Spring 2004

# Parallel Decomposition Procedures for Large-scale Linear Programming Problems

Yusong Hu
*Old Dominion University*

# PARALLEL DECOMPOSITION PROCEDURES FOR LARGE-SCALE LINEAR PROGRAMMING PROBLEMS

by

Yusong Hu
M.S. April 1992, Northern Jiaotong University, Beijing, China

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

CIVIL ENGINEERING

OLD DOMINION UNIVERSITY
May 2004

Approved by:

Duc T. Nguyen (Director)

Zia Razzaq (Member)

Laura J. Harrell (Member)

Hideaki Kaneko (Member)

# ABSTRACT

## PARALLEL DECOMPOSITION PROCEDURES FOR LARGE-SCALE LINEAR PROGRAMMING PROBLEMS

Yusong Hu
Old Dominion University, 2004
Director: Dr. Duc T. Nguyen

In practice, many large-scale linear programming problems are too large to be solved effectively due to the computer's speed and/or memory limitation, even though today's computers have many more capabilities than before. Algorithms are exploited to solve such large linear programming problems, either in the sequential or parallel computation environment. This study focuses on two parallel algorithms for solving large-scale linear programming problems efficiently.

The first parallel decomposition algorithm discussed in this study is from the theory

problems in a special block-angular structure. The theory of the decomposition principle is first examined. Since the subproblems of a linear programming problem can be in any of the three possible cases – optimal solution case, unbounded solution case and no solution case, examples are provided for solving the problem when its subproblems are in any of these cases. The concept of extreme directions is discussed due to its direct connection with the unbounded solution case. A parallel computation code, which can handle all these cases, is implemented in this study with the decomposition principle theory and its performance is tested for large-scale linear programming problems.

Only the problems in the special block-angular structure can be solved with the decomposition principle. For general linear programming problems, this study proposed a

new decomposition algorithm named "division by the interior point". The idea of this new algorithm is as follows: with a found interior point inside the feasible region, divide the feasible region into multiple subregions and use multiple processors to solve the problem in each subregion. This new algorithm is first demonstrated with a few small numerical examples. A parallel computation code in this new idea is implemented and tested with large-scale linear programming problems.

This thesis is dedicated to

my parents,

my wife, Wei,

and my son, Sam.

# ACKNOWLEDGMENTS

There are many people who have contributed to the successful completion of this dissertation. I extend many, many thanks to my committee members for their patience and hours of guidance on my research. The untiring efforts of my major advisor, Dr. Duc T. Nguyen, and helpful comments for Dr. Gene Hou, deserve special recognition. I would like to thank Mr. Siroj Tungkahotara for providing subroutines and test results related to Chapter V. I would also like to thank the library of the University of Maryland at College Park for allowing me to borrow several useful textbooks.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

Figure Page

# CHAPTER I

# INTRODUCTION

Linear programming is a branch of applied mathematics that deals with methods of optimizing a linear objective function of a set of decision variables subject to linear constraints. Since George B. Dantzig proposed the simplex method in 1947 [1], linear programming has been extensively used in the industry, military, government, urban planning, etc. In a recent survey of Fortune 500 companies, 85% of those who responded said that they had used linear programming algorithms and/or software [2].

## 1.1 Overview

The standard form of linear programming problems is in the following format [1]:

$$\text{Minimize } z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{1.1}$$

$$\text{subject to} \quad a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \tag{1.2}$$

$$\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m$$

$$(x_1, x_2, \ldots, x_n \geq 0) \tag{1.3}$$

Or, in a simpler matrix notation, it can be written as

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} \tag{1.4}$$

$$\text{subject to} \quad \mathbf{Ax} = \mathbf{b} \tag{1.5}$$

$$(\mathbf{x} \geq \mathbf{0}) \tag{1.6}$$

where **x** and **c** are vectors of size n, **b** is a vector of size m, and **A** is an mxn matrix.

