การแก้ปัญหากำหนดการเชิงเส้นด้วยวิธีจุดภายใน

นายสอาด ม่วงจันทร์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคณนา ภาควิชาคณิตศาสตร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2544

ISBN 974-170-534-4

SOLVING LINEAR PROGRAMMING PROBLEMS
BY THE INTERIOR-POINT METHOD

Mr. Sa-at Moungjun

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Computational Science

Department of Mathematics

Faculty of Science

Chulalongkorn University

Academic Year 2001

ISBN 974-170-534-4

Thesis Title	SOLVING LINEAR PROGRAMMING PROBLEMS BY THE INTERIOR-POINT METHOD
By	Mr. Sa-at Moungjun
Field of Study	Computational Science
Thesis Advisor	Associate Professor Wanida Hemakul, Ph.D.
Thesis Co-advisor	Krung Sinapiromsaran, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master 's Degree

……………………………………………….. Deputy Dean for Administrative Affairs

(Associate Professor Pipat Karntiang, Ph.D.) Acting Dean, Faculty of Science

THESIS COMMITTEE

……………………………………………….. Chairman

(Associate Professor Virasak Ratanasomboon)

……………………………………………….. Thesis Advisor

(Associate Professor Wanida Hemakul, Ph.D.)

……………………………………………….. Thesis Co-advisor

(Krung Sinapiromsaran, Ph.D.)

……………………………………………….. Member

(Assistant Professer Pornchai Satravaha, Ph.D.)

สอาด ม่วงจันทร์ : การแก้ปัญหากำหนดการเชิงเส้นด้วยวิธีจุดภายใน. (SOLVING LINEAR PROGRAMMING PROBLEMS BY THE INTERIOR-POINT METHOD) อ. ที่ปรึกษา : รศ. ดร. วนิดา เหมะกุล, อ.ที่ปรึกษาร่วม : อ.ดร. กรุง สินอภิรมย์สราญ ; จำนวนหน้า 71 หน้า. ISBN 974-170-534-4.

งานวิจัยนี้รวบรวมขั้นตอนวิธีในการแก้ปัญหากำหนดการเชิงเส้นด้วยวิธีจุดภายใน 3 ขั้นตอนวิธี คือ ขั้นตอนวิธีโพรเจกทีฟสเกลลิ่งของคาร์มาร์คาร์, ขั้นตอนวิธีไพรมอลแอฟฟีนสเกลลิ่ง และขั้นตอนวิธีไพรมอลดูอัล เราสร้างโปรแกรมด้วยภาษาซีบวกบวกบนระบบปฏิบัติการวินโดว์

ผลการทดสอบโปรแกรมกับปัญหาขนาดเล็กและปัญหาที่เก็บอยู่ในแฟ้มข้อมูลรูปแบบเอ็มพีเอส (ระบบโปรแกรมมิ่งเชิงคณิตศาสตร์) พบว่าถ้าปัญหาขนาดเล็ก โปรแกรมของเราใช้จำนวนรอบในการประมวลผลมากกว่าโปรแกรมที่ใช้วิธีซิมเพลกซ์ ถ้าปัญหาขนาดใหญ่ โปรแกรมของเราใช้จำนวนรอบในการประมวลผลน้อยกว่าโปรแกรมที่ใช้วิธีซิมเพลกซ์

ภาควิชา......คณิตศาสตร์ .... ลายมือชื่อนิสิต..

สาขาวิชา.....วิทยาการคณนา.................... ลายมือชื่ออาจารย์ที่ปรึกษา......................................

ปีการศึกษา…2544……………………….. ลายมือชื่ออาจารย์ที่ปรึกษาร่วม................................

4272420023 : MAJOR COMPUTATIONAL SCIENCE

KEY WORD: LINEAR PROGRAMMING/ THE INTERIOR-POINT METHOD

SA-AT MOUNGJUN : SOLVING LINEAR PROGRAMMING PROBLEMS BY THE INTERIOR-POINT METHOD. THESIS ADVISOR : ASSOC. PROF. WANIDA HEMAKUL, Ph.D., THESIS COADVISOR : KRUNG SINAPIROMSARAN, Ph.D., 71 pp. ISBN 974-170-534-4.

This research aims to survey methods of solving linear programming problems by the interior-point method in 3 approaches: Karmarkar's projective scaling, the primal affine scaling and the primal-dual algorithm. We construct a program by C++ language on the Windows operating system.

In our results, we tested our program with our small problems and problems in MPS (Mathematical Programming System) files. Typically, if problem has a small size then our program uses iteration numbers of processing more than program which bases on the simplex method. If problem has a large-scale size then our program uses iteration numbers of processing less than program which bases on the simplex method.

Department.... Mathematics..................... Student’s signature...

Field of study. ...Computational Science... Advisor’s signature...

Academic year …2001...........…………..... Co-advisor’s signature.......................................

ACKNOWLEDGEMENTS

I am greatly indebted to Associate Professor Dr. Wanida Hemakul, my thesis advisor and Dr. Krung Sinapiromsaran, my thesis co-advisor, for their untired offering me some thoughtful and helpful advice in preparing and writing this thesis.

Furthermore, I would like to thank all my instructors for their valuable instructions and I also thank the Feed Specialty Company Limited, Thailand for providing data of feed formulation of pig starter diet. Finally, I would like to express my sincere gratitude to my parents for their love, hearty encouragement and unselfish sacrifice during my study.

TABLE OF CONTENTS

PAGE

ABSTRACT IN THAI iv

ABSTRACT IN ENGLISH v

ACKNOWLEDGEMENTS vi

CHAPTER

I INTRODUCTION 1

II THE INTERIOR-POINT METHOD 9

-KARMARKAR'S PROJECTIVE SCALING ALGORITHM 9

-THE PRIMAL AFFINE SCALING ALGORITHM 22

-THE PRIMAL-DUAL ALGORITHM 30

III IMPLEMENTATIONS, NUMERICAL RESULTS

AND CONCLUSIONS 43

REFERENCES 55

APPENDICES

APPENDIX A 58

APPENDIX B 62

VITA 71

CHAPTER I

INTRODUCTION

This chapter describes the basic background on linear programming problems, the duality theory and linear programming algorithms such as the simplex method and the interior-point method.

1.1 Linear Programming Problems

A linear programming problem is an optimization problem with a linear objective function and linear constraints. It arises frequently in Economics, Engineering and Science areas such as the network flow problem, the scheduling problem and the assignment problem [2,8].

1.1.1 Standard Form

There are many different ways to represent a linear programming problem. In this thesis, we are interested in solving linear programming problems using the following standard form:

$$\begin{array}{ll} \text{minimize} & c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{subject to} & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \\ & x_1, x_2, \dots, x_n \geq 0 \end{array} \tag{1.1}$$

where $c_1, c_2, \ldots, c_n$ are the cost coefficients for nonnegative variables $x_1, x_2, \ldots, x_n$, respectively, a_{ij}, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, are the coefficients of linear constraints and $b_1, b_2, \ldots, b_m$ are the right-hand-side values.

A linear programming problem is to find a specific nonnegative value for each decision variable such that the objective function achieves its minimum while all the constraints are satisfied.

If we denote $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T, \mathbf{c} = (c_1, c_2, \ldots, c_n)^T, \mathbf{b} = (b_1, b_2, \ldots, b_m)^T$ and $\mathbf{A}$ is a matrix $[a_{ij}]$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$ then the above linear programming problem can be written in the matrix notation as follows:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0. \end{aligned} \tag{1.2}$$

In this thesis, we assume that $\mathbf{A}$ has full row rank, i.e., all rows of $\mathbf{A}$ are linearly independent.

In general, we can convert any linear programming problem into this standard form. For the maximization of the objective function, we multiply the objective function by -1 and change the maximization problem into the minimization problem. To recover the objective solution of the original problem, we multiply the objective value by -1, that is

$$maximize\ \mathbf{c}^T\mathbf{x} = -minimize\ (-\mathbf{c}^T\mathbf{x}).$$

All linear inequality constraints in the $\leq$ or $\geq$ form can be converted to the equality constraints by adding a slack variable or subtracting a surplus variable. If the k^{th} constraint is of the form

$$a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n \leq b_k,$$

then we add a slack variable $s_k \geq 0$ to the left-hand-side of the inequality constraint to get

$$a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n + s_k = b_k.$$

Similarly, if the k^{th} constraint is of the form

$$a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n \geq b_k,$$

then we subtract the left-hand-side by a surplus variable $e_k \geq 0$

$$a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n - e_k = b_k.$$

If $x_j \geq l$, we replace x_j by $\overline{x}_j + l$ where $\overline{x}_j \geq 0$ in all constraints. Similarly, If $x_j \leq u$, we can replace x_j by $u - \overline{x}_j$ where $\overline{x}_j \geq 0$. And if the j^{th} variable is unrestricted, we replace x_j by $x_j^{'} - x_j^{''}$ where $x_j^{'}$ and $x_j^{''} \geq 0$.

1.1.2 Feasible Domain, Optimal Solution, Polyhedral Set, Convex Set and Extreme Point

Consider a linear programming problem in its standard form (1.2), we define $I\!F = \{\mathbf{x} \in I\!R^n | \ \mathbf{Ax} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}\}$ to be the *feasible domain* or *feasible region* of the linear programming problem. If $I\!F$ is not empty, then the linear programming problem is said to be *consistent*. For a consistent linear programming problem with a feasible solution $\mathbf{x}^* \in I\!F$, if $\mathbf{c}^T\mathbf{x}^*$ attains the minimum value of the objective function over the feasible domain $I\!F$, then we say $\mathbf{x}^*$ is an *optimal solution* to the linear programming problem.

A fundamental geometric entity occurring in linear optimization is the *hyperplane*

$$H = \{\mathbf{x} \in I\!R^n | \ \mathbf{a}^T\mathbf{x} = \beta\}$$

whose description involves a nonzero n-dimensional column vector $\mathbf{a}$ and a scalar β. A hyperplane separates the whole space into two *closed halfspaces*

$$H_L = \{\mathbf{x} \in I\!R^n |\ \mathbf{a}^T\mathbf{x} \leq \beta\}$$

and

$$H_U = \{\mathbf{x} \in I\!R^n |\ \mathbf{a}^T\mathbf{x} \geq \beta\}.$$

We define a *polyhedral set* or *polyhedron* to be a set formed by the intersection of a finite number of closed halfspaces. If the intersection is not empty and bounded, it is called a *polytope.* For a linear programming problem in its standard form, if we denote $\mathbf{A}_i$ to be the i^{th} row of the constraint matrix $\mathbf{A}$ and b_i the i^{th} element of the right-hand-side vector $\mathbf{b}$, then we have m-hyperplanes

$$H_i = \{\mathbf{x} \in I\!R^n |\ \mathbf{A}_i\mathbf{x} = b_i\}, \quad \text{for } i = 1, 2, \ldots, m$$

and the feasible domain $I\!F = \{\mathbf{x} \in I\!R^n |\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$ becomes the intersection of these hyperplanes and the first orthant of $I\!R^n$. Notice that each hyperplane H_i is an intersection of two closed halfspaces $(H_i)_L$ and $(H_i)_U$ and the first orthant of $I\!R^n$ is the intersection of n closed halfspaces $\{\mathbf{x} \in I\!R^n |\ x_i \geq 0\}$ for $i = 1, 2, \ldots, n$. Hence the feasible domain $I\!F$ is a polyhedral set.

We define $\mathbf{S} \subset I\!R^n$ is a *convex set* if $\mathbf{x}$ and $\mathbf{y}$ are in $\mathbf{S}$, then the line segment connecting $\mathbf{x}$ and $\mathbf{y}$ is also in $\mathbf{S}$. Every set defined by a system of linear constraints is a convex set. Hence $I\!F = \{\mathbf{x} \in I\!R^n |\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \ \mathbf{x} \geq \mathbf{0}\}$ is a convex set.

A point $\mathbf{x}$ in a convex set F is said to be an *extreme point* (or a vertex) of F if $\mathbf{x}$ is not a convex combination of any other two distinct points in F. In other words, an extreme point is a point that does not lie strictly within a line segment connecting two other points of the convex set.

A function f is convex on a convex set $\mathbf{S}$ if it satisfies

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$$

for all $0 \leq \alpha \leq 1$ and for all $x, y \in \mathbf{S}$. We say that a function f is strictly convex if

$$f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$$

for all $x \neq y$ and $0 < \alpha < 1$ where $x, y \in \mathbf{S}$.

Theorem 1.1.1. Fundamental Theorem of Linear Programming [4]. For a consistent linear programming problem in its standard form with a feasible domain $\mathbb{F} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, the minimum objective value of $\mathbf{c}^T\mathbf{x}$ over $\mathbb{F}$ is either unbounded below or is achievable at least at one extreme point of $\mathbb{F}$.

1.2 The Duality Theory

The notion of duality is one of the most important concepts in linear programming problems. Basically, associated with each linear programming problem (we call it the *primal problem*), defined by the constraint matrix $\mathbf{A}$, the right-hand-side $\mathbf{b}$, and the cost vector $\mathbf{c}$, there is a corresponding linear programming problem (we call it the *dual problem*) which is constructed by the same set of data $\mathbf{A}, \mathbf{b}$, and $\mathbf{c}$. A pair of the primal and dual problems are closely related. The interesting relation between the primal and the dual problem reveals important insights into solving linear programming problems.

Consider a linear programming problem in its standard form

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array} \qquad \text{(P)}$$

and the corresponding dual problem will have the form

$$\begin{aligned} \text{maximize} \quad & \mathbf{b}^T\mathbf{w} \\ \text{subject to} \quad & \mathbf{A}^T\mathbf{w} \leq \mathbf{c} \\ & \mathbf{w} \text{ unrestricted.} \end{aligned} \tag{D}$$

We call a minimization problem (P), as the *primal problem* and a maximization problem (D), as the *dual problem.* We can write the dual problem in its standard form as follows:

$$\begin{aligned} \text{maximize} \quad & \mathbf{b}^T\mathbf{w} \\ \text{subject to} \quad & \mathbf{A}^T\mathbf{w} + \mathbf{s} = \mathbf{c} \\ & \mathbf{s} \geq \mathbf{0}, \ \mathbf{w} \text{ unrestricted.} \end{aligned}$$

Theorem 1.2.1. Weak Duality Theorem [4,7,10]. Let $\mathbf{x}$ be a feasible point for the primal problem and let $(\mathbf{w}, \mathbf{s})$ be a feasible point for the dual problem. Then

$$\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{w}.$$

Corollary 1.2.2 [4,7,10]. If $\mathbf{x}^*$ is a feasible solution for the primal problem, $(\mathbf{w}^*, \mathbf{s}^*)$ is a feasible solution for the dual problem and $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{w}^*$, then $\mathbf{x}^*$ is an optimal solution for the primal problem and $(\mathbf{w}^*, \mathbf{s}^*)$ is an optimal solution for the dual problem.

Corollary 1.2.3 [4,7,10]. If the primal problem is unbounded below, then the dual problem is infeasible. If the dual problem is unbounded above, then the primal problem is infeasible.

Theorem 1.2.4. Strong Duality Theorem [4,7,10].

1. If either the primal or the dual problem has an optimal solution, then they achieve the same optimal value.

2. If either problem has an unbounded objective value, then the other has no feasible solution.

Theorem 1.2.5. Karush-Kuhn-Tucker Optimality Conditions (K-K-T Conditions) [4,7,10]. Given a linear programming problem in its standard form, $\mathbf{x}^*$ is an optimal solution for the primal problem if and only if, there exist $(\mathbf{w}^*, \mathbf{s}^*)$ such that

1. $\mathbf{A}\mathbf{x}^* = \mathbf{b}, \mathbf{x}^* \geq \mathbf{0}$ (primal feasibility)

2. $\mathbf{A}^T\mathbf{w}^* + \mathbf{s}^* = \mathbf{c},\ \mathbf{s}^* \geq \mathbf{0}$ (dual feasibility)

3. $(\mathbf{x}^*)^T\mathbf{s}^* = \mathbf{0}$ (complementary slackness).

In this case, $(\mathbf{w}^*, \mathbf{s}^*)$ is an optimal solution for the dual problem.

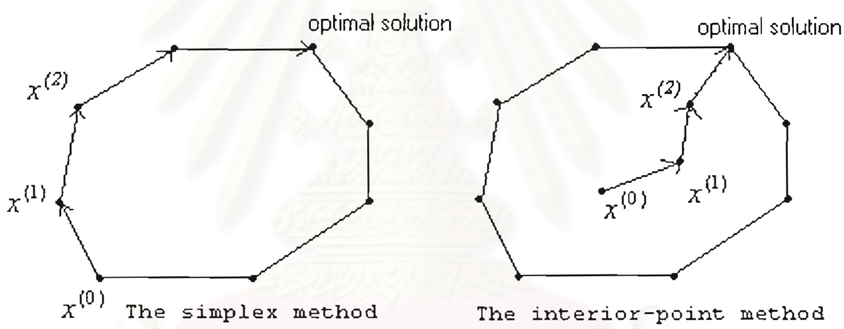
1.3 The Simplex Method

The simplex method was proposed in the summer of 1949 by G.B. Dantzig. It is the most widely used method for solving a linear programming problem. When the simplex algorithm is applied to a nondegenerated problem, it moves from one extreme point to another. If the current feasible solution is not an optimal solution, the method selects and moves to an adjacent extreme point that has a better objective value. By repeating this search, the simplex method will eventually achieve an optimal value of the objective function.

1.4 The Interior-Point Method

The interior-point method is the new method for solving linear programming problems which is bounded by the smaller computational complexity for the large scale linear programming problem. The interior-point method starts at an initial point within a feasible region. Then, at each iteration the interior-point method searches for the direction which improve the objective value and satisfy linear constraints, until achieve an optimal solution.

Figure 1 : The simplex method and the interior-point method



We are interested in three algorithms of the interior-point method, the Karmarkar's projective scaling, the primal affine scaling and the primal-dual algorithms which will be described in Chapter II.

In Chapter III, we describe the design of our software and our numerical results which we test with a small tested problems and MPS files.

The MPS files and small tested problems can be found in Appendix A and B.

CHAPTER II

THE INTERIOR-POINT METHOD

This chapter describes Karmarkar's projective scaling algorithm, the primal affine scaling algorithm and the primal-dual algorithm. It is known that Karmarkar's projective scaling algorithm and the primal-dual algorithm are the polynomial-time algorithms while the primal affine scaling algorithm has not been proved to be a polynomial-time algorithm yet.

2.1 Karmarkar's Projective Scaling Algorithm

Karmarkar noticed two fundamental insights, assuming the feasible domain is a polytope.

