+1}^{k+1} = \sum_{j=1}^{n} a_j(t_{k+1})(x_j^k + \Delta x_j^k) - b(t_{k+1}) > 0, \\ x_j^k + \Delta x_j^k > 0, \text{ for } j = 1, 2, \cdots, n+k. \end{cases} \tag{3.20}$$

$$\begin{cases} A(k)\Delta x^k = 0, \\ x_{n+k+1}^{k+1} = \sum_{j=1}^{n} a_j(t_{k+1})(x_j^k + \Delta x_j^k) - b(t_{k+1}) > 0, \\ x_j^k + \Delta x_j^k > 0, \quad \text{for} \quad j = 1, 2, \cdots, n+k. \end{cases} \tag{3.21}$$

can be obtained by reducing (3.20).

By the particular form of the matrix

$$A(k) = \begin{bmatrix} a_1(t_1) & \cdots & a_n(t_1) & -1 & 0 & \cdots & 0 \\ a_1(t_2) & \cdots & a_n(t_2) & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1(t_k) & \cdots & a_n(t_k) & 0 & 0 & \cdots & -1 \end{bmatrix},$$

it is obviously that its null space is spanned by

$$\begin{aligned} z_1 &= (1, 0, 0, \cdots, 0, a_1(t_1), a_1(t_2), \cdots, a_1(t_k))^T, \\ z_2 &= (0, 1, 0, \cdots, 0, a_2(t_1), a_2(t_2), \cdots, a_2(t_k))^T, \\ &\vdots \\ z_n &= (0, 0, 0, \cdots, 1, a_n(t_1), a_n(t_2), \cdots, a_n(t_k))^T, \end{aligned}$$

which are $n+k$ dimensional column vectors. Since $A(k)\Delta x^k = 0$, Δx^k is the linear combination of $z_1, z_2, \cdots, z_n$.

Define α_k, θ_k, and x_j^{k+1} as follows :

$$\alpha_k := sign(\sum_{j=1}^{n} a_j(t_{k+1})\Delta x_j^k),$$

$$\theta_k := \tau Minimum\{\frac{x_j^k}{-\alpha_k \Delta x_j^k} : \alpha_k \Delta x_j^k < 0\} > 0, \text{ where } 0 < \tau < 1,$$

$$x_j^{k+1} := x_j^k + \theta_k \alpha_k \Delta x_j^k > 0 \text{ for } j = 1, 2, \cdots, n+k,$$

then

$$\begin{aligned} J(x^{k+1}) &= \sum_{j=1}^{n} a_j(t_{k+1})(x_j^k + \theta_k \alpha_k \Delta x_j^k) - b(t_{k+1}), \\ &= \theta_k[\alpha_k \sum_{j=1}^{n} a_j(t_{k+1}) \Delta x_j^k] + J(x^k), \end{aligned} \tag{3.22}$$

where

$$\theta_k[\alpha_k \sum_{j=1}^{n} a_j(t_{k+1}) \Delta x_j^k] \geq 0,$$

by the definition of α_k, θ_k and the assumption: $J(x^k) < 0$. Hence $J(x^{k+1}) \geq J(x^k)$. Without loss of the generality, repeat this process until $J(x^{k+1}) \geq 0$. Δx^k can be found to satisfy (3.22) by the above discussion, i.e., an initial interior feasible solution of SLP_{k+1} can be found easily by using the combination of the vectors $z_1, \cdots, z_n$ and α_k, θ_k.

3.4.2 Find An Interior Feasible Solution x^{k+1} of SLP_{k+1} Satisfying $d(x^{k+1}, \mu_{k+1}) \leq \beta_{k+1}$

Now the remained problem is "how to get an interior feasible solution of SLP_{k+1} satisfying $d(x^{k+1}\mu_{k+1}) \leq \beta_{k+1}$." If there is a primal feasible solution x^{k+1} with $d(x^{k+1}, \mu_{k+1}) > \beta_{k+1}$, then Step 8 of the Algorithm 3.1 will eventually generate a primal feasible solution x' of SLP_{k+1} such that $d(x', \mu_{k+1}) \leq \beta_{k+1}$ by the result of Theorem 2.7.2.