1. If the current interior solution is near the center of the polytope, then it makes sense to move in the direction of the steepest descent of objective function to achieve a minimum value.

2. Without changing the problem, an appropriate transformation can be applied to the solution space such that the current interior solution is placed near the center of the transformed solution space.

With these two fundamental insights, the basic strategy of Karmarkar's projective scaling algorithm is straightforward. We take an interior solution, transform the solution so as to place the current solution near the center of the polytope in the transformed space, and then move it in the direction of steepest descent. Then take the inverse transformation to map the improved solution back to the original solution space as a new interior solution. We repeat the process until an optimal solution is obtained.

2.1.1 Karmarkar's Standard Form

Following the basic strategy of the projective scaling, Karmarkar's algorithm has a preferred standard form for linear programming problem:

$$\text{minimize} \quad \mathbf{c}^T\mathbf{x} \tag{2.1a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{0} \tag{2.1b}$$

$$\mathbf{e}^T\mathbf{x} = 1, \quad \mathbf{x} \geq \mathbf{0} \tag{2.1c}$$

where $\mathbf{A}$ is an $m \times n$ dimensional matrix of full row rank, $\mathbf{e}^T = (1, 1, \ldots, 1)$ is an $n-$vector of all ones and $\mathbf{c}, \mathbf{x} \in I\!R^n$.

A feasible solution vector $\mathbf{x}$ of the problem (2.1) is defined to be an *interior solution* if every variable x_i is strictly positive. Note from (2.1c) that the feasible domain is a bounded set, hence it becomes a polytope. There are two assumptions for the Karmarkar's algorithm.

(A1) $\mathbf{Ae} = \mathbf{0}$, so that $\mathbf{x}^{(0)} = \frac{\mathbf{e}}{n} = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)^T$ is an initial interior solution.

(A2) The optimal objective value of problem (2.1) is zero.

2.1.2 The Simplex Structure

Expression (2.1c) defines a regular polygon in the n-dimensional Euclidean space, namely

$$\Delta = \{\mathbf{x} \in I\!R^n | \sum_{i=1}^{n} x_i = 1, \; x_i \geq 0\}. \tag{2.2}$$

For example, a unit simplex in $I\!R^1$ is $\Delta = \{1\}$ which is the singleton.

In $I\!R^2$, Δ is the line segment between the points (0,1) and (1,0).

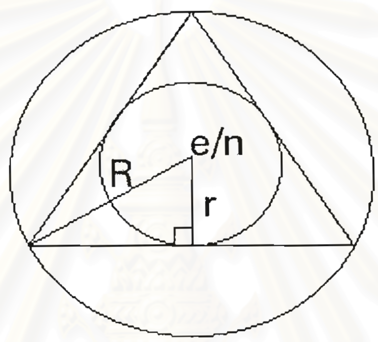
In $I\!R^3, \Delta$ is the triangular area formed by (0,0,1), (0,1,0) and (1,0,0).

In $I\!R^4, \Delta$ is the pyramid with vertices at (0,0,0,1), (0,0,1,0), (0,1,0,0) and (1,0,0,0).

In $I\!R^n$, Δ has n vertices, $\begin{pmatrix} n \\ 2 \end{pmatrix}$ edges, $\begin{pmatrix} n \\ n-1 \end{pmatrix}$ facets and its center at $\mathbf{e}/n$.

We can show that the radius of the smallest circumscribing spheroid of Δ is given by $R = \frac{\sqrt{n-1}}{\sqrt{n}}$ and the radius of the largest inscribing in Δ is given by $r = \frac{1}{\sqrt{n(n-1)}}$.

Figure 2 : The simplex structure



2.1.3 Projective Transformation on the Simplex

Let $\overline{\mathbf{x}}$ be an interior point of Δ, i.e., $\overline{x}_j > 0$ for $j = 1, 2, \ldots, n$ with $\sum_{j=1}^{n} \overline{x}_j = 1$ and $\overline{\mathbf{X}}$ be an $n \times n$ diagonal matrix

$$\overline{\mathbf{X}} = diag(\overline{\mathbf{x}}) = \begin{bmatrix} \overline{x}_1 & 0 & \ldots & 0 \\ 0 & \overline{x}_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \overline{x}_n \end{bmatrix}.$$

It is obvious that matrix $\overline{\mathbf{X}}$ is a nonsingular matrix and its inverse matrix $\overline{\mathbf{X}}^{-1}$ is also a diagonal matrix but with $1/\overline{x}_i$ as its i^{th} diagonal elements for $i = 1, 2, \ldots, n$.
We define a projective transformation $\mathbf{T}_{\overline{\mathbf{x}}}$ from Δ to Δ such that

$$\mathbf{T}_{\overline{\mathbf{x}}}(\mathbf{x}) = \frac{\overline{\mathbf{X}}^{-1}\mathbf{x}}{\mathbf{e}^T\overline{\mathbf{X}}^{-1}\mathbf{x}}, \text{ for } \mathbf{x} \in \Delta.$$

2.1.4 Karmarkar's Projective Scaling Algorithm

Consider a linear programming problem in Karmarkar's standard form (2.1), its feasible domain is a polytope formed by the intersection of the null space of the constraint matrix $\mathbf{A}$, i.e., $\{\mathbf{x}|\mathbf{A}\mathbf{x} = \mathbf{0}\}$ and the simplex Δ in $I\!R^n$. Let $\overline{\mathbf{x}} > \mathbf{0}$ be an interior feasible solution, then the projective transformation $\mathbf{T}_{\overline{\mathbf{x}}}$ maps $\mathbf{x} \in \Delta$ to

$$\mathbf{y} = \mathbf{T}_{\overline{\mathbf{x}}}(\mathbf{x}) = \frac{\overline{\mathbf{X}}^{-1}\mathbf{x}}{\mathbf{e}^T\overline{\mathbf{X}}^{-1}\mathbf{x}}$$

and we can find $\mathbf{x}$ in terms of $\mathbf{y}$ by

$$\mathbf{x} = \mathbf{T}_{\overline{\mathbf{x}}}^{-1}(\mathbf{y}) = \frac{\overline{\mathbf{X}}\mathbf{y}}{\mathbf{e}^T\overline{\mathbf{X}}\mathbf{y}} .$$

Then we have a corresponding problem in the transformed solution space as follows:

$$\begin{aligned} \text{minimize} \quad & (\mathbf{c}^T\overline{\mathbf{X}}\mathbf{y})/(\mathbf{e}^T\overline{\mathbf{X}}\mathbf{y}) \\ \text{subject to} \quad & \mathbf{A}\overline{\mathbf{X}}\mathbf{y} = \mathbf{0} \\ & \mathbf{e}^T\mathbf{y} = 1,\ \mathbf{y} \geq \mathbf{0}. \end{aligned} \tag{2.3}$$

Note that in the problem (2.3), the image of $\overline{\mathbf{x}}$, i.e., $\overline{\mathbf{y}} = \mathbf{T}_{\overline{\mathbf{x}}}(\overline{\mathbf{x}}) = \mathbf{e}/n$ becomes a feasible solution and is at the center of the simplex Δ. If we denote the constraint matrix by

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}\overline{\mathbf{X}} \\ \mathbf{e}^T \end{bmatrix}$$

then any direction $\mathbf{d} \in I\!R^n$ in the null space of matrix $\mathbf{B}$, i.e., $\mathbf{B}\mathbf{d} = \mathbf{0}$, is a feasible direction of movement for $\overline{\mathbf{y}}$. Since the distance from the center of Δ to boundary is given by the radius $r = \frac{1}{\sqrt{n(n-1)}}$. Therefore, if we denote the norm of $\mathbf{d}$ by $||\mathbf{d}||$ and $0 \leq \alpha < 1$, then

$$\mathbf{y}_\alpha^{new} = \overline{\mathbf{y}} + \alpha r\left(\frac{\mathbf{d}}{||\mathbf{d}||}\right) \tag{2.4}$$

remains a new interior solution to the problem (2.3) and its inverse image

$$\mathbf{x}_\alpha^{new} = \mathbf{T}_{\overline{\mathbf{x}}}^{-1}(\mathbf{y}_\alpha^{new}) = \frac{\overline{\mathbf{X}}\mathbf{y}_\alpha{}^{new}}{\mathbf{e}^T\overline{\mathbf{X}}\mathbf{y}_\alpha{}^{new}}$$

becomes a new interior solution to the original problem (2.1). Since $r > \frac{1}{n}$ we can replace r in equation (2.4) by

$$\mathbf{y}_{\alpha}^{new} = \overline{\mathbf{y}} + \frac{\alpha}{n}\left(\frac{\mathbf{d}}{\|\mathbf{d}\|}\right)$$

for $0 \leq \alpha \leq 1$, to obtain a new interior feasible solution.

After determining the structure of the feasible directions in the transformed space, we want to find a good feasible direction that eventually leads to an optimal solution. Since $\overline{\mathbf{y}}$ is at the center of Δ, it makes sense to move along the steepest descent of the objective function. Although the objective function of the problem (2.3) is a fractional linear function. Karmarkar pointed out that the linear numerator function $\mathbf{c}^T\overline{\mathbf{X}}\mathbf{y}$ could be a good indication of the reduction of the objective function. Therefore, we take its negative gradient which is $-\mathbf{c}^T\overline{\mathbf{X}}$, or equivalent to $-\overline{\mathbf{X}}\mathbf{c}$, as a good candidate. In order to keep feasibility, we project the negative gradient into the null space of the constraint matrix $\mathbf{B}$. We have the projected negative gradient by the following formula

$$\mathbf{d} = -\left[\mathbf{I} - \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}\right]\overline{\mathbf{X}}\mathbf{c}. \tag{2.5}$$

Here we provide an iterative procedure for the implementation of Karmarkar's algorithm.

Step 1. (initialization) :

Set $k = 0, \mathbf{x}^{(0)} = \mathbf{e}/n$ where n is a number of variables in Karmarkar's standard form, choose α to lie between 0 and 1, ϵ to be a small positive integer.

Step 2. (optimality check) :

If $\mathbf{c}^T\mathbf{x}^{(k)} \leq \epsilon$ then STOP with an *optimal solution* $\mathbf{x}^* = \mathbf{x}^{(k)}$ else go to Step 3.

Step 3. (find a better direction):

$$\begin{aligned}
\overline{\mathbf{X}} &= diag(\mathbf{x}^{(k)}) \\
\mathbf{B} &= \begin{bmatrix} \mathbf{A}\overline{\mathbf{X}} \\ \mathbf{e}^T \end{bmatrix} \\
\mathbf{d}^{(k)} &= -\left[\mathbf{I} - \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}\right]\overline{\mathbf{X}}\mathbf{c}
\end{aligned}$$

Step 4. (find a new solution) :

$$\mathbf{y}^{(k+1)} = \frac{\mathbf{e}}{n} + \frac{\alpha}{n}\left(\frac{\mathbf{d}^{(k)}}{\|\mathbf{d}^{(k)}\|}\right)$$

$$\mathbf{x}^{(k+1)} = \frac{\overline{\mathbf{X}}\mathbf{y}^{(k+1)}}{\mathbf{e}^T\overline{\mathbf{X}}\mathbf{y}^{(k+1)}}$$

Set $k = k + 1$; go to Step 2.

2.1.5 Converting to Karmarkar's Standard Form

Consider a linear programming problem in its standard form

$$\begin{aligned}
\text{minimize} \quad & \mathbf{c}^T\mathbf{x} \\
\text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\
& \mathbf{x} \geq \mathbf{0}.
\end{aligned} \tag{2.6}$$

Our objective is to convert this problem into Karmarkar's standard form, while satisfying the assumption (A1) and (A2).

The key feature of Karmarkar's standard form is the simplex structure. Thus we want to regularize problem (2.6) by adding a bounding constraint

$$\sum_{j=1}^{n} x_j \leq Q$$

for some positive integer Q derived from the feasibility and optimality considerations. By introducing a slack variable x_{n+1}, we have a new linear programming problem:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{e}^T\mathbf{x} + x_{n+1} = Q \\ & \mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0. \end{aligned} \tag{2.7}$$

In order to keep the matrix structure of $\mathbf{A}$ undisturbed for sparsity manipulation, we introduce a new variable $x_{n+2} = 1$ and rewrite the problem (2.7) as

$$\text{minimize} \quad \mathbf{c}^T\mathbf{x} \tag{2.8a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} - \mathbf{b}x_{n+2} = \mathbf{0} \tag{2.8b}$$

$$\mathbf{e}^T\mathbf{x} + x_{n+1} - Qx_{n+2} = 0 \tag{2.8c}$$

$$\mathbf{e}^T\mathbf{x} + x_{n+1} + x_{n+2} = Q + 1 \tag{2.8d}$$

$$\mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0,\ x_{n+2} \geq 0. \tag{2.8e}$$

To normalize (2.8d) for the required simplex structure, we can apply the transformation $x_j = (Q+1)\acute{x}_j$, for $j = 1, 2, \ldots, n+2$, to the problem (2.8). In this way, we have an equivalent linear programming problem

$$\begin{aligned} \text{minimize} \quad & (Q+1)\mathbf{c}^T\acute{\mathbf{x}} \\ \text{subject to} \quad & \mathbf{A}\acute{\mathbf{x}} - \mathbf{b}\acute{x}_{n+2} = \mathbf{0} \\ & \mathbf{e}^T\acute{\mathbf{x}} + \acute{x}_{n+1} - Q\acute{x}_{n+2} = 0 \\ & \mathbf{e}^T\acute{\mathbf{x}} + \acute{x}_{n+1} + \acute{x}_{n+2} = 1 \\ & \acute{\mathbf{x}} \geq \mathbf{0},\ \acute{x}_{n+1} \geq 0,\ \acute{x}_{n+2} \geq 0. \end{aligned} \tag{2.9}$$

Problem (2.9) is now in Karmarkar's standard form. In order to satisfy assumption (A1), we may introduce an artificial variable $\acute{x}_{n+3}$ with a large cost coefficient M

and consider the following problem

$$\begin{aligned} \text{minimize} \quad & (Q+1)\mathbf{c}^T\acute{\mathbf{x}} + M\acute{x}_{n+3} \\ \text{subject to} \quad & \mathbf{A}\acute{\mathbf{x}} - \mathbf{b}\acute{x}_{n+2} - [\mathbf{Ae} - \mathbf{b}]\acute{x}_{n+3} = \mathbf{0} \\ & \mathbf{e}^T\acute{\mathbf{x}} + \acute{x}_{n+1} - Q\acute{x}_{n+2} - (n+1-Q)\acute{x}_{n+3} = 0 \\ & \mathbf{e}^T\acute{\mathbf{x}} + \acute{x}_{n+1} + \acute{x}_{n+2} + \acute{x}_{n+3} = 1 \\ & \acute{\mathbf{x}} \geq \mathbf{0},\ \acute{x}_{n+1} \geq 0,\ \acute{x}_{n+2} \geq 0,\ \acute{x}_{n+3} \geq 0. \end{aligned} \tag{2.10}$$

Note that $\acute{\mathbf{x}} = \mathbf{e}/(n+3)$ is clearly initial interior feasible solution to the problem (2.10). Moreover, at an optimal, we would have $\acute{x}_{n+3} = 0$.

To ensure the second assumption of Karmarkar's standard form, we convert problem (2.6) to have the optimal objective value is zero. We consider the dual problem of the standard form

$$\begin{aligned} \text{maximize} \quad & \mathbf{b}^T\mathbf{w} \\ \text{subject to} \quad & \mathbf{A}^T\mathbf{w} + \mathbf{s} = \mathbf{c} \\ & \mathbf{s} \geq \mathbf{0} \end{aligned} \tag{2.11}$$

where $\mathbf{s}$ is n-dimensional vector, $\mathbf{w}$ is an m-dimensional vector and unrestricted.

By the strong duality theorem, we know that $\mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{w} = 0$ when the problem is optimal, then we can write a new linear programming problem by

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{w} \\ \text{subject to} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{A}^T\mathbf{w} + \mathbf{s} = \mathbf{c} \\ & \mathbf{x}, \mathbf{s} \geq \mathbf{0} \end{aligned} \tag{2.12}$$

and replace $\mathbf{w} = \mathbf{w}' - \mathbf{w}''$ where $\mathbf{w}', \mathbf{w}'' \geq 0$

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} - \mathbf{b}^T(\mathbf{w}' - \mathbf{w}'') \\ \text{subject to} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{A}^T(\mathbf{w}' - \mathbf{w}'') + \mathbf{s} = \mathbf{c} \\ & \mathbf{x}, \mathbf{s}, \mathbf{w}', \mathbf{w}'' \geq \mathbf{0}. \end{aligned} \tag{2.13}$$

So that the problem (2.13) satisfies the second assumption of Karmarkar's standard form and then we can convert the problem (2.13) to the problem (2.10) which satisfies two assumptions of Karmarkar's standard form.

Example 1.1

$$\begin{aligned} \text{Minimize} \quad & -8x_1 - 10x_2 \\ \text{subject to} \quad & 2x_1 + x_2 + x_3 = 50 \\ & x_1 + 2x_2 + x_4 = 70 \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned} \tag{2.14}$$

In this case,

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 50 \\ 70 \end{bmatrix} \text{ and } \mathbf{c} = \begin{bmatrix} -8 \\ -10 \\ 0 \\ 0 \end{bmatrix}.$$

Since problem (2.14) is not in Karmarkar's standard form, we can convert problem (2.14) to Karmarkar's standard form. We start by consider the dual problem of problem (2.14)

$$\begin{aligned} \text{maximize} \quad & 50w_1 + 70w_2 \\ \text{subject to} \quad & 2w_1 + w_2 + s_1 = -8 \\ & w_1 + 2w_2 + s_2 = -10 \\ & w_1 + s_3 = 0 \\ & w_2 + s_4 = 0 \\ & s_1, s_2, s_3, s_4 \geq 0 \\ & w_1, w_2 \ \text{unrestricted}. \end{aligned}$$

Then we convert the above problem to the form of problem (2.12) as follows:

$$\begin{aligned}
\text{minimize} \quad & -8x_1 - 10x_2 - [50w_1 + 70w_2] \\
\text{subject to} \quad & 2x_1 + x_2 + x_3 = 50 \\
& x_1 + 2x_2 + x_4 = 70 \\
& 2w_1 + w_2 + s_1 = -8 \\
& w_1 + 2w_2 + s_2 = -10 \\
& w_1 + s_3 = 0 \\
& w_2 + s_4 = 0 \\
& x_1, x_2, x_3, x_4, s_1, s_2, s_3, s_4 \geq 0 \\
& w_1, w_2 \text{ unrestricted.}
\end{aligned}$$

Since w_1 and w_2 are not strictly positive variables, we replace $w_1 = (w_1^{'} - w_1^{''})$ and $w_2 = (w_2^{'} - w_2^{''})$ where $w_1^{'}, w_1^{''}, w_2^{'}, w_2^{''} \geq 0$.

$$\begin{aligned}
\text{minimize} \quad & -8x_1 - 10x_2 - \left[50(w_1^{'} - w_1^{''}) + 70(w_2^{'} - w_2^{''})\right] \\
\text{subject to} \quad & 2x_1 + x_2 + x_3 = 50 \\
& x_1 + 2x_2 + x_4 = 70 \\
& 2(w_1^{'} - w_1^{''}) + (w_2^{'} - w_2^{''}) + s_1 = -8 \\
& (w_1^{'} - w_1^{''}) + 2(w_2^{'} - w_2^{''}) + s_2 = -10 \\
& (w_1^{'} - w_1^{''}) + s_3 = 0 \\
& (w_2{'} - w_2^{''}) + s_4 = 0 \\
& x_1, x_2, x_3, x_4, w_1^{'}, w_1^{''}, w_2^{'}, w_2^{''}, s_1, s_2, s_3, s_4 \geq 0.
\end{aligned} \tag{2.15}$$

Let

$$\widehat{\mathbf{A}} = \begin{bmatrix} 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & -2 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & -1 & -2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}, \widehat{\mathbf{b}} = \begin{bmatrix} 50 \\ 70 \\ -8 \\ -10 \\ 0 \\ 0 \end{bmatrix},$$

$$\widehat{\mathbf{x}}^T = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & w_1' & w_2' & w_1'' & w_2'' & s_1 & s_2 & s_3 & s_4 \end{bmatrix},$$

and

$$\widehat{\mathbf{c}} = \begin{bmatrix} -8 & -10 & 0 & 0 & -50 & -70 & 50 & 70 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

then, we add a bounding constraint

$$\mathbf{e}^T\widehat{\mathbf{x}} \leq Q$$

for some positive integer Q. By introducing a slack variable $\widehat{x}_{13}$, we have

$$\begin{aligned} \text{minimize} \quad & \widehat{\mathbf{c}}^T\widehat{\mathbf{x}} \\ \text{subject to} \quad & \widehat{\mathbf{A}}\widehat{\mathbf{x}} = \widehat{\mathbf{b}} \\ & \mathbf{e}^T\widehat{\mathbf{x}} + \widehat{x}_{13} = Q \\ & \widehat{\mathbf{x}} \geq \mathbf{0}, \widehat{x}_{13} \geq 0. \end{aligned}$$

We introduce a new variable $\widehat{x}_{14} = 1$ and rewrite the above problem as

$$\begin{aligned} \text{minimize} \quad & \widehat{\mathbf{c}}^T\widehat{\mathbf{x}} \\ \text{subject to} \quad & \widehat{\mathbf{A}}\widehat{\mathbf{x}} - \widehat{\mathbf{b}}\widehat{x}_{14} = \mathbf{0} \\ & \mathbf{e}^T\widehat{\mathbf{x}} + \widehat{x}_{13} - Q\widehat{x}_{14} = 0 \\ & \mathbf{e}^T\widehat{\mathbf{x}} + \widehat{x}_{13} + \widehat{x}_{14} = Q + 1 \\ & \widehat{\mathbf{x}} \geq \mathbf{0}, \widehat{x}_{13} \geq 0, \widehat{x}_{14} \geq 0. \end{aligned}$$

We apply the transformation $\widehat{x}_j = (Q+1)\acute{x}_j$ for $j = 1, 2, \ldots, 14$. In this way, we have an equivalent problem

$$\begin{aligned} \text{minimize} \quad & (Q+1)\widehat{\mathbf{c}}^T\acute{\mathbf{x}} \\ \text{subject to} \quad & \widehat{\mathbf{A}}\acute{\mathbf{x}} - \widehat{\mathbf{b}}\acute{x}_{14} = \mathbf{0} \\ & \mathbf{e}^T\acute{\mathbf{x}} + \acute{x}_{13} - Q\acute{x}_{14} = 0 \\ & \mathbf{e}^T\acute{\mathbf{x}} + \acute{x}_{13} + \acute{x}_{14} = 1 \\ & \acute{\mathbf{x}} \geq \mathbf{0},\ \acute{x}_{13} \geq 0,\ \acute{x}_{14} \geq 0. \end{aligned}$$

Then we convert this problem to satisfy the first assumption of Karmarkar's standard form by introducing an artificial variable $\acute{x}_{15}$ with a large cost coefficient M, we have

$$\begin{aligned} \text{minimize} \quad & (Q+1)\widehat{\mathbf{c}}^T\acute{\mathbf{x}} + M\acute{x}_{15} \\ \text{subject to} \quad & \widehat{\mathbf{A}}\acute{\mathbf{x}} - \widehat{\mathbf{b}}\acute{x}_{14} - [\widehat{\mathbf{A}}\mathbf{e} - \widehat{\mathbf{b}}]\acute{x}_{15} = \mathbf{0} \\ & \mathbf{e}^T\acute{\mathbf{x}} + \acute{x}_{13} - Q\acute{x}_{14} - (12+1-Q)\acute{x}_{15} = 0 \\ & \mathbf{e}^T\acute{\mathbf{x}} + \acute{x}_{13} + \acute{x}_{14} + \acute{x}_{15} = 1 \\ & \acute{\mathbf{x}} \geq \mathbf{0},\ \acute{x}_{13} \geq 0,\ \acute{x}_{14} \geq 0,\ \acute{x}_{15} \geq 0. \end{aligned} \tag{2.16}$$

Now, problem (2.16) is in Karmarkar's standard form, we may rewrite the matrix $\acute{\mathbf{A}}$ and $\acute{\mathbf{c}}$ of problem (2.16) as follows

$$\acute{\mathbf{A}} = \begin{bmatrix} \widehat{\mathbf{A}} & 0 & -\widehat{\mathbf{b}} & -(\widehat{\mathbf{A}}\mathbf{e} - \widehat{\mathbf{b}}) \\ \mathbf{e}^T & 1 & -Q & -(12+1-Q) \end{bmatrix}, \ \acute{\mathbf{c}} = \begin{bmatrix} (Q+1)\widehat{\mathbf{c}} \\ 0 \\ 0 \\ M \end{bmatrix}.$$

It's obvious that $\acute{\mathbf{x}}^{(0)} = \mathbf{e}/15$ is an initial interior feasible solution to the problem (2.16). Hence for Step 1, we can start with $Q = 100$, $M = 1000$, $\alpha = 0.99$, $\epsilon = 10^{-8}$ and

$$\acute{\mathbf{x}}^{(0)} = \left[\frac{1}{15}, \frac{1}{15}, \ldots, \frac{1}{15}\right]^T.$$

For Step 2, we have $\acute{\mathbf{c}}^T\acute{\mathbf{x}}^{(0)} = 65.4667 > 10^{-8}$. Therefore we have to find a better solution. Then in Step 3, we compute

$$\overline{\mathbf{X}} = diag(\acute{\mathbf{x}}^{(0)}),$$

$$\mathbf{B} = \begin{bmatrix} \acute{\mathbf{A}}\overline{\mathbf{X}} \\ \mathbf{e}^T \end{bmatrix}$$

and

$$\begin{aligned} \mathbf{d}^{(0)} &= -\left[\mathbf{I} - \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}\right]\overline{\mathbf{X}}\acute{\mathbf{c}} \\ &= [4.893, 14.557, -2.208, 7.320, -2.805, -2.922, 1.347, 1.465, -1.444, -3.129, \\ &\quad -2.308, -2.07, -0.728, -5.501, -6.461]^T. \end{aligned}$$

For Step 4, we compute a new solution ;

$$\begin{aligned} \acute{\mathbf{y}}^{(1)} &= \frac{\mathbf{e}}{15} + \frac{0.99}{15}\left(\frac{\mathbf{d}^{(0)}}{\|\mathbf{d}^{(0)}\|}\right) \\ &= [0.082, 0.114, 0.059, 0.090, 0.057, 0.057, 0.071, 0.071, \\ &\quad 0.056, 0.059, 0.064, \; 0.062, 0.048, 0.045]^T \\ \acute{\mathbf{x}}^{(1)} &= \frac{\overline{\mathbf{X}}\acute{\mathbf{y}}^{(1)}}{\mathbf{e}^T\overline{\mathbf{X}}\acute{\mathbf{y}}^{(1)}} \\ &= [0.082, 0.114, 0.059, 0.090, 0.057, 0.057, 0.071, 0.071, \\ &\quad 0.062, 0.056, 0.059, 0.064, \; 0.048, 0.045]^T. \end{aligned}$$

Continuing this iterative process, Karmarkar's algorithm will stop at the optimal $\acute{x}_1^* = 0.0990099$, $\acute{x}_2^* = 0.2970297$, $\acute{x}_3^* = 0.53 \times 10^{-9}$, $\acute{x}_4^* = 0.14 \times 10^{-9}$. Remember that $x_j = (1+Q)\acute{x}_j$ for $j = 1, 2, \ldots, 4$. Hence we have the optimal $x_1^* = 10$, $x_2^* = 30$, $x_3^* = 0$, $x_4^* = 0$ to problem (2.14) with objective value $= -380$.

□

2.2 The Primal Affine Scaling Algorithm

Let us consider a linear programming problem in its standard form:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{2.17}$$

where $\mathbf{A}$ is an $m \times n$ matrix of full row rank, $\mathbf{c}$ and $\mathbf{x}$ are n-dimensional column vectors, $\mathbf{b}$ is an m-dimensional vector.

The guiding principles for the primal affine scaling algorithm are two fundamental insights from the Karmarkar's algorithm, we repeat them here:

1. if the current interior solution is near the center of the polytope, then it makes sense to move in the direction of the steepest descent of the objective function to achieve a minimum value;

2. without changing the problem, an appropriate transformation can be applied to the solution space such that the current interior solution is placed near the center in the transformed solution space.

In this algorithm, we directly work on the standard form problem. The simplex structure is no longer available, and the feasible domain could become an unbounded polyhedral set. If the feasible domain is not unbounded, then the feasible domain is the intersection of the affine space $\{\mathbf{x} \in I\!R^n | \mathbf{A}\mathbf{x} = \mathbf{b}\}$ and the positive orthant $\{\mathbf{x} \in I\!R^n | \mathbf{x} \geq \mathbf{0}\}$. It is obvious that the nonnegative orthant does not have a real "center" point. However, if we position ourselves at the point $\mathbf{e} = (1, 1, \ldots, 1)^T$, at least we still keep equal distance from each facet, or "wall" of the nonnegative orthant. As long as the moving distance is less than one unit, any new interior point that moves from $\mathbf{e}$ remains in the interior of the nonnegative orthant. Consequently, if we were able to find an appropriate transformation that maps a current interior point to the point $\mathbf{e}$, then, in parallel with Karmarkar's projective scaling algorithm, we can state a modified strategy as follows:

" Take an interior solution, apply the appropriate transformation to the solution space so as to place the current solution at $\mathbf{e} = (1, 1, \dots, 1)^T$ in the transformed solution space, and then move in the direction of the steepest descent in the null space of the transformed explicit constraints, but not all the way to the nonnegativity walls in order to remain as an interior solution. Then we take the inverse transformation to map the improved solution back to the original solution space as a new interior solution. Repeat this process until the optimality or the stopping conditions are met."

2.2.1 Affine Scaling Transformation on the Nonnegative Orthant

Let $\overline{\mathbf{x}} \in I\!R^n$ be an interior solution of the nonnegative orthant $I\!R^n_+$, ie., $\overline{x}_j > 0$ for $j = 1, 2, \dots, n$ and define $\overline{\mathbf{X}}$ as in 2.1.3, i.e.,

$$\overline{\mathbf{X}} = diag(\overline{\mathbf{x}}) = \begin{bmatrix} \overline{x}_1 & 0 & \dots & 0 \\ 0 & \overline{x}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \overline{x}_n \end{bmatrix}.$$

The affine scaling transformation is defined from the nonnegative orthant $I\!R^n_+$ to $I\!R^n_+$ by

$$\mathbf{y} = \mathrm{T}(\mathbf{x}) = \overline{\mathbf{X}}^{-1}\mathbf{x} \quad \text{for} \quad \mathbf{x} \in I\!R^n_+. \tag{2.18}$$

Note that, we can find $\mathbf{x}$ in terms of its image $\mathbf{y}$ by the formula

$$\mathbf{x} = \mathrm{T}^{-1}(\mathbf{y}) = \overline{\mathbf{X}}\mathbf{y} \quad \text{for } \mathbf{y} \in I\!R^n_+. \tag{2.19}$$

2.2.2 The Primal Affine Scaling Algorithm

Suppose that $\overline{\mathbf{x}}$ is an interior solution of the linear programming problem (2.17), we can apply the affine scaling transformation $\mathbf{T}$ to map $\overline{\mathbf{x}}$ to the center $\mathbf{e} = (1, 1, \ldots, 1)^T$ in the transformed solution space. By the relationship $\mathbf{x} = \overline{\mathbf{X}}\mathbf{y}$, we have a corresponding problem in the transformed solution space as follows:

$$\begin{aligned} \text{minimize} \quad & (\hat{\mathbf{c}})^T\mathbf{y} \\ \text{subject to} \quad & \hat{\mathbf{A}}\mathbf{y} = \mathbf{b} \\ & \mathbf{y} \geq \mathbf{0} \end{aligned} \tag{2.20}$$

where $\hat{\mathbf{c}} = \overline{\mathbf{X}}\mathbf{c}$ and $\hat{\mathbf{A}} = \mathbf{A}\overline{\mathbf{X}}$.

In the problem (2.20), the image of $\overline{\mathbf{x}}$, i.e., $\overline{\mathbf{y}} = \mathbf{T}(\overline{\mathbf{x}})$ becomes $\mathbf{e}$ that keeps unit distance away from the walls of the nonnegative orthant. We want to find a direction $\mathbf{d}_{\overline{\mathbf{y}}}$ which lies in the null space of the matrix $\hat{\mathbf{A}} = \mathbf{A}\overline{\mathbf{X}}$ for an appropriate step-length $\alpha > 0$, then the new point $\mathbf{y}^{new} = \mathbf{e} + \alpha\mathbf{d}_{\overline{\mathbf{y}}}$ remains interior feasible to the problem (2.20) and its inverse image $\mathbf{x}^{new} = \mathbf{T}^{-1}(\mathbf{y}^{new}) = \overline{\mathbf{X}}\mathbf{y}^{new}$ becomes a new interior solution to the problem (2.17).

Since our objective is to minimize the value of the objective function, we can improve the objective value by moving the point $\overline{\mathbf{y}}$ in a direction of the steepest descent. In other words, we want to project the negative gradient $-\hat{\mathbf{c}}$ onto the null space of the matrix $\hat{\mathbf{A}}$ to create a good direction $\mathbf{d}_{\overline{\mathbf{y}}}$. In order to do so, we first define the *null space projective matrix* by

$$\mathbf{P} = \mathbf{I} - \hat{\mathbf{A}}^T(\hat{\mathbf{A}}\hat{\mathbf{A}}^T)^{-1}\hat{\mathbf{A}} = \mathbf{I} - \overline{\mathbf{X}}\mathbf{A}^T(\mathbf{A}\overline{\mathbf{X}}^2\mathbf{A}^T)^{-1}\mathbf{A}\overline{\mathbf{X}}.$$

Then, the moving direction $\mathbf{d}_{\mathbf{y}}$, similar to (2.5), is given by

$$\mathbf{d}_{\overline{\mathbf{y}}} = \mathbf{P}(-\hat{\mathbf{c}}) = -\left[\mathbf{I} - \overline{\mathbf{X}}\mathbf{A}^T(\mathbf{A}\overline{\mathbf{X}}^2\mathbf{A}^T)^{-1}\mathbf{A}\overline{\mathbf{X}}\right]\overline{\mathbf{X}}\mathbf{c}.$$

Now, we are in a position to translate, in the transformed solution space, the current interior solution $\overline{\mathbf{y}} = \mathbf{e}$ along the direction of $\mathbf{d}_{\overline{\mathbf{y}}}$ to a new interior solution

$\mathbf{y}^{new} > \mathbf{0}$ with an improved objective value. In doing so, we have to choose an appropriate step-length $\alpha > 0$ such that $\mathbf{y}^{new} = \overline{\mathbf{y}} + \alpha \mathbf{d}_{\overline{\mathbf{y}}} > \mathbf{0}$.

Notice that if $\mathbf{d}_{\overline{\mathbf{y}}} \geq \mathbf{0}$, then α can be any positive number without leaving the interior region. On the other hand, if $(\mathbf{d}_{\overline{\mathbf{y}}})_i < 0$ for some i, then α has to be smaller than

$$\frac{\overline{\mathbf{y}}_i}{-(\mathbf{d}_{\overline{\mathbf{y}}})_i} = \frac{1}{-(\mathbf{d}_{\overline{\mathbf{y}}})_i}.$$

Therefore we can choose $0 < \delta < 1$ and apply the minimum ratio test

$$\alpha = min \left\{ \frac{\delta}{-(\mathbf{d}_{\overline{\mathbf{y}}})_i} \mid (\mathbf{d}_{\overline{\mathbf{y}}})_i < 0, \quad i = 1, 2, \ldots, n \right\}$$

to determine an appropriate step-length that guarantees the positivity of $\mathbf{y}^{new}$.

Our next task is to take the inverse transformation to map $\mathbf{y}^{new}$ back to the original solution space to obtain a new interior solution $\mathbf{x}^{new}$ by

$$\begin{aligned} \mathbf{x}^{new} &= \mathbf{T}^{-1}(\mathbf{y}^{new}) = \overline{\mathbf{X}}\mathbf{y}^{new} \\ &= \overline{\mathbf{x}} + \alpha \overline{\mathbf{X}} \mathbf{d}_{\overline{\mathbf{y}}} \\ &= \overline{\mathbf{x}} - \alpha \overline{\mathbf{X}} \left[\mathbf{I} - \overline{\mathbf{X}} \mathbf{A}^T (\mathbf{A} \overline{\mathbf{X}}^2 \mathbf{A}^T)^{-1} \mathbf{A} \overline{\mathbf{X}} \right] \overline{\mathbf{X}} \mathbf{c} \\ &= \overline{\mathbf{x}} - \alpha \overline{\mathbf{X}}^2 \left[\mathbf{c} - \mathbf{A}^T (\mathbf{A} \overline{\mathbf{X}}^2 \mathbf{A}^T)^{-1} \mathbf{A} \overline{\mathbf{X}}^2 \mathbf{c} \right] \\ &= \overline{\mathbf{x}} - \alpha \overline{\mathbf{X}}^2 \left[\mathbf{c} - \mathbf{A}^T \mathbf{w} \right] \end{aligned}$$

where

$$\mathbf{w} = (\mathbf{A} \overline{\mathbf{X}}^2 \mathbf{A}^T)^{-1} \mathbf{A} \overline{\mathbf{X}}^2 \mathbf{c}. \tag{2.21}$$

Lemma 2.2.1 [4]. If there exists an $\overline{\mathbf{x}} \in \{\mathbf{x} \in I\!R^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} > \mathbf{0}\}$ with $\mathbf{d}_{\overline{\mathbf{y}}} > 0$, then the linear programming problem (2.17) is unbounded.