3.4.3 The Revised Algorithm

With the help of the above discussion, an initial interior feasible solution of SLP_{k+1} such that $d(x^{k+1}, \mu_{k+1}) \leq \beta_{k+1}$ can be found. Algorithm 3.2.1 can be

modified as follows :

Algorithm 3.4.1

Step 1. (Initialization and Notations)

Given $0 < \delta_1, \delta_2, \delta_3, \beta_1 < 1$, $\delta_1, \delta_2, \delta_3, \beta_1$ *are constants,* $\epsilon > 0$ *sufficiently small,* $\epsilon_1, \mu_1 > 0$, $t_1 \in T$.

Set $T_1 = \{t_1\}$.

Set $k := 1$.

Step 2. *Find an initial interior feasible solution* x^k *of* SLP_k.

If $k > 1$ *then using the discussion of subsection 3.4.1 to find it.*

Step 3. (Compute Dual Estimate of SLP_k and Step Length of x^k)

$y_k := [A(k)X_k^2A(k)^T]^{-1}A(k)X_k^2(c + \mu_k \bigtriangledown \phi_k(x^k))$,

$s_k := c - A(k)^T y_k$,

$r_k := X_k(\frac{1}{\mu_k}s_k + \bigtriangledown\phi_k(x^k))$,

where $X_k := diag(x_1^k, x_2^k, \cdots x_{n+k}^k)$.

$d(x^k, \mu_k) :=\| r_k \|_2$,

$\Delta x^k = -X_k r_k$,

where $\phi_k(x)$ *is a GBLP function which satisfies Condition 3.1.2*

Let

$$f_{\mu_k}(x) = \sum_{j=1}^{n+k} c_j x_j + \mu_k \phi_k(x) = c^T x + \mu_k \phi_k(x).$$

Step 4. (Check Criterion of Closeness)

If $d(x^k, \mu_k) > \beta_k$ *then*

$\delta := argmin_{0<\alpha\leq 1}\{f_{\mu_k}(x^k - \alpha X_k r_k) \mid x_k - \alpha X_k r_k > 0\}$,

$x^k = x^k - \delta \Delta x^k$,

go to Step 3.

Otherwise, go to Step 5.

Step 5. (Check Feasibility)

Find minimizer t_{k+1} *of* $G_{k+1}(t)$, *where*

$$G_{k+1}(t) = \sum_{j=1}^{n} a_j(t) x_j^k - b(t).$$

If $G_{k+1}(t_{k+1}) < -\epsilon_k$ *then*

$T_{k+1} = T_k \cup \{t_{k+1}\}$,

$\epsilon_{k+1} = (1 - \delta_2)\epsilon_k$,

$\mu_{k+1} = (1 - \delta_1)\mu_k$,

$\beta_{k+1} = (1 - \delta_3)\beta_k$,

go to Step 2.

Otherwise, (i.e., $G_{k+1}(t_{k+1}) \geq -\epsilon_k$*)*

go to Step 6.

Step 6. (Check Optimum)

Check μ_k.

If $\mu_k \leq \epsilon$ *then* **Stop!**

$(x_1^k, x_2^k, \cdots, x_n^k)$ *is the optimal solution of LSIP.*

Otherwise,

$$x^k := x^k - X_k r_k = x^k + \Delta x^k,$$

$$\mu_k = (1 - \delta_1)\mu_k,$$

$$\epsilon_k = (1 - \delta_2)\epsilon_k,$$

go to Step 3.