Lemma 2.2.2 [4]. If there exists an $\overline{\mathbf{x}} \in \{\mathbf{x} \in I\!R^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} > \mathbf{0}\}$ with $\mathbf{d}_{\overline{\mathbf{y}}} = 0$, then the linear programming problem (2.17) is optimal.

Observation [4]. If $\overline{\mathbf{x}}$ is actually a vertex point, then $\mathbf{w} = (\mathbf{A} \overline{\mathbf{X}}^2 \mathbf{A}^T)^{-1} \mathbf{A} \overline{\mathbf{X}}^2 \mathbf{c}$ is a *dual vector*. Hence we call $\mathbf{w}$ the *dual estimates* in the primal affine scaling algorithm. Moreover, in this case, the quantity

$$\mathbf{r} = \mathbf{c} - \mathbf{A}^T\mathbf{w} \tag{2.22}$$

is a *reduced cost vector.*

Note that when $\mathbf{r} \geq \mathbf{0}$, the dual estimate $\mathbf{w}$ becomes a dual feasible and $(\overline{\mathbf{x}})^T\mathbf{r} = \mathbf{e}^T\overline{\mathbf{X}}\mathbf{r}$ becomes the duality gap. Hence, if $\mathbf{e}^T\overline{\mathbf{X}}\mathbf{r} = 0$ with $\mathbf{r} \geq \mathbf{0}$, then $\overline{\mathbf{x}}$ is primal optimal and $\mathbf{w}$ is dual optimal.

Based on the above discussions, we outline an iterative procedure for the primal affine scaling algorithm.

Step 1. (Initialization) :

Set $k = 0$ and choose $\mathbf{x}^{(0)} > \mathbf{0}$ such that $\mathbf{A}\mathbf{x}^{(0)} = \mathbf{b}$,

choose δ to lie between 0 and 1, and ϵ to be a small positive integer.

Step 2. Compute $\overline{\mathbf{X}} = diag(\mathbf{x}^{(k)})$ and $\mathbf{w}^{(k)} = (\mathbf{A}\overline{\mathbf{X}}^2\mathbf{A}^T)^{-1}\mathbf{A}\overline{\mathbf{X}}^2\mathbf{c}$.

Step 3. Compute $\mathbf{r}^{(k)} = \mathbf{c} - \mathbf{A}^T\mathbf{w}^{(k)}$.

Step 4. (Check for optimality) :

If $\mathbf{r}^{(k)} \geq \mathbf{0}$ and $\mathbf{e}^T\overline{\mathbf{X}}\mathbf{r}^{(k)} \leq \epsilon$

then STOP with $\mathbf{x}^{(k)}$ is *primal optimal* and $\mathbf{w}^{(k)}$ is *dual optimal.*

Otherwise, go to the next step.

Step 5. (Compute the direction): $\mathbf{d}_y^{(k)} = -\overline{\mathbf{X}}\mathbf{r}^{(k)}$.

Step 6. (Check for unboundedness and constant objective value) :

If $\mathbf{d}_y^{(k)} > \mathbf{0}$, then STOP and The problem is *unbounded.*

If $\mathbf{d}_y^{(k)} = \mathbf{0}$, then STOP with $\mathbf{x}^{(k)}$ is *primal optimal solution.*

Otherwise, go to the next step.

Step 7. (Compute the step-length):

$$\alpha = min\left\{ \frac{\delta}{-(\mathbf{d}_y^{(k)})_j} | (\mathbf{d}_y^{(k)})_j < 0, \quad j = 1, 2, \ldots, n \right\}$$

Step 8. (Move to a new solution):

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha\overline{\mathbf{X}}\mathbf{d}_y^{(k)}$$

and $k = k + 1$ go to Step 2.

Lemma 2.2.3 [4]. If the linear programming problem (2.17) is bounded below and its objective function is not constant, then the sequence $\{\mathbf{c}^T\mathbf{x}^{(k)} \mid k = 1, 2, \ldots\}$ is well defined and strictly decreasing.

2.2.3 The Initial Interior Feasible Solution

Consider a linear programming problem (2.17), we want to find an initial interior feasible solution $\mathbf{x}^{(0)}$ for the primal affine scaling algorithm such that $\mathbf{A}\mathbf{x}^{(0)} = \mathbf{0}$ and $\mathbf{x}^{(0)} > \mathbf{0}$. Let us choose an arbitrary $\mathbf{x}^{(0)} > \mathbf{0}$ and calculate $\mathbf{v} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$. If $\mathbf{v} = \mathbf{0}$ then $\mathbf{x}^{(0)}$ is an initial interior feasible solution of the primal affine scaling algorithm, otherwise, solve the following linear programming problem

$$\begin{aligned} &\text{minimize} \quad \mu \\ &\text{Subject to} \quad [\mathbf{A} \;\; \mathbf{v}] \begin{bmatrix} \mathbf{x} \\ \mu \end{bmatrix} = \mathbf{b} \\ &\qquad\qquad \mathbf{x} \geq \mathbf{0}, \; \mu \geq 0 \end{aligned} \tag{2.23}$$

with an initial interior feasible solution $\hat{\mathbf{x}}^{(0)} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mu^{(0)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(0)} \\ 1 \end{bmatrix}$. Hence the primal affine scaling algorithm can be applied to solve the problem (2.23). At the optimal of the problem with an optimal solution $\hat{\mathbf{x}}^* = \begin{bmatrix} \mathbf{x}^* \\ \mu^* \end{bmatrix}$, if $\mu^* > 0$, the problem (2.17) is infeasible. If $\mu^* = 0$ then the primal affine scaling algorithm can be apply to solve the problem (2.17) and $\mathbf{x}^*$ will become an initial interior feasible solution which satisfies $\mathbf{A}\mathbf{x}^* = \mathbf{0}$.

Example 1.2

Consider the same problem as in Example 1.1. In this case, we have

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 50 \\ 70 \end{bmatrix}, \ \mathbf{c} = \begin{bmatrix} -8 \\ -10 \\ 0 \\ 0 \end{bmatrix}$$

For Step 1, let us start with $\mathbf{x}^{(0)} = [5 \ \ 5 \ \ 5 \ \ 5]^T$. We see that $\mathbf{v} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)} = [30 \ \ 50]^T \neq \mathbf{0}$. Hence, we want to find an initial interior feasible solution by solving the following problem

$$\begin{aligned} &\text{minimize} \quad && \hat{\mathbf{c}}^T\hat{\mathbf{x}} \\ &\text{subject to} \quad && \hat{\mathbf{A}}\hat{\mathbf{x}} = \mathbf{b} \\ & && \hat{\mathbf{x}} \geq \mathbf{0} \end{aligned} \tag{2.24}$$

where $\hat{\mathbf{A}} = [\mathbf{A} \ \ \mathbf{v}]$, $\hat{\mathbf{c}} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$ and $\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mu \end{bmatrix}$.

Now, the primal affine scaling algorithm can be apply to solve the problem (2.24) with an initial feasible solution $\hat{\mathbf{x}}^{(0)} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mu^{(0)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(0)} \\ 1 \end{bmatrix}$ such that $\hat{\mathbf{A}}\hat{\mathbf{x}}^{(0)} = \mathbf{b}$ and we can choose $\delta = 0.99$. For Step 2, we compute

$$\overline{\mathbf{X}} = diag(\hat{\mathbf{x}}^{(0)}) = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{w}^{(0)} = (\hat{\mathbf{A}}\overline{\mathbf{X}}^2\hat{\mathbf{A}}^T)^{-1}\hat{\mathbf{A}}\overline{\mathbf{X}}^2\hat{\mathbf{c}} = [-0.0022 \ \ \ 0.0202]^T.$$

For Step 3, we compute

$$\begin{aligned}\mathbf{r}^{(0)} &= \hat{\mathbf{c}} - \hat{\mathbf{A}}^T\mathbf{w}^{(0)} \\ &= [-0.0157 \ \ 0.0382 \ \ 0.0022 \ \ -0.0202 \ \ 0.0561]^T.\end{aligned}$$

In Step 4, since some components of $\mathbf{r}^{(0)}$ are negative, go to Step 5. In Step 5, we compute the direction with

$$\mathbf{d}_\mathbf{y}^{(0)} = -\overline{\mathbf{X}}\mathbf{r}^{(0)} = [0.0787 \ \ 0.191 \ \ -0.0112 \ \ 0.1011 \ \ -0.0562]^T.$$

Since $\mathbf{d}_y^{(0)} \not> \mathbf{0}$ and $\mathbf{d}_y^{(0)} \neq \mathbf{0}$, go to Step 7. then the step-length

$$\alpha = min\left\{ \frac{0.99}{-(\mathbf{d}_\mathbf{y}^{(0)})_j} | (\mathbf{d}_\mathbf{y}^{(0)})_j < 0, \quad j = 1, 2, \ldots, 5 \right\} = 17.622$$

Therefore, the new solution is

$$\hat{\mathbf{x}}^{(1)} = \hat{\mathbf{x}}^{(0)} + \alpha\overline{\mathbf{X}}\mathbf{d}_\mathbf{y}^{(0)} = [11.93 \ \ 21.83 \ \ 4.01 \ \ 13.91 \ \ 0.01]^T.$$

Continuing the iterative process, the primal affine scaling algorithm will stop at the optimal $\hat{\mathbf{x}}^* = [11.9747 \ \ 22.0394 \ \ 4.0110 \ \ 13.9463 \ \ 10^{-9}]^T$(assume that we accepted $10^{-9} \approx 0$) to problem (2.24). Therefore, the primal affine scaling algorithm can be apply to solve the problem (2.14) with an initial feasible solution $\mathbf{x}^{(0)} = [11.9747 \ \ 22.0394 \ \ 4.0110 \ \ 13.9463]^T$. Hence, we have

$$\overline{\mathbf{X}} = diag(\mathbf{x}^{(0)}) = \begin{bmatrix} 11.9747 & 0 & 0 & 0 \\ 0 & 22.0394 & 0 & 0 \\ 0 & 0 & 4.0110 & 0 \\ 0 & 0 & 0 & 13.9463 \end{bmatrix},$$

$$\mathbf{w}^{(0)} = (\mathbf{A}\overline{\mathbf{X}}^2\mathbf{A}^T)^{-1}\mathbf{A}\overline{\mathbf{X}}^2\mathbf{c} = [-3.0413 \ \ -3.0844]^T$$

and

$$\begin{aligned}\mathbf{r}^{(0)} &= \mathbf{c} - \mathbf{A}^T\mathbf{w}^{(0)} \\ &= [1.1670 \ \ -0.7898 \ \ 3.0413 \ \ 3.0844]^T\end{aligned}$$

since some components of $\mathbf{r}^{(0)}$ are negative. Hence, we compute the direction

$$\mathbf{d}_{\mathrm{y}}^{(0)} = -\overline{\mathbf{X}}\mathbf{r}^{(0)} = [-13.9756 \;\; 17.4070 \;\; -12.1988 \;\; -43.0166]^T$$

then the step-length

$$\alpha = min\left\{ \frac{0.99}{-(\mathbf{d}_{\mathrm{y}}^{(0)})_j} | (\mathbf{d}_{\mathrm{y}}^{(0)})_j < 0, \quad j = 1, 2, \ldots, 5 \right\} = 0.02301$$

Therefore, we have the new solution

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha \overline{\mathbf{X}} \mathbf{d}_{\mathrm{y}}^{(0)} = [8.1232 \;\; 30.8686 \;\; 2.8849 \;\; 0.1394]^T$$

Repeating the iterative process, the algorithm will achieve the optimal $\mathbf{x}^* = [10 \;\; 30 \;\; 0 \;\; 0]^T$ with objective value $= -380$.

□

2.3 The Primal-Dual Algorithm

Consider a linear programming problem in its standard form:

$$\begin{aligned} &\text{minimize} && \mathbf{c}^T\mathbf{x} \\ &\text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \end{aligned} \qquad (P)$$

and its dual:

$$\begin{aligned} &\text{maximize} && \mathbf{b}^T\mathbf{w} \\ &\text{subject to} && \mathbf{A}^T\mathbf{w} + \mathbf{s} = \mathbf{c} \\ & && \mathbf{w} \text{ unrestricted}, \quad \mathbf{s} \geq \mathbf{0}. \end{aligned} \qquad (D)$$

We impose the following assumption for the primal-dual algorithm:

(A1) The set $\mathbb{F}_P \equiv \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \; \mathbf{x} > \mathbf{0}\}$ is nonempty.

(A2) The set $\mathbb{F}_D \equiv \{(\mathbf{w}, \mathbf{s}) \in \mathbb{R}^m \times \mathbb{R}^n | \mathbf{A}^T\mathbf{w} + \mathbf{s} = \mathbf{c},\ \mathbf{s} > \mathbf{0}\}$ is nonempty.

(A3) The constraint matrix $\mathbf{A}$ has full row rank.

Under these assumptions, it is clearly seen from the duality theorem that problems (P) and (D) have optimal solutions. Moreover, the sets of the optimal solutions of (P) and (D) are bounded. Note that, for $\mathbf{x} > \mathbf{0}$ in (P), we may apply the logarithmic barrier function technique, and consider the following of nonlinear programming problem (P_μ):

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} - \mu \textstyle\sum_{j=1}^{n} log_e x_j \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} > \mathbf{0} \end{aligned} \qquad (\mathrm{P}_\mu)$$

where $\mu > 0$ is a *barrier* or *penalty* parameter.

As $\mu \to 0$, we would expect the optimal solution of problem (P_μ) to converge to an optimal solution of the original problem (P). Observe that the objective function of the problem (P_μ) is a strictly convex function, hence problem (P_μ) has at most one global minimum. The convex programming theory further implies that the global minimum [7,10], if it exists, is completely characterized by the Kuhn-Tucker conditions:

$$\mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} > \mathbf{0} \quad \text{(primal feasibility)} \qquad (2.21a)$$

$$\mathbf{A}^T\mathbf{w} + \mathbf{s} = \mathbf{c},\ \mathbf{s} > \mathbf{0} \quad \text{(dual feasibility)} \qquad (2.21b)$$

$$\mathbf{X}\mathbf{S}\mathbf{e} - \mu\mathbf{e} = 0 \quad \text{(complementary slackness)} \qquad (2.21c)$$

where $\mathbf{X}$ and $\mathbf{S}$ are diagonal matrices using the components of vectors $\mathbf{x}$ and $\mathbf{s}$ as diagonal elements, respectively.

Lemma 2.3.1[4]. Under the assumptions (A1) and (A2), both problem (P_μ) and system (2.21) have a unique solution.

Observe that system (2.21) also provides the necessary and sufficient conditions (the K-K-T conditions) for $(\mathbf{w}(\mu); \mathbf{s}(\mu))$ being a maximum solution of the following problem (D_μ);

$$\begin{aligned} \text{maximize} \quad & \mathbf{b}^T\mathbf{w} + \mu \textstyle\sum_{j=1}^{n} log_e s_j \\ \text{subject to} \quad & \mathbf{A}^T\mathbf{w} + \mathbf{s} = \mathbf{c} \\ & \mathbf{s} > \mathbf{0}, \mathbf{w} \text{ unrestricted.} \end{aligned} \qquad (D_\mu)$$

Note that Equation (2.21c) can be written componentwise as

$$x_j s_j = \mu, \quad \text{for } j = 1, 2, \ldots, n. \qquad (2.21c')$$

Therefore, when the assumption (A3) is imposed, $\mathbf{x}$ uniquely determines $\mathbf{w}$ from the Equations (2.21c') and (2.21b). We let $(\mathbf{x}(\mu); \mathbf{w}(\mu); \mathbf{s}(\mu))$ denote the unique solution to system (2.21) for each $\mu > 0$. Obviously, we see $\mathbf{x}(\mu) \in I\!F_P$ and $(\mathbf{w}(\mu); \mathbf{s}(\mu)) \in I\!F_D$. Moreover, the duality gap becomes

$$\begin{aligned} g(\mu) &= \mathbf{c}^T\mathbf{x}(\mu) - \mathbf{b}^T\mathbf{w}(\mu) \\ &= (\mathbf{c}^T - \mathbf{w}(\mu)^T\mathbf{A})\mathbf{x}(\mu) \\ &= \mathbf{s}(\mu)^T\mathbf{x}(\mu) \\ &= n\mu. \end{aligned}$$

Therefore, as $\mu \to 0$, the duality gap $g(\mu)$ converges to zero. This implies that $\mathbf{x}(\mu)$ and $(\mathbf{w}(\mu); \mathbf{s}(\mu))$ indeed converge to the optimal solutions of problem (P) and (D), respectively.

Lemma 2.3.2[4]. Under the assumptions (A1)-(A3), as $\mu \to 0$, $\mathbf{x}(\mu)$ converges to the optimal solution of (P) and $(\mathbf{w}(\mu); \mathbf{s}(\mu))$ converges to the optimal solution of problem (D).

For $\mu > 0$, we let Γ denote the curve, or path, consisting of the solution of system (2.21), i.e.,

$$\Gamma = \{(\mathbf{x}(\mu); \mathbf{w}(\mu); \mathbf{s}(\mu)) | (\mathbf{x}(\mu); \mathbf{w}(\mu); \mathbf{s}(\mu)) \text{ solves (2.21) for some } \mu > 0\} \quad (2.22)$$

As $\mu \rightarrow 0$, the path Γ leads to a pair of primal optimal solution $\mathbf{x}^*$ and dual optimal solution $(\mathbf{w}^*; \mathbf{s}^*)$. Thus following the path Γ serves as a theoretical model for a class of primal-dual interior-point methods for linear programming. For this reason, people may classify the primal-dual approach as a *path-following* approach.

2.3.1 Direction and Step-Length of Movement

Let us begin by synthesizing a direction of translation (moving direction) $(\mathbf{d_x}; \mathbf{d_w}; \mathbf{d_s})$ at a current point $(\overline{\mathbf{x}}; \overline{\mathbf{w}}; \overline{\mathbf{s}})$ such that the translation is made along the the curve Γ to a new point $(\mathbf{x}^{new}; \mathbf{w}^{new}; \mathbf{s}^{new})$. This task is taken care of by applying the Newton's method to the system of equations (2.21a)-(2.21c).

Newton's method is one of the most commonly used techniques for finding a root of a system of nonlinear equations via successively approximating the system by linear equations. To be more specific, suppose that $F(\mathbf{z})$ is a nonlinear mapping from $I\!R^n$ to $I\!R^n$ and we need to find a $\mathbf{z}^* \in I\!R^n$ such that $F(\mathbf{z}^*) = \mathbf{0}$. By using the multivariable Taylor series expansion (say at $\mathbf{z} = \overline{\mathbf{z}}$), we obtain a linear approximation:

$$F(\overline{\mathbf{z}} + \Delta\mathbf{z}) \approx F(\overline{\mathbf{z}}) + \mathbf{J}(\overline{\mathbf{z}})\Delta\mathbf{z} \tag{2.23}$$

where $\mathbf{J}(\overline{\mathbf{z}})$ is the Jacobian matrix whose $(i, j)^{th}$ element is given by

$$\left[\frac{\partial F_i(\mathbf{z})}{\partial z_j}\right]_{\mathbf{z}=\overline{\mathbf{z}}}$$

and $\Delta\mathbf{z}$ is a translation vector. As the left-hand side of (2.23) evaluates at a root of $F(\mathbf{z}) = \mathbf{0}$, we have a linear system

$$\mathbf{J}(\overline{\mathbf{z}})\Delta\mathbf{z} = -F(\overline{\mathbf{z}}) \tag{2.24}$$

A solution vector of equation (2.24) provides one Newton iterate from $\overline{\mathbf{z}}$ to $\mathbf{z}^{new} = \overline{\mathbf{z}} + \mathbf{d_z}$ with a *Newton direction* $\mathbf{d_z}$ and a unit step-length.

Let us focus on the nonlinear system (2.21a-c). Assume that we are at a point $(\overline{\mathbf{x}};\overline{\mathbf{w}};\overline{\mathbf{s}})$ for some $\overline{\mu} > 0$, such that $\overline{\mathbf{x}},\ \overline{\mathbf{s}} > \mathbf{0}$. The Newton direction $(\mathbf{d_x};\mathbf{d_w};\mathbf{d_s})$ is determined by the following system of linear equations:

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \mathbf{d_x} \\ \mathbf{d_w} \\ \mathbf{d_s} \end{bmatrix} = - \begin{bmatrix} \mathbf{A}\overline{\mathbf{x}} - \mathbf{b} \\ \mathbf{A}^T\overline{\mathbf{w}} + \overline{\mathbf{s}} - \mathbf{c} \\ \mathbf{XSe} - \overline{\mu}\mathbf{e} \end{bmatrix} \tag{2.25}$$

where $\mathbf{X}$ and $\mathbf{S}$ are the diagonal matrices formed by $\overline{\mathbf{x}}$ and $\overline{\mathbf{s}}$, respectively. Multiplying it out, we have

$$\mathbf{A}\mathbf{d_x} = \mathbf{t} \tag{2.26}$$

$$\mathbf{A}^T\mathbf{d_w} + \mathbf{d_s} = \mathbf{u} \tag{2.27}$$

$$\mathbf{S}\mathbf{d_x} + \mathbf{X}\mathbf{d_s} = \mathbf{v} \tag{2.28}$$

where $\mathbf{t} = \mathbf{b} - \mathbf{A}\overline{\mathbf{x}}$, $\mathbf{u} = \mathbf{c} - \mathbf{A}^T\overline{\mathbf{w}} - \overline{\mathbf{s}}$, and $\mathbf{v} = \overline{\mu}\mathbf{e} - \mathbf{XSe}$. Notice that if $\overline{\mathbf{x}} \in I\!\!F_P$ and $(\overline{\mathbf{w}};\overline{\mathbf{s}}) \in I\!\!F_D$, then $\mathbf{t} = \mathbf{0}$ and $\mathbf{u} = \mathbf{0}$ correspondingly. To solve (2.25), we multiply both sides of Equation (2.27) by $\mathbf{AXS}^{-1}$. Then we have

$$\mathbf{AXS}^{-1}\mathbf{A}^T\mathbf{d_w} = \mathbf{AXS}^{-1}\mathbf{u} - \mathbf{AXS}^{-1}\mathbf{d_s}. \tag{2.29}$$

Now from Equation (2.28), we have

$$\mathbf{d_s} = \mathbf{X}^{-1}\mathbf{v} - \mathbf{X}^{-1}\mathbf{S}\mathbf{d_x}. \tag{2.30}$$

We can denote $\mathbf{p} = \mathbf{X}^{-1}\mathbf{v}$. Using Equation (2.30) and Equation (2.26) in the last term of (2.29) would produce

$$\begin{aligned} \mathbf{AXS}^{-1}\mathbf{d_s} &= \mathbf{AXS}^{-1}(\mathbf{X}^{-1}\mathbf{v} - \mathbf{X}^{-1}\mathbf{S}\mathbf{d_x}) \\ &= \mathbf{AXS}^{-1}\mathbf{p} - \mathbf{t}. \end{aligned} \tag{2.31}$$

Substituting Equation (2.31) back into Equation (2.29) yields

$$\mathbf{d_w} = \left[\mathbf{AXS}^{-1}\mathbf{A}^T\right]^{-1}(\mathbf{AXS}^{-1}(\mathbf{u}^k - \mathbf{p}) + \mathbf{t}) \tag{2.32}$$

where $\mathbf{XS}^{-1}$ is a positive definite diagonal matrix.

Once $\mathbf{d}_w$ is obtained, $\mathbf{d}_s$ and $\mathbf{d}_x$ can be readily computed by

$$\mathbf{d}_s = \mathbf{u} - \mathbf{A}^T\mathbf{d}_w \tag{2.33}$$

and

$$\mathbf{d}_x = \mathbf{XS}^{-1}(\mathbf{p} - \mathbf{d}_s). \tag{2.34}$$

Hence, for $(\overline{\mathbf{x}}; \overline{\mathbf{w}}; \overline{\mathbf{s}}) \in I\!F_P \times I\!F_D$, Equations (2.32-2.34) are simplified as

$$\begin{aligned} \mathbf{d}_w &= -\left[\mathbf{AD}^2\mathbf{A}^T\right]^{-1}\mathbf{AS}^{-1}\mathbf{v} \\ \mathbf{d}_s &= -\mathbf{A}^T\mathbf{d}_w \\ \mathbf{d}_x &= \mathbf{S}^{-1}(\mathbf{v} - \mathbf{X}\mathbf{d}_s) \end{aligned} \tag{2.35}$$

where $\mathbf{D}^2 = \mathbf{XS}^{-1}$.

After obtaining a Newton direction, the primal-dual algorithm iterates to a new point according to the following translation:

$$\mathbf{x}^{new} = \overline{\mathbf{x}} + \beta\mathbf{d}_x$$

$$\mathbf{w}^{new} = \overline{\mathbf{w}} + \beta\mathbf{d}_w$$

$$\mathbf{s}^{new} = \overline{\mathbf{s}} + \beta\mathbf{d}_s$$

where β is a step-length with $\beta \in (0, 1]$. Unfortunately, we can often take only a small step-length along the direction before violating the condition $\mathbf{x}^{new} > \mathbf{0}$ and $\mathbf{s}^{new} > \mathbf{0}$, hence, the Newton direction often does not allow us to make much process toward a solution. Once the moving direction is obtained, we are ready to move to a new point $(\mathbf{x}^{new}; \mathbf{w}^{new}; \mathbf{s}^{new})$ with $\mathbf{x}^{new} > \mathbf{0}$ and $\mathbf{s}^{new} > \mathbf{0}$. Let

$$\begin{aligned} \mathbf{x}^{new} &= \overline{\mathbf{x}} + \beta_P\mathbf{d}_x \\ \mathbf{w}^{new} &= \overline{\mathbf{w}} + \beta_D\mathbf{d}_w \\ \mathbf{s}^{new} &= \overline{\mathbf{s}} + \beta_D\mathbf{d}_s \end{aligned} \tag{2.36}$$

where β_P and β_D are the step-lengths in the primal and dual spaces, respectively. The nonnegative requirements of $\mathbf{x}^{new}$ and $\mathbf{s}^{new}$ dictate the choice of the step-lengths β_P and β_D. We can define the step-lengths by

$$\beta_P = \frac{1}{max\{1, -(\mathbf{d_x})_j/(\alpha \overline{x}_j)\}} \tag{2.37}$$

and

$$\beta_D = \frac{1}{max\{1, -(\mathbf{d_s})_j/(\alpha \overline{s}_j)\}} \tag{2.38}$$

where $\alpha < 1$, $(\mathbf{d_x})_j$ is the j^{th} component of $\mathbf{d_x}$, $\overline{x}_j$ is the j^{th} component of $\overline{\mathbf{x}}$, $(\mathbf{d_s})_j$ is the j^{th} component of $\mathbf{d_s}$, $\overline{s}_j$ is the j^{th} component of $\overline{\mathbf{s}}$.

For the penalty parameter $\overline{\mu}$, remembering the notations defined in (2.21), since we want to reduce the duality gap, we may choose the penalty parameter to be a smaller number by setting

$$\overline{\mu} = \sigma(\overline{\mathbf{s}})^T \overline{\mathbf{x}}/n$$

where $0 < \sigma < 1$.

Based on the above discussions, here we outline an iterative procedure for the primal-dual algorithm.

Step 1. (initialization):

Set $k = 0$, choose α and σ lies between 0 and 1.

Set ϵ to be a small positive number.

Find a starting solution $(\mathbf{x}^{(0)}; \mathbf{w}^{(0)}; \mathbf{s}^{(0)}) \in \mathbb{F}_P \times \mathbb{F}_D$.

Step 2. (checking for optimality):

If $\mathbf{c}^T\mathbf{x}^{(k)} - \mathbf{b}^T\mathbf{w}^{(k)} < \epsilon$

then STOP and The solution is *optimal.*

Otherwise, go to Step 3.

Step 3. Compute $\mu^{(k)} = \sigma(\mathbf{s}^{(k)})^T\mathbf{x}^{(k)}/n$, $\mathbf{v}^{(k)} = \mu^{(k)}\mathbf{e} - \mathbf{XSe}$ and $\mathbf{D}^2 = \mathbf{XS}^{-1}$

where $\mathbf{X}$ and $\mathbf{S}$ are diagonal matrices using the components of vectors

$\mathbf{x}^{(k)}$ and $\mathbf{s}^{(k)}$ as diagonal elements, respectively.

Step 4. (finding the direction of translation):

$$\begin{aligned}
\mathbf{d}_{\mathbf{w}}^{(k)} &= -\left[\mathbf{A}\mathbf{D}^2\mathbf{A}^T\right]^{-1}\mathbf{A}\mathbf{S}^{-1}\mathbf{v}^{(k)} \\
\mathbf{d}_{\mathbf{s}}^{(k)} &= -\mathbf{A}^T\mathbf{d}_{\mathbf{w}}^{(k)} \\
\mathbf{d}_{\mathbf{x}}^{(k)} &= \mathbf{S}^{-1}(\mathbf{v}^{(k)} - \mathbf{X}\mathbf{d}_{\mathbf{s}}^{(k)})
\end{aligned}$$

Step 5. (checking for unboundedness):

If $\mathbf{A}\mathbf{x}^{(k)} = \mathbf{b}$, $\mathbf{d}_{\mathbf{x}}^{(k)} > \mathbf{0}$ and $\mathbf{c}^T\mathbf{d}_{\mathbf{x}}^{(k)} < 0$ then STOP and the primal problem is *unbounded*.

If $\mathbf{A}^T\mathbf{w}^{(k)} + \mathbf{s}^{(k)} = \mathbf{c}$, $\mathbf{d}_{\mathbf{s}}^{(k)} > \mathbf{0}$ and $\mathbf{c}^T\mathbf{d}_{\mathbf{w}}^{(k)} > 0$ then STOP and the dual problem is *unbounded*.

Otherwise, go to the next step.

Step 6. (calculating step-length):

$$\beta_P^{(k)} = \frac{1}{max\{1, -(\mathbf{d}_{\mathbf{x}}^{(k)})_j/(\alpha x_j^{(k)})\}}$$

$$\beta_D^{(k)} = \frac{1}{max\{1, -(\mathbf{d}_{\mathbf{s}}^{(k)})_j/(\alpha s_j^{(k)})\}}$$

Step 7. (moving to a new solution):

$$\begin{aligned}
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \beta_P^{(k)}\mathbf{d}_{\mathbf{x}}^{(k)} \\
\mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} + \beta_D^{(k)}\mathbf{d}_{\mathbf{w}}^{(k)} \\
\mathbf{s}^{(k+1)} &= \mathbf{s}^{(k)} + \beta_D^{(k)}\mathbf{d}_{\mathbf{s}}^{(k)}
\end{aligned}$$

Set $k = k + 1$ and go to Step 2.

2.3.2 Starting the Primal-Dual Algorithm

In order to apply the primal-dual algorithm, we start with an arbitrary point $(\mathbf{x}^{(0)}; \mathbf{w}^{(0)}; \mathbf{s}^{(0)}) \in I\!R^{n+m+n}$ such that $\mathbf{x}^{(0)} > \mathbf{0}$ and $\mathbf{s}^{(0)} > \mathbf{0}$. If $\mathbf{A}\mathbf{x}^{(0)} = \mathbf{b}$ and $\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} = \mathbf{c}$ then we have an initial feasible solution for the primal-dual algorithm else we consider the following pair of artificial primal and dual problems:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} + \delta x_{n+1} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + (\mathbf{b} - \mathbf{A}\mathbf{x}^{(0)})x_{n+1} = \mathbf{b} \\ & (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})^T\mathbf{x} + x_{n+2} = \lambda \\ & (\mathbf{x}; x_{n+1}; x_{n+2}) \geq \mathbf{0} \end{aligned} \tag{AP}$$

where x_{n+1} and x_{n+2} are two artificial variables and δ and λ are sufficiently large positive numbers to be specified later;

$$\begin{aligned} \text{maximize} \quad & \mathbf{b}^T\mathbf{w} + \lambda w_{m+1} \\ \text{subject to} \quad & \mathbf{A}^T\mathbf{w} + \mathbf{s} + (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})w_{m+1} = \mathbf{c} \\ & (\mathbf{b} - \mathbf{A}\mathbf{x}^{(0)})^T\mathbf{w} + s_{n+1} = \delta \\ & w_{m+1} + s_{n+2} = 0 \\ & (\mathbf{s}; s_{n+1}; s_{n+2}) \geq \mathbf{0} \end{aligned} \tag{AD}$$

where w_{m+1}, s_{n+1} and s_{n+2} are artificial variables.

Notice that if we choose δ and λ such that

$$\delta > (\mathbf{b} - \mathbf{A}\mathbf{x}^{(0)})^T\mathbf{w}^{(0)} \tag{2.39}$$

$$\lambda > (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})^T\mathbf{x}^{(0)} \tag{2.40}$$

then $(\mathbf{x}^{(0)}, x_{n+1}^{(0)}, x_{n+2}^{(0)})$ and $(\mathbf{w}^{(0)}, w_{m+1}^{(0)}; \mathbf{s}^{(0)}, s_{n+1}^{(0)}, s_{n+2}^{(0)})$ are feasible solutions to the artificial problems (AP) and (AD), respectively, where

$$\begin{aligned} x_{n+1}^{(0)} &= 1 \\ x_{n+2}^{(0)} &= \lambda - (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})^T\mathbf{x}^{(0)} \\ w_{m+1}^{(0)} &= -1 \\ s_{n+1}^{(0)} &= \delta - (\mathbf{b} - \mathbf{A}\mathbf{x}^{(0)})^T\mathbf{w}^{(0)} \\ s_{n+2}^{(0)} &= 1. \end{aligned}$$

In this case, the primal-dual algorithm can be apply to the artificial problems (AP) and (AD) with a known starting solution. Actually, the optimal solutions of (AP) and (AD) are closely related to those of the original problems (P) and (D).

Thorem 2.3 [4]. Let $\mathbf{x}^*$ and $(\mathbf{w}^*; \mathbf{s}^*)$ be optimal solutions of the original problems (P) and (D). In addition to (2.3.2) and (2.40), suppose that

$$\delta > (\mathbf{b} - \mathbf{A}\mathbf{x}^{(0)})^T\mathbf{w}^*$$

and

$$\lambda > (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})^T\mathbf{x}^*.$$

Then the following statements are true:

(i) A feasible solution $(\overline{\mathbf{x}}, \overline{x}_{n+1}, \overline{x}_{n+2})$ of (AP) is a minimizer if and only if $\overline{\mathbf{x}}$ solves (P) and $\overline{x}_{n+1} = 0$.

(ii) A feasible solution $(\overline{\mathbf{w}}, \overline{w}_{m+1}; \overline{\mathbf{s}}, \overline{s}_{n+1}, \overline{s}_{n+2})$ of (AD) is a maximizer if and only if $(\overline{\mathbf{w}}; \overline{\mathbf{s}})$ solves (D) and $\overline{w}_{m+1} = 0$.

Example 1.3

Consider the same problem as in Example 1.1. In this case, we have

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 50 \\ 70 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} -8 \\ -10 \\ 0 \\ 0 \end{bmatrix}$$

We begin with an arbitrary assignment of $\mathbf{x}^{(0)} = [10 \;\; 10 \;\; 10 \;\; 10]^T$, $\mathbf{w}^{(0)} = [10 \;\; 10]^T$ and $\mathbf{s}^{(0)} = [10 \;\; 10 \;\; 10 \;\; 10]^T$, we see that $\mathbf{Ax}^{(0)} \neq \mathbf{b}$ and $\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} \neq \mathbf{c}$. Hence, we start by considering the following pair of the artificial primal and dual problems:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} + \delta x_5 \\ \text{subject to} \quad & \mathbf{Ax} + (\mathbf{b} - \mathbf{Ax}^{(0)})x_5 = \mathbf{b} \\ & (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})^T\mathbf{x} + x_6 = \lambda \\ & (\mathbf{x}; x_5; x_6) \geq \mathbf{0} \end{aligned} \tag{2.41}$$

and

$$\begin{aligned} \text{maximize} \quad & \mathbf{b}^T\mathbf{w} + \lambda w_3 \\ \text{subject to} \quad & \mathbf{A}^T\mathbf{w} + \mathbf{s} + (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})w_5 = \mathbf{c} \\ & (\mathbf{b} - \mathbf{Ax}^{(0)})^T\mathbf{w} + s_5 = \delta \\ & w_3 + s_6 = 0 \\ & (\mathbf{s}; s_5; s_6) \geq \mathbf{0} \end{aligned} \tag{2.42}$$

where we can choose $\delta = (\mathbf{b} - \mathbf{Ax}^{(0)})^T\mathbf{w}^{(0)} + 10 = 410$ and $\lambda = (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})^T\mathbf{x}^{(0)} + 10 = 1390$.