CHAPTER IV

NUMERICAL EXAMPLES AND THE CONCLUDING REMARK

Compared to the traditional extreme-point methods [6,10], at each iteration $k, k = 1, 2, \cdots$, for $\mu_k > 0$, the perturbation approach solves an approximate solution of LP_{μ_k} which is a "relaxed problem" of LP_k. In view of the interior-point method for linear programming problems [5], solving LP_{μ_k} only requires a partial amount of computational effort spending for LP_k, because the common practice of employing the perturbation interior-point method for solving LP_k is to solve a perturbed problem {Minimize $c(k)^T x + \mu_k \phi_k(x) \mid A(k)x = B(k), x \geq 0, \phi_k(x)$ is a convex "barrier function"} and then drive the positive parameter μ_k to zero [3]. In the case of this paper, the perturbation approach for solving linear semi-infinite programming problems takes $\phi_k(x)$ as the $GBLP$ function which satisfies some conditions and reduces the parameter μ_k such that the approximate solution of LP_{μ_k} satisfying the "closeness criterion": $d(x^k, \mu_k) \leq \beta_{k+1}$ whenever $G_{k+1}(t_{k+1}) < -\epsilon_k$.

Some computational experiences on implementing Algorithm 3.4.1 for solving $LSIP$ will be reported in this chapter in order to demonstrate that the new method which is developed in Chapter III is very efficient and useful in practice.

Three sets of commonly seen problems as following were tested.

Problem 1

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{n} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{n} t^{j-1} x_j \geq \sin t, \ \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ \ j = 1, 2, \cdots, n. \end{aligned}$$

Problem 2

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{n} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{n} t^{j-1} x_j \geq \frac{1}{2-t}, \ \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ \ j = 1, 2, \cdots, n. \end{aligned}$$

Problem 3

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{n} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{n} t^{j-1} x_j \geq \frac{1}{1+t^2}, \ \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ \ j = 1, 2, \cdots, n. \end{aligned}$$

4.1 Test 1 (The Adding Constraint Method and the Entropic Perturbation Method for $n = 5$ Case)

Set $n = 5$ as a baseline case in the following tests. For the purpose of illustration and comparison, the "adding constraint" method was implemented simply by using LOQO(an interior-point LP and QP package provided by Vanderbei [11]) to solve LP_k in step 2 of each iteration. All default parameters of LOQO were kept

unchanged and the stopping rule of LOQO was set when the duality gap $\leq 10^{-8}$. The stopping rule in Step 4 of Algorithm 1.2.1 was set when $G_k(t_{k+1}) \geq 10^{-4}$. A starting point with $t_1 = 0.5$ was used; on the other hand, Algorithm 3.4.1 was implemented as follows :

$$\text{the starting point } t_1 = 0.5,$$

$$\delta_1 = 0.75,$$

$$\delta_2 = 0.75,$$

$$\delta_3 = 0.05,$$

$$\beta_1 = 0.95,$$

$$\epsilon_1 = 0.1,$$

$$\epsilon = 10^{-6},$$

$$\mu = 0.1.$$

Moreover, the entropic barrier functions were used as the $GBLP$ functions, i.e.,

$$\phi_k(x) = \sum_{i=1}^{n+k} x_i \ln x_i.$$

μ_{k+1} and ϵ_{k+1}, for $k = 1, 2, \cdots$ were defined as follows :

$$\mu_{k+1} := \begin{cases} (1-\delta_1)\mu_k & \text{when} \quad \mu_k \geq \epsilon, \\ \mu_k & \text{when} \quad \mu_k < \epsilon. \end{cases}$$

$$\epsilon_{k+1} := \begin{cases} (1-\delta_2)\epsilon_k & \text{when} \quad \epsilon_k \geq 10^{-4}, \\ \epsilon_k & \text{when} \quad \epsilon_k < 10^{-4}. \end{cases}$$

For $n = 5$ cases, Problems 1,2,3 can be written as follows :

Problem 1

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{5} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{5} t^{j-1} x_j \geq \sin t, \ \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ \ j = 1, 2, \cdots, 5. \end{aligned}$$

Problem 2

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{5} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{5} t^{j-1} x_j \geq \frac{1}{2-t}, \ \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ \ j = 1, 2, \cdots, 5. \end{aligned}$$

Problem 3

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{5} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{5} t^{j-1} x_j \geq \frac{1}{1+t^2}, \ \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ \ j = 1, 2, \cdots, 5. \end{aligned}$$

Numerical results of the tested problems are reported in Tables 4.1.1, 4.1.2, and 4.1.3.

Table 4.1.1 The result of Test 1 for Problem 1

Algorithm 1.2.1 Adding Constraint Method			**Algorithm 3.4.1 Perturbation Method**		
k	Number of Cholesky	Objective value	Number of Cholesky for finding a suitable initial solution of SLP_k	Number of Cholesky spent after finding a suitable initial solution of SLP_k	Objective value
1	9	0.479430	2	2	0.548731
2	9	0.479430	4	0	0.491573
3	8	0.479430	3	0	0.481737
4	9	0.479430	3	0	0.479730
5			3	0	0.479522
6			3	6	0.479425
Total number of Cholesky		35	18	8	

Table 4.1.2 The result of Test 1 for Problem 2

Algorithm 1.2.1 Adding Constraint Method			**Algorithm 3.4.1 Perturbation Method**		
k	Number of Cholesky	Objective value	Number of Cholesky for finding a suitable initial solution of SLP_k	Number of Cholesky spent after finding a suitable initial solution of SLP_k	Objective value
1	9	0.666670	2	2	0.692265
2	9	0.666670	4	0	0.677401
3	9	0.692859	4	2	0.693360
4	9	0.693446	4	8	0.693454
5	9	0.693489			
Total number of Cholesky		45	14	12	

Table 4.1.3 The result of Test 1 for Problem 3

Algorithm 1.2.1 Adding Constraint Method			**Algorithm 3.4.1 Entropic Perturbation Method**		
k	Number of Cholesky	Objective value	Number of Cholesky for finding a suitable initial solution of SLP_k	Number of Cholesky spent after finding a suitable initial solution of SLP_k	Objective value
1	9	0.800000	2	0	0.875792
2	8	1.000000	2	16	1.000000
Total number of Cholesky		17	4	16	

The optimal solution of Problem 1 obtained by Algorithm 1.2.1 is $x_1 = 0.040632, x_2 = 0.877595, x_3 = 0.000000, x_4 = 0.000000, x_5 = 0.000000$, while Algorithm 3.4.1 provides an optimal solution of Problem 1 : $x_1 = 0.044744, x_2 = 0.869361, x_3 = 0.000000, x_4 = 0.000000, x_5 = 0.000000$. The optimal value of Problem 1 obtained by Algorithm 1.2.1 is 0.479430, while Algorithm 3.4.1 provided an optimal value of Problem 1 : 0.479425.

The optimal solution of Problem 2 obtained by Algorithm 1.2.1 is $x_1 = 0.500000, x_2 = 0.252752, x_3 = 0.132483, x_4 = 0.000000, x_5 = 0.114763$, while Algorithm 3.4.1 provides an optimal solution of Problem 2 : $x_1 = 0.500000, x_2 = 0.255819, x_3 = 0.125300, x_4 = 0.000002, x_5 = 0.118858$. The optimal solution of Problem 2 obtained by Algorithm 1.2.1 is 0.693489, while Algorithm 3.4.1 obtained an optimal value of Problem 2 : 0.693454.

The optimal solution of Problem 3 obtained by Algorithm 1 is $x_1 = 1.000000, x_2 = 0.000000, x_3 = 0.000000, x_4 = 0.000000, x_5 = 0.000000$, while Algorithm 3.4.1 provides an optimal solution of Problem 3 : $x_1 = 1.000000, x_2 = 0.000000, x_3 = 0.000000, x_4 = 0.000000, x_5 = 0.000000$. The optimal value of Problem 3 obtained by Algorithm 1.2.1 is 1.000000, while Algorithm 3.4.1 obtained an optimal value of Problem 3 : 1.000000.

After finding a suitable initial solution of SLP_k, that the "perturbation" method is more efficient than the "adding constraint" method can be found from the above data.