Now, we denote that

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & (\mathbf{b} - \mathbf{Ax}^{(0)}) & 0 \\ (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})^T & 0 & 1 \end{bmatrix}, \tilde{\mathbf{b}} = \begin{bmatrix} \mathbf{b} \\ \lambda \end{bmatrix}, \tilde{\mathbf{c}} = \begin{bmatrix} \mathbf{c} \\ \delta \\ 0 \end{bmatrix},$$

$$\tilde{\mathbf{x}} = [\mathbf{x} \;\; x_5 \;\; x_6]^T, \tilde{\mathbf{w}} = [\mathbf{w} \;\; w_3]^T \text{ and } \tilde{\mathbf{s}} = [\mathbf{s} \;\; s_5 \;\; s_6]^T.$$

It is obvious that $\tilde{\mathbf{x}}^{(0)} = [\mathbf{x}^{(0)}, x_5^{(0)}, x_6^{(0)}]^T$ and $(\tilde{\mathbf{w}}^{(0)}; \tilde{\mathbf{s}}^{(0)}) = ([\mathbf{w}^{(0)}, w_3^{(0)}]^T; [\mathbf{s}^{(0)}, s_5^{(0)}, s_6^{(0)}]^T)$ are feasible solutions to problems (2.41) and (2.42), respectively, where

$$\begin{aligned}
x_5^{(0)} &= 1 \\
x_6^{(0)} &= \lambda - (\mathbf{A}^T\mathbf{w}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c})^T\mathbf{x}^{(0)} = 10 \\
w_3^{(0)} &= -1 \\
s_5^{(0)} &= \delta - (\mathbf{b} - \mathbf{A}\mathbf{x}^{(0)})^T\mathbf{w}^{(0)} = 10 \\
s_6^{(0)} &= 1.
\end{aligned}$$

Therefore, the primal-dual algorithm can be apply to solve problems (2.41) and (2.42) with a feasible solution $(\tilde{\mathbf{x}}^{(0)}; \tilde{\mathbf{w}}^{(0)}; \tilde{\mathbf{s}}^{(0)})$ such that $\tilde{\mathbf{x}}^{(0)} > \mathbf{0}$ and $\tilde{\mathbf{s}}^{(0)} > \mathbf{0}$.

Step 1, we choose $\alpha = 0.99$, $\epsilon = 10^{-8}$ and $\sigma = 0.5$.

In Step 2, we compute $\tilde{\mathbf{c}}^T\tilde{\mathbf{x}}^{(0)} - \tilde{\mathbf{b}}^T\tilde{\mathbf{w}}^{(0)} = 420 > \epsilon = 10^{-8}$.

Hence, go to Step 3, we have

$$\begin{aligned}
\mu^{(0)} &= (0.5)\Sigma_{j=1}^{6}(\tilde{x}_j^{(0)} \times \tilde{s}_j^{(0)})/6 = 35 \\
\tilde{\mathbf{X}} &= diag(\tilde{\mathbf{x}}^{(0)}), \;\; \tilde{\mathbf{S}} = diag(\tilde{\mathbf{s}}^{(0)}) \\
\mathbf{v}^{(0)} &= \mu^{(0)}\mathbf{e} - \tilde{\mathbf{X}}\tilde{\mathbf{S}}\mathbf{e} = [-60.8001 \;\; -60.8001 \;\; -60.8001 \;\; -60.8001 \;\; 29.1999 \;\; 29.1999]^T \\
\mathbf{D}^2 &= \tilde{\mathbf{X}}\tilde{\mathbf{S}}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}
\end{aligned}$$

For Step 4, we find the directions of translation:

$$\begin{aligned}
\mathbf{d}_{\tilde{\mathbf{w}}}^{(0)} &= -\left[\tilde{\mathbf{A}}\mathbf{D}^2\tilde{\mathbf{A}}^T\right]^{-1}\tilde{\mathbf{A}}\tilde{\mathbf{S}}^{-1}\mathbf{v}^{(0)} \\
&= [-0.4912 \;\; -0.8052 \;\; 0.1828]^T \\
\mathbf{d}_{\tilde{\mathbf{s}}}^{(0)} &= -\tilde{\mathbf{A}}^T\mathbf{d}_{\tilde{\mathbf{w}}}^{(0)} \\
&= [-6.9903 \;\; -7.0421 \;\; -3.1662 \;\; -2.8523 \;\; 29.0688 \;\; -0.1828]^T \\
\mathbf{d}_{\tilde{\mathbf{x}}}^{(0)} &= \tilde{\mathbf{S}}^{-1}(\mathbf{v}^{(0)} - \tilde{\mathbf{X}}\mathbf{d}_{\tilde{\mathbf{s}}}^{(0)}) \\
&= [0.9103 \;\; 0.9621 \;\; -2.9138 \;\; -3.2277 \;\; 0.0131 \;\; 31.0287]^T
\end{aligned}$$

Since $\mathbf{d}_{\tilde{\mathbf{x}}}^{(0)} \not\geq \mathbf{0}$ and $\mathbf{d}_{\tilde{\mathbf{s}}}^{(0)} \not\geq \mathbf{0}$, go to Step 6, we compute the step-lengths:

$$\beta_P^{(0)} = \frac{1}{max\{1, -(\mathbf{d}_{\tilde{\mathbf{x}}}^{(0)})_j/(0.99\tilde{x}_j^{(0)})\}} = 1$$

$$\beta_D^{(0)} = \frac{1}{max\{1, -(\mathbf{d}_{\tilde{\mathbf{s}}}^{(0)})_j/(0.99\tilde{s}_j^{(0)})\}} = 1$$

Hence, in Step 7, we compute a new solution $(\tilde{\mathbf{x}}^{(1)}; \tilde{\mathbf{w}}^{(1)}; \tilde{\mathbf{s}}^{(1)})$ by

$$\begin{aligned}
\tilde{\mathbf{x}}^{(1)} &= \tilde{\mathbf{x}}^{(0)} + \beta_P^{(0)}\mathbf{d}_{\tilde{\mathbf{x}}}^{(0)} \\
&= [10.9931 \;\; 11.1371 \;\; 6.8317 \;\; 6.5977 \;\; 1.0045 \;\; 36.89]^T \\
\tilde{\mathbf{w}}^{(1)} &= \tilde{\mathbf{w}}^{(0)} + \beta_D^0\mathbf{d}_{\tilde{\mathbf{w}}}^{(0)} \\
&= [9.5516 \;\; 9.3176 \;\; -0.811]^T \\
\tilde{\mathbf{s}}^{(1)} &= \tilde{\mathbf{s}}^{(0)} + \beta_D^0\mathbf{d}_{\tilde{\mathbf{s}}}^{(0)} \\
&= [2.5069 \;\; 2.3629 \;\; 6.6683 \;\; 6.9023 \;\; 34.955 \;\; 0.811]^T
\end{aligned}$$

Continuing this iterative process, the primal-dual algorithm will stop at the optimal solutions $\mathbf{x}^* = [10 \;\; 30 \;\; 0 \;\; 0]^T$ to the primal problem and $(\mathbf{w}^*; \mathbf{s}^*) = ([-2 \;\; -4]^T; [0 \;\; 0 \;\; 2 \;\; 4]^T)$ to the dual problem with objective value $= -380$.

□

CHAPTER III

IMPLEMENTATIONS, NUMERICAL RESULTS AND CONCLUSIONS

This chapter describes the design of software which is divided in two modules, the input module and the MPS module. The input module is admitted a linear programming problem from a keyboard via the user interface while the MPS module is admitted a problem from MPS files.

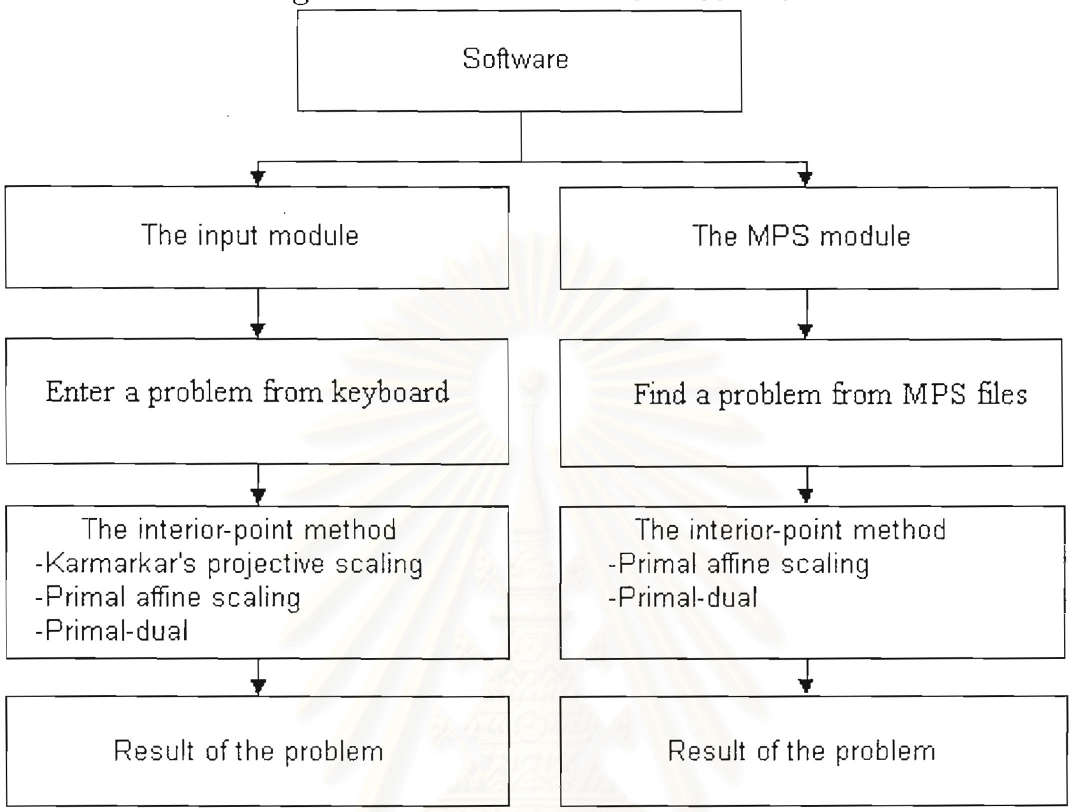
In our results, we test our software with small tested problems and MPS files. Then we compare the iteration numbers with LINDO and MINOS which base on the simplex method.

3.1 The Structure of Software

We divide the implementation into two modules, the input module and the MPS module. After reading the problem data, software will convert the problem into the standard form before solving it using the interior-point method.

In the input module, users can save the problem or retrieve the problem to be solved by the Karmarkar's projective scaling, the primal affine scaling and the primal-dual algorithms. While the MPS module can solve the problem in the MPS file format using only the primal-dual and the primal affine scaling algorithms.

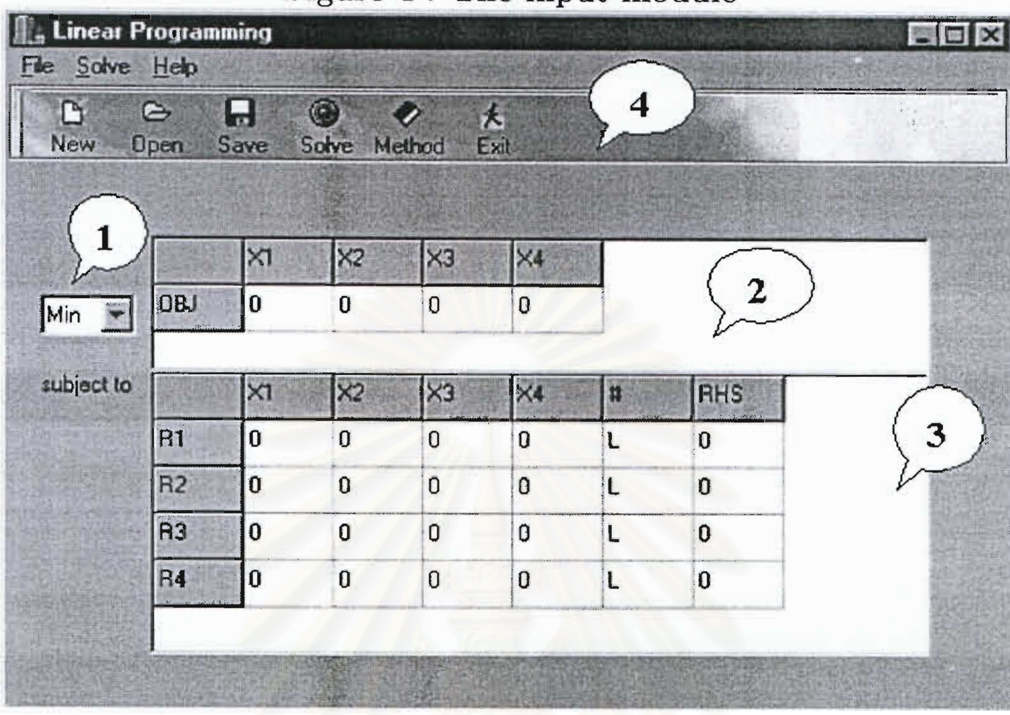
Figure 3 : The structure of software



3.1.1 The Input Module

The input module is shown in the Figure 4. The number 1 is the objective direction which users can choose to maximize or minimize, the number 2 is an input box for the linear objective function, the number 3 is an input box for all linear constraints and the number 4 is a tool bar. Users can save, open, solve problem and select the optimization algorithms from Karmarkar's projective scaling, the primal affine scaling and the primal-dual algorithms. The default algorithm is set as the primal-dual algorithm.

Figure 4 : The input module



The restriction of an input coefficient data

1. The default of all coefficients are zeroes and the default of row type is "L".

2. Coefficient value must be an integer or real number.

3. The available relation types are

Table 1 : The relation types

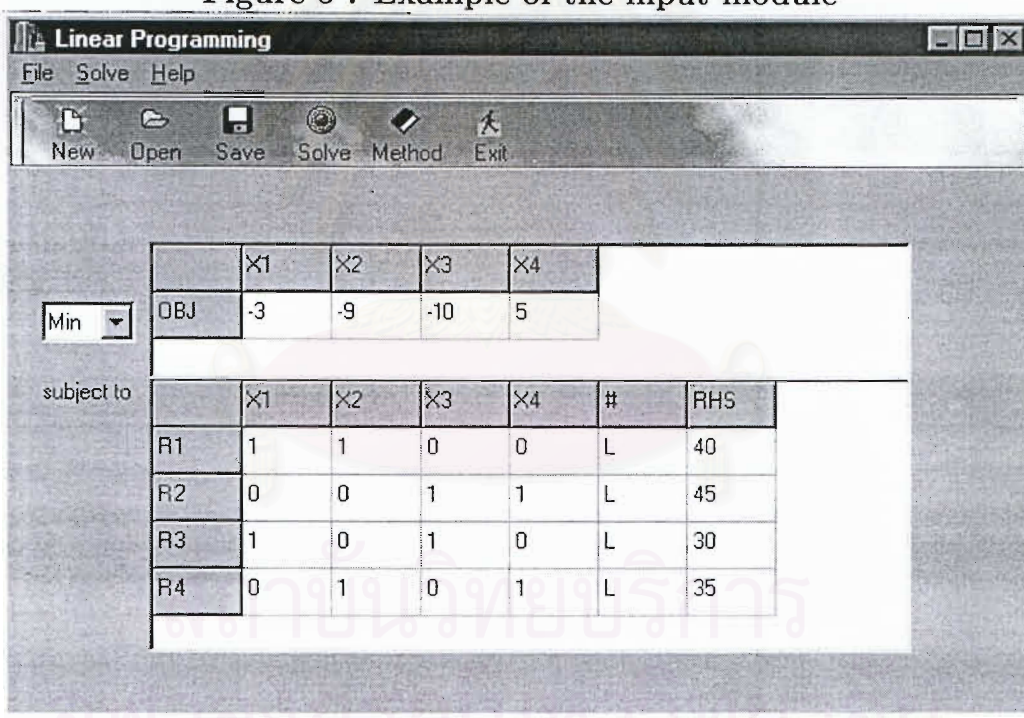
Type (#)	Indicator
"E" or "e"	equal to
"G" or "g"	greater than or equal to
"L" or "l"	less than or equal to

For example, if we have a linear programming problem as follows:

$$
\begin{aligned}
\text{minimize} \quad & -3x_1 - 9x_2 - 12x_3 + 5x_4 \\
\text{subject to} \quad & x_1 + x_2 \leq 40 \\
& x_3 + x_4 \leq 45 \\
& x_1 + x_3 \leq 30 \\
& x_2 + x_4 \leq 35 \\
& x_1, x_2, x_3, x_4 \geq 0
\end{aligned}
$$

then we can enter the problem in the input module as Figure 5.