4.2 Test 2 (The Minus Logarithmic Perturbation Method for $n = 5$ Case)

The baseline case of $n = 5$ was chosen in the following tests. The parameters for Algorithm 3.4.1 were set as follows :

$$\text{the starting point } t_1 = 0.5,$$

$$\delta_1 = 0.75,$$

$$\delta_2 = 0.75,$$

$$\delta_3 = 0.05,$$

$$\beta_1 = 0.95,$$

$$\epsilon_1 = 0.1,$$

$$\epsilon = 10^{-6},$$

$$\mu = 0.1,$$

and the $GBLP$ functions were chosen as minus logarithmic barrier functions, i.e.,

$$\phi_k(x) = -\sum_{i=1}^{n+k} \ln x_i.$$

Moreover, μ_{k+1} and ϵ_{k+1}, for $k = 1, 2, \cdots$ were defined as follows :

$$\mu_{k+1} := \begin{cases} (1-\delta_1)\mu_k & \text{when} \quad \mu_k \geq \epsilon, \\ \mu_k & \text{when} \quad \mu_k < \epsilon. \end{cases}$$

$$\epsilon_{k+1} := \begin{cases} (1-\delta_2)\epsilon_k & \text{when} \quad \epsilon_k \geq 10^{-4}, \\ \epsilon_k & \text{when} \quad \epsilon_k < 10^{-4}. \end{cases}$$

The solution process is shown by Tables 4.2.1, 4.2.2, and 4.2.3.

Table 4.2.1 The result of Test 2 for Problem 1

Iteration	Objective Value
1	0.507489
2	0.509230
3	0.479431
4	0.479425

Problem 1

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{5} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{5} t^{j-1} x_j \geq \sin t, \ \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ \ j = 1, 2, \cdots, 5. \end{aligned}$$

The optimal solution obtained by Algorithm 3.4.1 is $x_1 = 0.041393, x_2 = 0.876063, x_3 = 0.000001, x_4 = 0.000000, x_5 = 0.000000$, and the optimal value is 0.479425.

Table 4.2.2 The result of Test 2 for Problem 2

Iteration	Objective Value
1	0.755381
2	0.690755
3	0.692997
4	0.693446

Problem 2

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{5} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{5} t^{j-1} x_j \geq \frac{1}{2-t}, \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ j = 1, 2, \cdots, 5. \end{aligned}$$

The optimal solution obtained by Algorithm 3.4.1 is $x_1 = 0.500000, x_2 = 0.256528, x_3 = 0.123644, x_4 = 0.000002, x_5 = 0.119800$, and the optimal value is 0.693446.

Table 4.2.3 the result of Test 2 for Problem 3

iteration	Objective value
1	0.111728
2	1.000000

Problem 3

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{5} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{5} t^{j-1} x_j \geq \frac{1}{1+t^2}, \quad \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \quad j = 1, 2, \cdots, 5. \end{aligned}$$

The optimal solution obtained by Algorithm 3.4.1 is $x_1 = 1.000000, x_2 = 0.000000, x_3 = 0.000000, x_4 = 0.000000, x_5 = 0.000000$, and the optimal value is 1.000000.

4.3 Test 3 (The Minus Perturbation Method and the Approximately Dividing Interval Method for $n = 50$ Case)

The baseline case of $n = 50$ was chosen in the following tests, the parameters for Algorithm 3.4.1 were set as follows :

$$\text{the starting point } t_1 = 0.5,$$

$$\delta_1 = 0.8,$$

$$\delta_2 = 0.8,$$

$$\delta_3 = 0.1,$$

$$\beta_1 = 0.9,$$

$$\epsilon_1 = 0.1,$$

$$\epsilon = 10^{-6},$$

$$\mu = 0.1,$$

the $GBLP$ functions were chosen as minus logarithmic functions, i.e.,

$$\phi_k(x) = -\sum_{i=1}^{n+k} \ln x_i.$$

Moreover, μ_{k+1} and ϵ_{k+1}, for $k = 1, 2, \cdots$ were defined as follows :

$$\mu_{k+1} := \begin{cases} (1-\delta_1)\mu_k & \text{when} \quad \mu_k \geq \epsilon, \\ \mu_k & \text{when} \quad \mu_k < \epsilon. \end{cases}$$

$$\epsilon_{k+1} := \begin{cases} (1-\delta_2)\epsilon_k & \text{when} \quad \epsilon_k \geq 10^{-6}, \\ \epsilon_k & \text{when} \quad \epsilon_k < 10^{-6}. \end{cases}$$