Figure 5 : Example of the input module

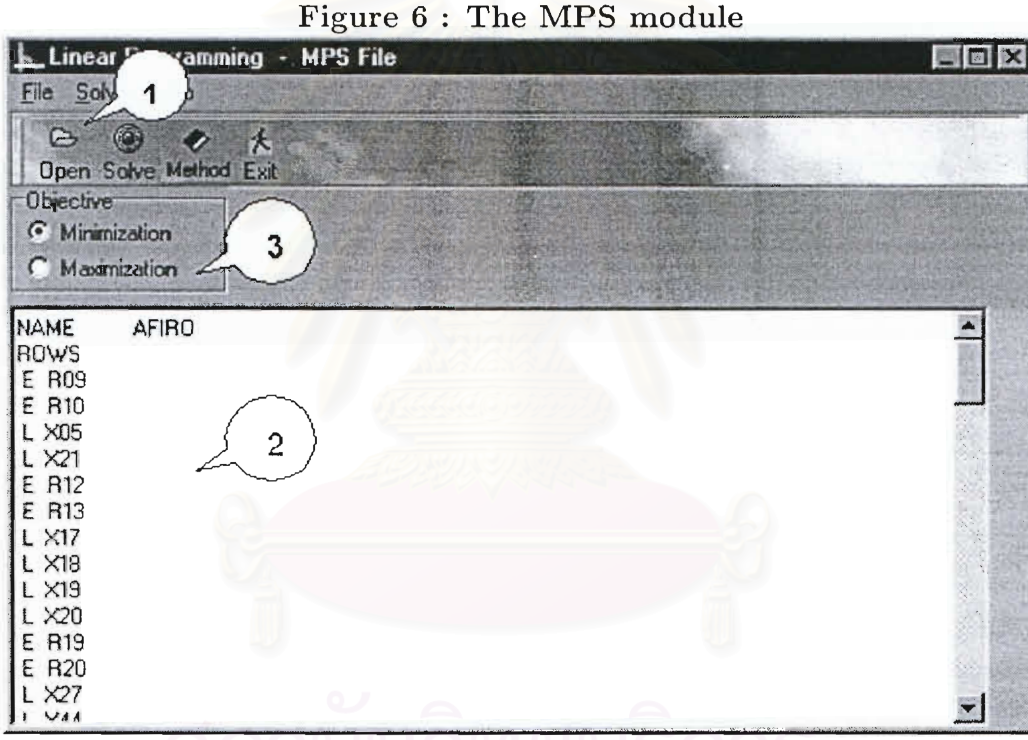


We click on the **solve** button. The software will scan an input data and solve the problem by the default algorithm (the primal-dual algorithm) or we can click on the **method** button to change the algorithm to Karmarkar's projective scaling algorithm or the primal affine scaling algorithm. Then we can save the problem to file by clicking on the **save** button. If we want to solve the new problem, we click on the **new** button to clear the input module.

3.1.2 The MPS Module

The MPS module is used for solving a linear programming problem in MPS file format which becomes the industry standard file format for a linear programming problem. The MPS module is shown in the Figure 6. The number 1 is an **open** button for opening the problem in MPS file format, the number 2 are details of the problem file and the number 3 is the objective of the problem.

Figure 6 : The MPS module



In this module, users can choose only two algorithms, the primal affine scaling and the primal-dual algorithms. For more details about MPS files, users should consult Appendix A.

3.2 Numerical Results

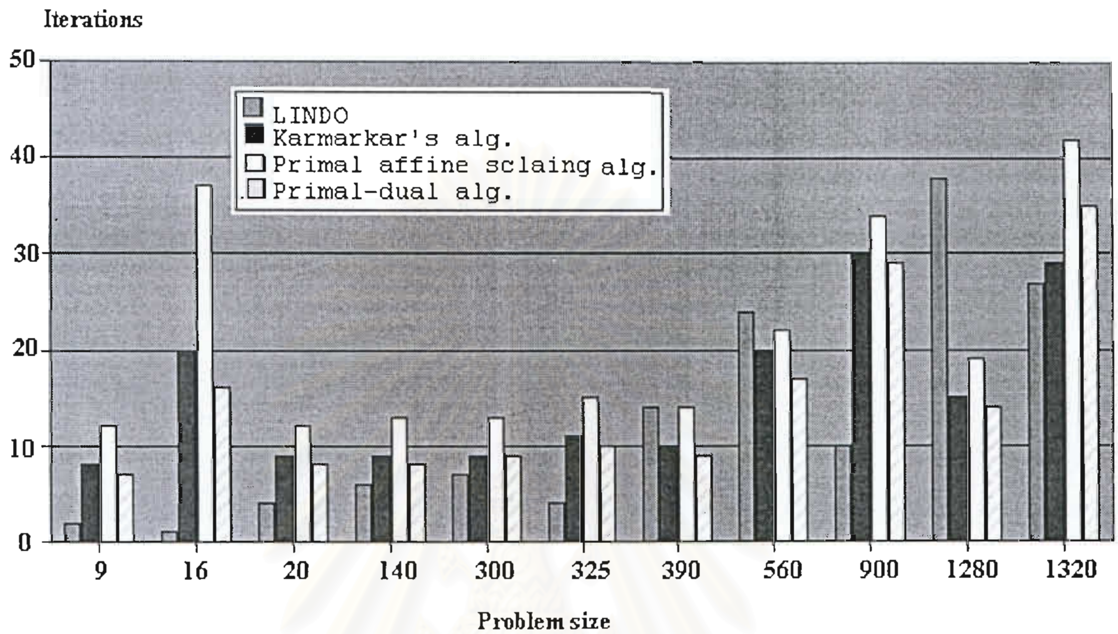
In this section we summarize the performance of the interior-point methods on problems from MPS files and our small tested problems.

The computational results were run under the Windows operating system with a Celeron 400 MHz processor and 64 MB of RAM. The source code was compiled using the Borland C++ Builder 5. We also compare the iteration numbers of our results with LINDO and MINOS 5.0 which are the popular used code of the simplex method. Table 2 shows the result from our small tested problems, we compare the iteration numbers of our results with LINDO program, where problem size is defined as the cross product of the number of rows and the number of columns of each problem.

Table 2 : The result of our small tested problems

Problem	Size	Iterations			
		LINDO	Karmarkar	Primal Affine	Primal-Dual
lp01	3×3	2	8	12	7
lp02	4×4	1	20	37	16
lp03	5×4	4	9	12	8
lp04	7×20	6	9	13	8
lp05	10×30	7	9	13	9
lp06	13×25	4	11	15	10
lp07	13×30	14	10	14	9
lp08	16×35	24	20	22	17
lp09	15×60	10	30	34	29
lp10	32×40	38	15	19	14
lp11	22×60	27	29	42	35

Figure 7 : Iteration comparisons of LINDO, Karmarkar's projective scaling, the primal affine scaling and the primal-dual algorithms



From the Figure 7, we see that LINDO uses less iterations than all the interior-point methods for most of our small tested problems. However, most of the real world problems are large. Therefore, an ordinary MPS file contains a large-scaled linear programming problem. We test with MPS files and compare the iteration numbers of our results from the primal affine scaling and the primal-dual algorithms with MINOS program.

Table 3 : The results of MINOS.

Problems	Sizes	Slack+Surplus variables	Nonzeros	MINOS Iterations	Optimum
Afiro	28×32	19	102	8	-4.6475314E+02
Adlittle	57×97	41	424	97	2.2549496E+05
Share2b	97×79	83	777	117	-4.1573224E+02
Scagr7	130×140	45	606	92	-2.3313898E+06
Share1b	118×225	28	1179	284	-7.6589319E+04
Israel	175×142	174	2443	327	-8.9664482E+05
Sc205	206×203	114	665	131	-5.2202061E+01
Beaconfd	174×262	33	3408	87	3.3592486E+04
Scsd1	78×760	0	2388	220	8.6666670E+00
E226	224×282	190	2768	686	-1.8751929E+01
Bandm	306×472	0	2494	463	-1.5862802E+02
Sctab1	301×480	180	1872	375	1.4122500E+03
Scsd6	148×1350	0	4316	550	5.0500000E+01
Scagr25	472×500	171	1725	92	-1.4753433E+07
Scrs8	491×1169	106	3288	933	9.0429696E+02

Table 4 : The primal affine scaling algorithm results

Problems	Sizes	Nonzeros	Iterations	Optimum
Afiro	28×32	102	36	-4.6475314E+02
Adlittle	57×97	424	90	2.2549496E+05
Share2b	97×79	777	54	-4.1573224E+02
Scagr7	130×140	606	90	-2.3313898E+06
Share1b	118×225	1179	73	-7.6589319E+04
Israel	175×142	2443	68	-8.9664482E+05
Sc205	206×203	665	34	-5.2202061E+01
Beaconfd	174×262	3408	63	3.3592486E+04
Scsd1	78×760	2388	28	8.6666670E+00
E226	224×282	2768	89	-1.8751928E+01
Bandm	306×472	2494	88	-1.5862802E+02
Sctab1	301×480	1872	95	1.4122500E+03
Scsd6	148×1350	4316	29	5.0499980E+01
Scagr25	472×500	1725	61	-1.4753433E+07
Scrs8	491×1169	3288	165	9.0429695E+02

Table 5 : The primal-dual algorithm results

Problems	Sizes	Nonzeros	Iterations	Optimum
Afiro	28×32	102	15	-4.6475314E+02
Adlittle	57×97	424	22	2.2549496E+05
Share2b	97×79	777	19	-4.1573224E+02
Scagr7	130×140	606	21	-2.3313898E+06
Share1b	118×225	1179	34	-7.6589319E+04
Israel	175×142	2443	35	-8.9664482E+05
Sc205	206×203	665	19	-5.2202061E+01
Beaconfd	174×262	3408	17	3.3592486E+04
Scsd1	78×760	2388	16	8.6666670E+00
E226	224×282	2768	27	-1.8751929E+01
Bandm	306×472	2494	23	-1.5862802E+02
Sctab1	301×480	1872	21	1.4122501E+03
Scsd6	148×1350	4316	18	5.0500000E+01
Scagr25	472×500	1725	27	-1.4753433E+07
Scrs8	491×1169	3288	27	9.0429695E+02

Figure 8 : Iteration comparisons of MINOS, the primal-dual and the primal affine scaling algorithms

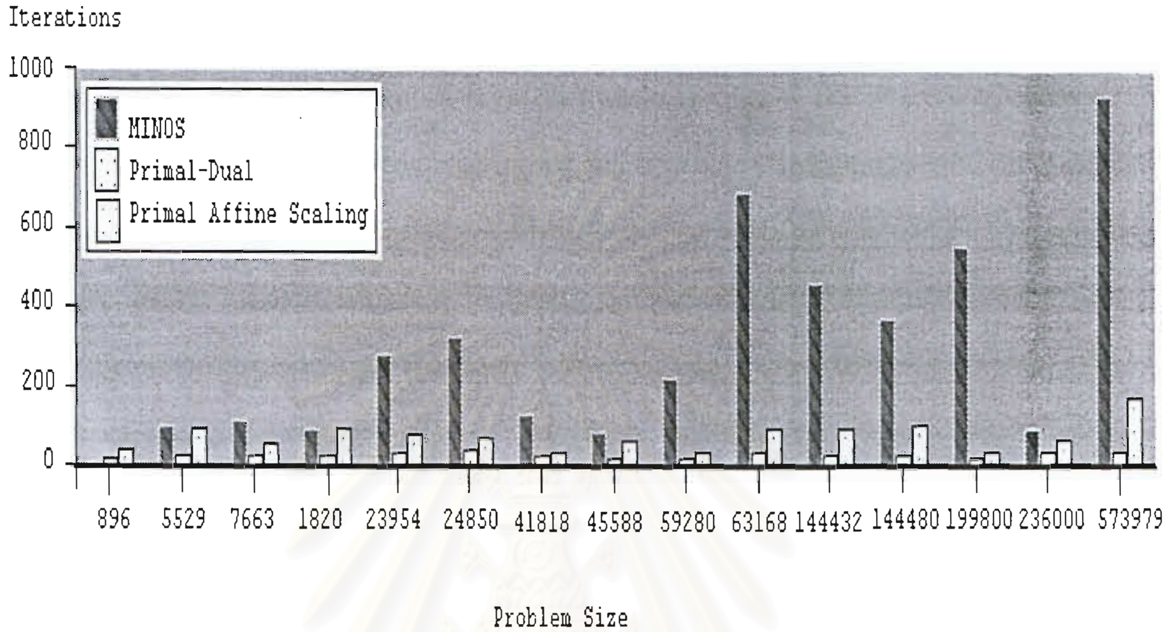


Figure 8 shows the iteration numbers of the primal affine scaling, the primal-dual algorithm and MINOS program. It's obvious that MINOS used more iterations than both of the primal affine scaling and the primal-dual algorithms. In addition, the primal-dual algorithm has the best performance among these three algorithms.

3.3 Conclusions

The time complexity of Karmarkar's projective scaling algorithm is polynomial [4]. However, in practice, it's difficult to convert the linear programming problem to Karmarkar's standard form. Moreover, the estimated problem size is double from the original problem.

The formulation of the primal affine scaling algorithm is simpler than that of the Karmarkar's projective scaling algorithm and the primal affine scaling algorithm also has a better performance than the Karmarkar's projective scaling algorithm in some large scale problems. In practice, it is not easy to find an initial solution ($\mathbf{x}^0$) which $\mathbf{A}\mathbf{x}^0 = \mathbf{b}$ of the primal affine scaling algorithm. Hence, the algorithm may take more iterations to find an initial solution. Currently, there is no proof that the primal affine scaling algorithm can run in polynomial-time [4].

The principle of the primal-dual algorithm uses both of the primal and dual variables which is more complicated than considering only the primal variables as in the primal affine scaling algorithm. However, our experiment shows that this algorithm has the best performance. Moreover, there has been a proof that the primal-dual algorithm run in polynomial-time [4,11]. Nowadays, this algorithm is the most widely used for solving the practical linear programming problems.

REFERENCES

1. Brixius, N., and Wright, S. J. **Interior-point methods online**[online].(n.d.). Available from: http://www-unix.mcs.anl.gov/otc/InteriorPoint [2000, Jan 17]

2. Chen, A. C. **Perturbed linear programming problems arising in process monitoring**. Doctoral dissertation, Department of Chemical Engineering, Graduate School, University of Notre Dame, 2001.

3. Dye, S. **Sample MPS input files**[online].(n.d.). Available from: http://www.mang.canterbury.ac.nz/people/shane/oslweb/features /feat24DT.htm#HDRABDATA[2000, Jan 12]

4. Fang, S., and Puthenpura, S. **Linear optimization and extensions**. Englewood Cliffs: Prentice Hall, 1993.

5. Goldberg, V. A., and others. Interior-point method in parallel computation. **IEEE.**(1989):350-355.

6. Lavelle, J. R. **MPS input files**[online].(n.d.). Available from: http://www-fp.mcs.anl.gov/otc/Guide/OptWeb/continuous/constrained/ linearprog/mps.html[2000, Jan 10]

7. Nash, S. G., and Sofer, A. **Linear and nonlinear programming**. Singapore : McGraw-Hill, 1996.

8. Ponnambalam, K., Quintana, V. H., and Vannelli, A. A fast algorithm for power system optimization problems using an interior-point method. **IEEE.** vol. 7(May,1992):892-899.

9. Winston, W. L. **Introduction to Mathematical programming.** California:Duxbury press, 1995.

10. Wright, S. J. **Interior-point research at Argonne**[online].(n.d.). Available from: http://www-unix.mcs.anl.gov/ wright/research.html [2000, Jan 25]

11. Wright, S. J. **Primal-dual interior-point methods**. Philadelphia: The Society for Industrial and Applied Mathematics, 1997.

APPENDICES

APPENDIX A

MATHEMATICAL PROGRAMMING SYSTEM FILES (MPS FILES)

MPS file is an input file format of a linear programming problem which has become the industry standard. The various sections of the data in MPS file are grouped according to the following order.

1. NAME: This section consists of the word "NAME" in columns 1-4, and the title of the problem in columns 15-22.

2. ROWS: This section defines row labels as well as the row type. The row type is entered in column 2 or 3 and the row label is entered in columns 5-12. This section of data is preceded by the word "ROWS" in columns 1-4, followed by a data for each row.

Table 6 : The code for specifying the row type

Row Type	Indicator
E	Equal to (=)
L	Less than or equal to ($\leq$)
G	Greater than or equal to ($\geq$)
N	Objective function

3. COLUMNS: This section defines the names of the variables, the coefficients of the objective function, and the nonzero coefficients of the linear constraints. The section is preceded by the word "COLUMNS" in columns 1-7, followed by the data. The data has the variable name in columns 5-12, the row label in columns 15-22, and the value of coefficient in columns 25-36.

The next row label can be inserted in columns 40-47 and the corresponding coefficient value in columns 50-61.

4. RHS: This section contains the nonzero elements of the right-hand side. The section is preceded by the word "RHS" in columns 1-3, followed by the data of the right-hand side. The data of the right-hand side has a label for the right-hand side in columns 5-12, the row label in columns 15-22, and the right-hand side value in columns 25-36. The next right-hand side may be in columns 40-47 and the corresponding right-hand side value in columns 50-61.

5. RANGES (optional): This section is for constraints of the following form $l_i \leq a_{i,1}x_1 + a_{i,2}x_2 + \ldots + a_{i,n}x_n \leq u_i$. That means both an upper and lower bound exist for the row. The range of the constraint is $r_i = u_i - l_i$. The value of u_i or l_i is specified in the RHS section data, and the value of r_i is specified in the RANGES section data. If b_i is the number entered in the RHS section and r_i is the number specified in the RANGES section, the u_i and l_i are defined as follows.

Table 7 : The code for specifying range type

Row Type	Lower Bound (l_i)	Upper Bound (u_i)
G	b_i	$b_i + r_i$
L	$b_i - r_i$	b_i

The section is preceded by the word "RANGES" in columns 1-6 . The data of the RANGES section has a label for the range in columns 5-12, the row label in columns 15-22, and the range value in columns 25-36. The next range of constraint may be in columns 40-47 and the corresponding range value in columns 50-61.

6. BOUNDS (optional): In this section, bounds on the variables are specified. The section is preceded by the word "BOUNDS" in columns 1-6. If the BOUNDS section is omitted, the usual bounds, $0 \leq x_i \leq \infty$, are assumed. The data of the BOUNDS section has the type of bound in columns 2-3, the bound row label in columns 5-12, the variable name in columns 15-22, and the bound value in columns 25-36.