The solution process is shown by Tables 4.3.1, 4.3.2, and 4.3.3.

Table 4.3.1 The result of Test 3 for Problem 1

Iteration	Objective Value
1	0.677836
2	0.480994
3	0.479434

Problem 1

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{50} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{50} t^{j-1} x_j \geq \sin t, \ \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ \ j = 1, 2, \cdots, 50. \end{aligned}$$

The optimal value obtained by Algorithm is 0.479434, and the iteration number k is 3.

Table 4.3.2 The result of Test 3 for Problem 2

Iteration	Objective Value
1	0.548778
2	0.863027
3	0.696241
4	0.692564
5	0.692534
6	0.692326
7	0.692281
8	0.692272

Problem 2

$$\begin{aligned} \textit{Minimize} \quad & \sum_{j=1}^{50} \frac{x_j}{j}. \\ \text{Subject to} \quad & \sum_{j=1}^{50} t^{j-1} x_j \geq \frac{1}{2-t}, \ \forall t \in [0,1], \\ & x_j \geq 0, \ j = 1, 2, \cdots, 50. \end{aligned}$$

The optimal value of Problem 2 obtained by Algorithm 3.4.1 is 0.692272, and the iteration number k is 8.

Table 4.3.3 The result of Test 3 for Problem 3

Iteration	Objective Value
1	5.632994
2	1.000001

Problem 3

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{50} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{50} t^{j-1} x_j \geq \frac{1}{1+t^2}, \ \forall t \in [0,1], \\ &\qquad\qquad x_j \geq 0, \ j = 1, 2, \cdots, 50. \end{aligned}$$

The optimal value obtained by Algorithm 3.4.1 is 1.000001, and the iteration number k is 2.

In order to make sure that the solutions generated by Algorithm 3.4.1 are really close to the true optimum, a brute force approach to approximate Problem 1, Problem 2, Problem 3 was taken by dividing the interval $[0,1]$ into 550 evenly space points $\{t(1),t(2),\cdots,t(550)\}$ and replacing the explicit constraints in Problem 1, Problem 2 , and Problem 3 by 550 liner inequalities evaluated at $t = t(1),\cdots,t(550)$, i.e., solve Problem 1, 2, 3 by solving the approximate linear programming problems :

Problem 1'

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{50} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{50} (\frac{i}{550})^{j-1} x_j \geq \sin(\frac{i}{550}), \;\; i = 1,2,\cdots,550, \\ &\qquad\qquad x_j \geq 0, \;\; j = 1,2,\cdots,50. \end{aligned}$$

Problem 2'

$$\begin{aligned} &\textit{Minimize} \quad \sum_{j=1}^{50} \frac{x_j}{j}. \\ &\text{Subject to} \quad \sum_{j=1}^{50} (\frac{i}{550})^{j-1} x_j \geq \frac{1}{2-(\frac{i}{550})}, \;\; i = 1,2,\cdots,550 \\ &\qquad\qquad x_j \geq 0, \;\; j = 1,2,\cdots,50. \end{aligned}$$

Problem 3'

$$\textit{Minimize} \quad \sum_{j=1}^{50} \frac{x_j}{j}.$$

$$\text{Subject to} \quad \sum_{j=1}^{50} \left(\frac{i}{550}\right)^{j-1} x_j \geq \frac{1}{1+\left(\frac{i}{550}\right)^2}, \quad i = 1, 2, \cdots, 550,$$

$$x_j \geq 0, \quad j = 1, 2, \cdots, 50.$$

The above corresponding linear programming problems were solved by a standard linear programming package. The approximated optimum objective values obtained in this way are 0.479430 for Problem 1, 0.693149 for Problem 2, and 1.000000 for Problem 3.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] E. J. Anderson and P. Nash, "Linear programming in infinite-dimensional space", Wiley, Chichester, (1987).

[2] S. C. Fang, "An unconstrained convex programming view to linear programming", *Zeischrift fur Operations Research (ZOR)* Vol 36 (1992), 149-161.

[3] S. C. Fang and R. L. Sheu, "On the generalized path-following methods for linear programming". *Optimization* Vol 30 (1994), 235-249.

[4] S. C. Fang and R. L. Sheu, "Insights into the interior point methods",*ZOR* Vol 136 (1992), 227-257.

[5] S. C. Fang and J. H. S. Tsao, "Linear programming with entropic perturbation", *ZOR - Method and Model of Operations Research* (1993), 171-186.

[6] K. Glashoff and S. A. Gustafson, "Linear optimization and approximation", Springer-Verlag, New York, (1982).

[7] S. A. Gustafson, "On the computational solution of a class of generalized moment problems", *SIAM J. Numerical Analysis* Vol 7 (1970), 343-357.

[8] K. A. Gustafson and K. Kortanek, "Numerical treatment of a class of semi-infinite programming problems", *Naval Research Logistics Quarterly*, Vol 20 (1973), 473-504.

[**9**] R. Hettich and K. Kortanek, "Semi-infinite programming: theory, method and applications". *Technical Report,* College of Business Administration, The University of Iowa, Iowa city, Iowa (1991).

[**10**] H. C. Lai and S. Y. Wu, "On linear semi-infinite programming problems, an algorithm", *Numerical Functional Analysis and Optimization* Vol 13 (1992) 287-304.

[**11**] R. J. Vanderbei, LOQO User's Manual. *Technical Report* SOL 92-05, Dept of Civil Engineering and Operations Research, Princeton University, Princeton, NJ 08544, USA, (1992).

APPENDIX

APPENDIX A

KARUSH-KUHN-TUCKER CONDITIONS

This appendix introduces the $karush-Kuhn-Tucker$ Conditions for the nonlinear programming problem P :

$$
\begin{aligned}
& \textit{Minimize} && f(x) \\
(P) & \\
& \text{Subject to} && g_i(x) \leq 0, \text{ for i=1,}\cdots,k, \\
& && h_i(x) = 0, \text{ for i=1,}\cdots,l, \\
& && x \in S,
\end{aligned}
$$

which has inequality and inequality constraints.

Theorem (Karush-Kuhn-Tucker Conditions)

Let S be a non-empty open set in R^n, and let $f : R^n \longmapsto R$, $g_i : R^n \longmapsto R$ for $i = 1, \cdots, k$, and $h_i : R^n \longmapsto R$ for $i = 1, \cdots, l$, be functions mapping from R^n to R. Consider Problem P to be

$$
\begin{aligned}
&\textit{Minimize} \quad f(x) \\
(P) \qquad & \\
&\text{Subject to} \quad g_i(x) \leq 0, \text{ for i=}1,\cdots,k, \\
&\qquad\qquad\quad\; h_i(x) = 0, \text{ for i=}1,\cdots,l, \\
&\qquad\qquad\quad\; x \in S.
\end{aligned}
$$

Let $\bar{x}$ be a feasible solution and $I = \{i : g_i(\bar{x}) = 0\}$. Suppose that f and g_i for $i \in I$ are differentiable at $\bar{x}$, that g_i for $i \notin I$ is continuous at $\bar{x}$, and that h_i for $i = 1,\cdots,l$ is continuously differentiable at $\bar{x}$. Furthermore, suppose that $\nabla g_i(\bar{x})$ for $i \in I$ and $\nabla h_i(\bar{x})$ for $i = 1,\cdots,l$ are linearly independent. If $\bar{x}$ solves problem P locally, then there exist scalars λ_i for $i \in I$ and y_i, $i = 1,\cdots,l$ such that

$$
\begin{cases}
\nabla f(\bar{x}) + \sum_{i\in I} \lambda_i \nabla g_i(\bar{x}) + \sum_{i=1}^{l} y_i \nabla h_i(\bar{x}) = 0, \\
\lambda_i \geq 0, \text{ for } i \in I.
\end{cases}
$$