Table 8 : The code for specifying bound type

Bound Type	Bound on the variable
LO	Lower Bound ($b_j \leq x_j$)
UP	Upper Bound ($x_j \leq b_j$)
FX	Fixed Variable ($x_j = b_j$)
FR	Free Variable ($-\infty \leq x_j \leq \infty$)
MI	Lower Bound $-\infty$ ($-\infty \leq x_j \leq 0$)
PL	Default Bound ($0 \leq x_j \leq \infty$)

7. ENDATA: This section is in columns 1-6, the end of the data input.

Here is a little sample problem written in MPS format (explained in more detail below):

```
NAME    EXAMPLE01
ROWS
 N   COST
 L   ROW1
 G   ROW2
 E   ROW3
```

```
COLUMNS
    X1    COST    1    ROW1    1
    X1    ROW2    1
    X2    COST    4    ROW1    1
    X2    ROW3    -1
    X3    COST    9    ROW2    1
    X3    ROW3    1
RHS
    RHS1    ROW1    5    ROW2    10
    RHS1    ROW3    7
ENDATA
```

For comparison, here is the same problem written out in an equation-oriented format:

Optimize

$X_1 + 4X_2 + 9X_3 \quad \text{(COST)}$

subject to

$X_1 + X_2 \leq 5 \quad \text{(ROW1)}$

$X_1 + X_3 \geq 10 \quad \text{(ROW2)}$

$-X_2 + X_3 = 7 \quad \text{(ROW3)}$

$X_1, X_2, X_3 \geq 0$

APPENDIX B

THE SMALL TESTED PROBLEMS

lp01

maximize $3x_1 + 4x_2 + 2x_3$

subject to $3x_1 + 2x_2 + 4x_3 \leq 15$

$x_1 + 2x_2 + 3x_3 \leq 7$

$2x_1 + x_2 + x_3 \leq 6$

$x_1, x_2, x_3 \geq 0$

lp02

maximize $1000x_1 + 100x_2 + 10x_3 + x_4$

subject to $x_1 \leq 1$

$20x_1 + x_2 \leq 100$

$200x_1 + 20x_2 + x_3 \leq 10000$

$2000x_1 + 200x_2 + 20x_3 + x_4 \leq 1000000$

$x_1, x_2, x_3, x_4 \geq 0$

lp03

maximize $x_1 + 3x_2 + 5x_3 + 2x_4$

subject to $2x_1 + x_2 \leq 9$

$5x_1 + 3x_2 + 4x_3 \geq 10$

$x_1 + 4x_2 \leq 8$

$x_3 - 5x_4 \leq 4$

$x_3 + x_4 \leq 10$

$x_1, x_2, x_3, x_4 \geq 0$

lp04

minimize $10x_1 + 9x_2 + 8x_3 + 7x_4 + 6x_5 + 5x_6 + 4x_7 + 3x_8 + 2x_9 + x_{10}$

$+x_{11} + 3x_{12} + 5x_{13} + x_{14} + 4x_{15} + 6x_{16} + x_{17} + x_{18} + x_{19} + x_{20}$

subject to $x_1 + x_2 + x_3 \geq 10$

$x_4 + x_5 + x_6 \geq 10$

$x_7 + x_8 + x_9 \geq 10$

$x_{10} + x_{11} + x_{12} \geq 20$

$x_{13} + x_{14} + x_{15} \geq 20$

$x_{16} + x_{17} + x_{18} \geq 20$

$x_1 + x_4 + x_7 + x_{10} + x_{13} + x_{16} + x_{20} \geq 100$

$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \geq 0$

lp05

minimize $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}$

$+x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19}$

$+x_{20} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{30}$

subject to $5x_1 + x_{11} + +4x_{21} \geq 100$

$x_2 + 8x_{12} + x_{22} \geq 100$

$x_3 + 2x_{13} + x_{23} \geq 100$

$-x_4 + x_{14} + 3x_{24} \geq 100$

$x_5 + 7x_{15} + x_{25} \geq 100$

$3x_6 + x_{16} + x_{26} \geq 100$

$x_7 - 3x_{17} + 2x_{27} \geq 100$

$x_8 + x_{18} + x_{28} \geq 100$

$4x_9 + x_{19} + x_{29} \geq 100$

$x_{10} + 7x_{20} + 2x_{30} \geq 100$

$x_1, x_2, ..., x_{30} \geq 0$

lp06

minimize $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}$

$+x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{20}$

$+x_{21} + x_{22} + x_{23} + x_{24} + x_{25}$

subject to $x_1 + 2x_2 + 3x_3 - x_4 \leq 30$

$4x_2 + x_4 - x_6 + x_8 - 2x_{10} \leq 50$

$x_1 - x_3 + x_5 + x_7 - x_9 \geq 10$

$3x_{10} + 2x_{12} - 3x_{13} + x_{14} + x_{15} + 2x_{16} + x_{18} + x_{20} \leq 100$

$x_{11} - 3x_{13} + x_{15} - x - 9 - 2x_7 + x_{17} + 3x_{19} + x_{25} \leq 80$

$x_{12} + x_{22} + x_{21} + x_{23} - 3x_{24} - x_{25} + 5x_1 \leq 99$

$x_1 + 5x_2 + x_3 \geq 20$

$x_{25} + x_{20} + x_{10} \geq 12$

$x_3 + 2x_6 + x_9 + x_{12} + x_{15} - 9x_{18} + x_{21} + x_{24} \geq 50$

$x_2 + 7x_4 + x_8 + x_{12} + x_{22} + 2x_{24} \geq 40$

$x_1 - x_5 + x_7 + x_{11} + x_{13} + 20x_{17} + x_{23} + x_{25} = 24$

$x_6 + x_7 = 15,$

$x_{10} + x_{20} = 26$

$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \geq 0$

$x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20} \geq 0$

$x_{21}, x_{22}, x_{23}, x_{24}, x_{25} \geq 0$

lp07

minimize $7x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + 3x_8 + x_9 + x_{10}$

$+x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{20}$

$+x_{21} + x_{22} + 5x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + 6x_{29} + x_{30}$

subject to $-2x_1 + x_{11} + x_{21} \geq 100$

$x_2 + 4x_{12} + x_{22} \geq 100$

$x_3 + 5x_{13} + 7x_{23} \geq 100$

$x_4 + x_{14} + x_{24} \geq 100$

$x_5 + 8x_{15} + x_{25} \geq 100$

$3x_6 + x_{16} + x_{26} \geq 100$

$x_7 + x_{17} - 3x_{27} \geq 100$

$x_8 + 2x_{18} + 5x_{28} \geq 100$

$6x_9 - x_{19} + 2x_{29} \geq 100$

$x_{10} + x_{20} + 2x_{30} \geq 100$

$x_1 + 17x_2 + x_3 + x_4 + 25x_5 + x_6 + x_7 + 11x_8 + 3x_9 + x_{10} \geq 500$

$x_{11} + x_{12} + x_{13} + 9x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + 8x_{20} \geq 500$

$x_{21} + x_{22} + 2x_{23} + x_{24} + x_{25} + x_{26} + 31x_{27} + x_{28} + 7x_{29} + x_{30} \geq 500$

$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \geq 0$

$x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20} \geq 0$

$x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30} \geq 0$

lp11

minimize $29x_1 + 4.65x_2 + 4x_3 + 9.9x_4 + 0.6x_5 + 8x_6 + 92x_7 + 6x_8$

$+3x_9 + 19.5x_{10} + 18x_{11} + 18x_{12} + 150x_{13} + 0.5x_{14} + 17x_{15} + 140x_{16}$

subject to $2977.0004x_1 + 3470.0003x_2 + 3376.0002x_3 + 2358.5x_4 + 2115.9x_6$

$+1011x_9 + 2250x_{10} + 2182.5768x_{12} + 3811.0005x_{15} \geq 1500$

$0.6005x_1 + 0.085x_2 + 0.088x_3 + 0.44x_4 + 0.414x_6 + 0.956x_7$

$+0.253x_8 + 0.17x_9 + 0.55x_{10} + 0.62x_{15} + 0.587x_{16} \geq 18$

$0.094x_1 + 0.02492x_2 + 0.029x_3 + 0.008x_4 + 0.0108x_6 + 0.062x_8$

$+0.025x_9 + 0.055x_{10} + 0.98505x_{12} + 0.025x_{15} \geq 3$

$0.007x_1 + 0.02492x_2 + 0.023x_3 + 0.07x_4 + 0.136x_6 + 0.153x_8$

$+0.241x_9 + 0.01x_{10} + 0.013x_{15} \leq 7$

$0.007x_1 + 0.02492x_2 + 0.023x_3 + 0.07x_4 + 0.136x_6 + 0.153x_8$

$+0.241x_9 + 0.01x_{10} + 0.013x_{15} \geq 0$

$0.0511x_1 + 0.000356x_2 + 0.0004x_3 + 0.0029x_4 + 0.395x_5 + 0.0015x_6$

$+0.0029x_8 + 0.0144x_9 + 0.056x_{10} + 0.16x_{11} + 0.003x_{14} \geq 3.4$

$0.0511x_1 + 0.000356x_2 + 0.0004x_3 + 0.0029x_4 + 0.395x_5 + 0.0015x_6$

$+0.0029x_8 + 0.0144x_9 + 0.056x_{10} + 0.16x_{11} + 0.003x_{14} \leq 5$

$0.028796x_1 + 0.0008x_2 + 0.0027x_4 + 0.0022x_6 + 0.0022x_9$

$+0.026x_{10} + 0.21x_{11} + 0.0014x_{15} \geq 0.45$

$17x_2 + 262.9999x_9 + 191x_{15} \geq 5$

$0.0451x_1 + 0.0026x_2 + 0.0021x_3 + 0.0269x_4 + 0.0176x_6 + 0.784x_7$

$+0.009x_8 + 0.0073x_9 + 0.0425x_{10} + 0.0103x_{15} \geq 0.7$

$0.0451x_1 + 0.0026x_2 + 0.0021x_3 + 0.0269x_4 + 0.0176x_6 + 0.784x_7$

$+0.009x_8 + 0.0073x_9 + 0.0425x_{10} + 0.0103x_{15} \leq 100$

$$0.0163x_1 + 0.0018x_2 + 0.0016x_3 + 0.0062x_4 + 0.0051x_6 + 0.0057x_8 + 0.0024x_9$$
$$+0.0185x_{10} + 0.0149x_{15} + 0.98x_{16} \geq 0.3$$
$$0.0163x_1 + 0.0018x_2 + 0.0016x_3 + 0.0062x_4 + 0.0051x_6 + 0.0057x_8 + 0.0024x_9$$
$$+0.0185x_{10} + 0.0149x_{15} + 0.98x_{16} \leq 100$$
$$0.022x_1 + 0.0036x_2 + 0.0033x_3 + 0.0128x_4 + 0.0113x_6 + 0.0096x_8 + 0.0043x_9$$
$$+0.035x_{10} + 0.0259x_{15} + 0.98x_{16} \geq 0.55$$
$$0.0049x_1 + 0.0006x_2 + 0.008x_3 + 0.0074x_4 + 0.0052x_6 + 0.0034x_8 + 0.0023x_9$$
$$+0.0042x_{10} + 0.0036x_{15} \geq 0.18$$
$$x_2 + x_3 + x_4 + x_5 + x_6 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} = 100$$
$$0 \leq x_1 \leq 12,\ 0 \leq x_2 \leq 100,\ 0 \leq x_3 \leq 25,\ 0 \leq x_4 \leq 100$$
$$0 \leq x_5 \leq 100,\ 0 \leq x_6 \leq 8,\ 0 \leq x_7 \leq 100$$
$$0 \leq x_8 \leq 6,\ 0 \leq x_9 \leq 4,\ 0 \leq x_{10} \leq 12,\ x_{10} \geq 3$$
$$0 \leq x_{11} \leq 100,\ 0 \leq x_{12} \leq 3,\ 0.5 \leq x_{13} \leq 0.51$$
$$0 \leq x_{14} \leq 0.35,\ 0 \leq x_{15} \leq 10,\ 0 \leq x_{16} \leq 100$$

lp09

minimize $x_1 + 7x_2 + x_3 + 4x_4 + x_5 - x_6 + x_7 - x_8 + x_9 + x_{10}$

$+x_{11} + 3x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + 9x_{17} + x_{18} + x_{19} + 2x_{20}$

$+x_{21} + 2x_{22} + x_{23} + x_{24} + x_{25} - x_{26} + x_{27} + x_{28} + 3x_{29} + x_{30}$

$+x_{31} + x_{32} + x_{33} + x_{34} + 7x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{40}$

$+x_{41} + x_{42} + 3x_{43} + x_{44} + x_{45} + x_{46} + 4x_{47} + x_{48} + x_{49} + x_{50}$

$+3x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + 5x_{56} + x_{57} + x58 + x_{59} + 11x_{60}$

subject to $x_1 + 3x_2 + x_3 + x_4 + x_5 + x_6 + 9x_7 + x_8 + x_9 + x_{10} \leq 1000$

$3x_{11} + x_{12} + x_{13} + x_{14} + 11x_{15} + 2x_{16} + x_{17} + 8x_{18} + x_{19} + x_{20} \leq 1000$

$5x_{21} + x_{22} + 7x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{30} \leq 1000$

$x_{31} + x_{32} + x_{33} + x_{34} + 20x_{35} + 3x_{36} + x_{37} + x_{38} + 15x_{39} + x_{40} \leq 1000$

$3x_{41} + x_{42} + 6x_{43} + x_{44} + x_{45} + x_{46} + x_{47} + x_{48} + 3x_{49} + x_{50} \leq 1000$

$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + x_{57} + x_{58} + x_{59} + 6x_{60} \leq 1000$

$2x_5 + 8x_{15} + x_{25} + 7x_{35} + x_{45} + x_{55} \geq 500$

$3x_{11} + x_{12} + x_{23} + x_{34} + x_{45} + x_{56} \geq 2500$

$x_1 + 7x_{11} + 4x_{21} + x_{31} + x_{41} + x_{51} \geq 1000$

$x_{11} + x_{31} + x_{41} \leq 800$

$3x_{10} + x_{20} + x_{30} + x_{40} + x_{50} + x_{60} \geq 500$

$x_{12} - x_{23} - x_{24} + 6x_3 + x_{52} + x_{48} - x_{36} + x_{11} \geq 2000$

$x_2 + x_6 \geq 500$

$3x_6 - x_1 + x_{10} - x_{40} + x_{59} \geq 1000$

$x_6 + x_7 + x_8 + x_9 \geq 750$

$x_1, x_2, x_3, ..., x_{60} \geq 0$

lp10

maximize $3x_1 - 8x_2 + x_3 + x_4 + x_5 + x_6 + 3x_7 + x_8 + 5x_9 + x_{10}$

$+4x_{11} + x_{12} + 7x_{13} + x_{14} - x_{15} + 2x_{16} + x_{17} - 7x_{18} + x_{19} + x_{20}$

$+x_{21} + 2x_{22} + 2x_{23} + 12x_{24} + x_{25} + x_{26} + 6x_{27} + x_{28} + x_{29} + 4x_{30}$

$+x_{31} - x_{32} + x_{33} - x_{34} + x_{35} + x_{36} + 18x_{37} + x_{38} + 2x_{39} + x_{40}$

subject to $2x_1 + x_{11} + x_{21} + x_{31} <= 150$

$x_2 + x_{12} + x_{22} + x_{32} \leq 150$

$x_3 + 2x_{13} + 6x_{23} + x_{33} \leq 150$

$x_4 + 2x_{14} + 3x_{24} + x_{34} \leq 150$

$3x_5 + x_{15} + x_{25} + x_{35} \leq 150$

$x_6 + 2x_{16} + 3x_{26} + x_{36} \leq 150$

$x_7 + x_{17} + 7x_{27} + x_{37} \leq 150$

$x_8 + x_{18} + 10x_{28} + x_{38} \leq 150$

$4x_9 + x_{19} + x_{29} + x_{39} \leq 150$

$x_{10} + 2x_{20} + x_{30} + 5x_{40} \leq 150$

$x_1 + 4x_{12} + 7x_{23} + x_{34} \leq 150$

$x_{34} + 3x_{25} + x_{16} + 5x_7 \leq 150$

$x_7 + x_{18} + 4x_{29} + x_{40} \leq 150$

$x_{20} + x_{21} + x_{22} \geq 50$

$x_{10} + x_{11} + x_{12} \geq 50$

$x_{30} + x_{31} + x_{32} \geq 50$

$x_1 + x_5 + x_{18} + x_{39} + x_{20} \geq 300$

$x_1 - x_4 + 2x_{35} - x_{21} \geq 300$

$x_1 \geq 10, x_2 \geq 10, x_3 \geq 10, x_{24} \geq 10, x_{39} \geq 19, x_{38} \geq 17$

$x_5 \geq 10, x_{16} \geq 10, x_{27} \geq 10, x_8 \geq 10, x_{40} \leq 50, x_{37} \geq 15$

$x_{19} \geq 10, x_{10} \geq 10, x_{37} \geq 15, \quad x_1, x_2, x_3, ..., x_{40} \geq 0$

lp11

maximize $x_1 + 4x_2 - 7x_3 + 2x_4 + x_5 + x_6 + x_7 + 4x_8 + x_9 - 3x_{10}$

$+x_{11} + 5x_{12} + x_{13} + 11x_{14} - 2x_{15} + x_{16} + x_{17} + x_{18} + 7x_{19} + 2x_{20}$

$+8x_{21} + 12x_{22} - 5x_{23} + x_{24} + x_{25} - x_{26} + x_{27} + 2x_{28} + x_{29} + x_{30}$

$+4x_{31} + x_{32} + x_{33} - 14x_{34} + x_{35} + x_{36} - 9x_{37} + 3x_{38} + x_{39} + 7x_{40}$

$+7x_{41} + x_{42} + x_{43} + x_{44} + 3x_{45} + x_{46} + x_{47} - 6x_{48} + x_{49} + x_{50}$

$+20x_{51} + 8x_{52} + x_{53} + 4x_{54} - x_{55} - 2x_{56} + x_{57} + x_{58} + x_{59} + 7x_{60}$

subject to $-2x_1 + x_2 + x_3 + x_4 + x_5 <= 500$

$x_6 + x_7 + x_8 + x_9 + x_{10} <= 1100$

$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} <= 1500$

$x_{16} + x_{17} + x_{18} + 2x_{19} + x_{20} <= 1300$

$x_{21} + x_{22} + 2x_{23} + x_{24} + x_{25} <= 1250$

$2x_{26} + x_{27} + x_{28} + x_{29} + x_{30} <= 1400$

$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} <= 2500$

$x_{36} + x_{37} + x_{38} + 3x_{39} + 3x_{40} <= 1200$

$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} <= 500$

$x_{46} + x_{47} + x_{48} + x_{49} + x_{50} <= 3300$

$x_{51} + 2x_{52} + x_{53} + x_{54} + x_{55} <= 1750$

$x_{56} + x_{57} + 4x_{58} + x_{59} + x_{60} <= 1000$

$x_1 - x_7 - x_{12} + x_{23} >= 1500,\ x_4 + x_{39} >= 125$

$x_{60} - x_{50} + x_{27} + x_{23} >= 1450,\ x_1 + x_{55} >= 550$

$x_{14} + x_{45} + x_{23} - x_1 >= 1000,\ x_2 + x_3 + 2x_{58} >= 750$

$x_{09} + x_{10} >= 120,\ x_{11} + x_5 - x_1 + x_{43} + 4x_{22} >= 1800$

$x_{16} + x_{41} + x_{57} >= 330,\ x_2 + x_{17} + x_{33} >= 450$

$x_1, x_2, x_3, ..., x_{60} >= 0$

VITA

Name : Mr.Sa-at Moungjun

Degree : Bachelor degree of Science in Mathematics, 1999,
Khon Kaen University, Thailand.

Scholarship : The Secondary Education Quality Improvement (SEQI) Project.