In addition to the above assumptions, if g_i for $i \notin I$ is also differentiable at $\bar{x}$, then the Karush-Kuhn-Tucker conditions could be written in the following equivalent form :

$$
\begin{cases}
\nabla f(\bar{x}) + \sum_{i=1}^{k} \lambda_i \nabla g_i(\bar{x}) + \sum_{i=1}^{l} y_i \nabla h_i(\bar{x}) = 0, \\
\lambda_i g_i(\bar{x}) = 0, \text{ for i} = 1,\cdots,k, \\
\lambda_i \geq 0, \text{ for i= } 1,\cdots,k.
\end{cases}
$$

Note that $\lambda = (\lambda_1, \cdots, \lambda_k)^T$ and $y = (y_1, \cdots, y_l)^T$ are called the Lagrangian multiplier vectors.

Example : Find the Karush-Kuhn-Tucker conditions of the problem P_μ which is defined in Chapter II :

$$(P_\mu) \qquad \begin{aligned} &\textit{Minimize} && c^T x + \mu\phi(x), \\ &\text{Subject to} && Ax = b, \\ & && x \geq 0. \end{aligned}$$

Solution : First, we assume $x^*(\mu)$ is the optimal solution of P_μ which can be written as follows :

$$(P_\mu) \qquad \begin{aligned} &\textit{Minimize} && (\sum_{i=1}^{n} c_i x_i) + \mu\phi(x), \\ &\text{Subject to} && -x_i \leq 0, \text{ for i} = 1, \cdots, n, \\ & && b_i - \sum_{j=1}^{n} a_{ij} x_j = 0, \text{ for i} = 1, \cdots, m. \end{aligned}$$

In this case, $f(x) = (\sum_{i=1}^{n} c_i x_i) + \mu\phi(x)$, n, m are with respect to k, l in the above Theorem, $S = R^n$. Furthermore, $g_i(x)$ and $h_i(x)$ are defined as follows :

$$g_i(x) = -x_i, \text{ for i= } 1, \cdots, n,$$

$$h_i(x) = b_i - \sum_{j=1}^{n} a_{ij} x_j, \text{ for i= } 1, \cdots, m.$$

em 2.2.1 $x^*(\mu)_i > 0$, for $i = 1, \cdots, n$, so I is an empty set, and

Bentary set of I, called $I^c = \{1, \cdots, n\}$. $g_i(x) = -x_i$ is continuously

th iable at $x^*(\mu)$, for $i \notin I$ and $h_i(x) = b_i - \sum_{j=1}^{n} a_{ij}x_j$, for $i = 1, \cdots, m$ is

inuously differentiable at $x^*(\mu)$. Moreover, $\nabla h_i(x^*(\mu)) = -[a_{i1}, \cdots, a_{in}]^T$, for $i = 1, \cdots, m$ are linearly independent because of the assumption that

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

which has a full row rank m. Therefore, if $x^*(\mu)$ solves the problem P, then there exist scalars λ_i, $i = 1, \cdots, n$ and $y_i, i = 1, \cdots, m$, such that $x^*(\mu)$ satisfies the following conditions :

$$\begin{cases} \nabla f(x) + \sum_{i=1}^{n} \lambda_i \nabla g_i(x) + \sum_{i=1}^{m} y_i \nabla h_i(x) = 0, \\ \lambda_i g_i(x) = 0, \text{ for i} = 1, \cdots, n, \\ \lambda_i \geq 0, \text{ for i} = 1, \cdots, n, \\ Ax = b, \\ x \geq 0, \end{cases}$$

by the above Theorem.