This matrix notation of the standard form is used throughout this study, although in some of the problems, maximization of the objective function is used instead of minimization. It is trivial to convert maximization to minimization:

$$\text{Maximize } \mathbf{c}^T\mathbf{x} = -\,(\text{Minimize } \mathbf{c}^T\mathbf{x})$$

Since 1947, the simplex method has dominated the linear programming field with its proven capability of solving real world problems, although in theory this method may have some difficulty. In 1984, N. Karmarkar made a real breakthrough in linear programming with his interior point method [3]. Since in theory this new method is superior to the simplex method, it has become the research focus in the past years. Both the simplex method and the interior point method are used in this study, while more discussion is devoted to the newer interior point method because it has been less experimented.

## 1.2 Objective and Scope

Both the simplex method and the interior point method perform well for solving small to medium size problems. However, they may not be able to solve large-scale problems fast enough due to the computer's computational speed. When the problems are too large, they may not be solved at all due to the limitation of computer memory. The

subproblems". The objective of this study is to solve large-scale linear programming problems efficiently with decomposition procedures using parallel computation. First, in this study, the decomposition principle procedure proposed by Dantzig and Wolfe [4] is examined (see Chapter 3). This technique has been of particular interest to researchers. However, the research that has been done is mostly in the sequential computation environment. In this study, a parallel decomposition computation code is implemented and tested with large-scale linear programming problems for efficiency (see Chapter 4). Since the procedure of the decomposition principle is customized to the "block angular" problems, it can only achieve satisfactory result for those special problems. For general large-scale linear programming problems, a new parallel decomposition algorithm is proposed in Chapter V and tested with numerical examples.

Since the idea of these two decomposition approaches comes right from the simplex method and/or the interior point method, these two methods are reviewed briefly in Chapter II to facilitate the future discussions. Chapter II also discusses one simple technique to find a starting interior point, which can be used in the "division by the interior point" decomposition procedure proposed in Chapter V.

It is interesting to note that both the names of the simplex method and the interior point method come from the geometry. Indeed, the intuition that is generated from the geometry of linear programming is one of the keys to understand the linear programming theory. The idea of the new decomposition procedure of Chapter 5 is also inspired by the geometric properties of linear programming. In Chapter 3, one geometric concept of linear programming, extreme directions, is discussed before the discussion of the decomposition principle procedure, because it is essential for solving the linear programming problems of the unbounded solution case.

# CHAPTER II

# LINEAR PROGRAMMING METHODS

The parallel algorithms for linear programming problems presented in this study are based upon two linear programming methods: the simplex method and the interior point method. This chapter will review these two methods in order to make future discussion about the parallel algorithm easier. While the simplex method is the basic method of linear programming and is introduced in every linear programming book, the interior point method is relatively newer and is discussed in much less detail. Hence, this chapter focuses on the interior point method. The simplex method is reviewed first only for its key ideas, in order to compare the difference between the interior point method and the simplex method.

## 2.1 The Simplex Method

a linear programming problem is not empty, it has either unbounded solution or an optimal solution on one of its extreme points. Thus, the simplex method only iterates on the extreme points. The procedure of the simplex method is as follows[2]

(1) Find a starting extreme point. Two commonly used methods, the two-phase method and the big-M method, can be used to find such a starting extreme point.

(2) Check if the current extreme point is optimal. If yes, stop the iteration. Otherwise go to step (3). The current solution is optimal if the objective cost function can no longer be improved.

(3) Move to another extreme point with improved objective value. Then return to step (2). The pivoting process is used to find such an extreme point.

For hand calculation, the simplex method can be done in the "simplex tableau" format. A numerical example solved in this procedure is given in Section 3.3.1. The simplex method can be implemented in more efficient approaches, such as the revised simplex method. The revised simplex method is more efficient because with matrix formulation, efficient linear algebra (such as linear equation solver) can be easily exploited [5]. All the numeric examples in Section 4.1 are solved with the revised simplex method.

## 2.2 The Interior Point Method

In the Fall of 1984, N. K. Karmarkar of AT&T Bell Laboratories proposed a new algorithm [3] for linear programming. This new algorithm was the first one in thirty years that not only outperforms the simplex method in theory, but also shows the potential to rival the simplex method for solving large-scale practical applications.

Karmarkar's method is radically different from the simplex method. The simplex method starts with a vertex (extreme point) of the feasible region and moves along the boundary to a better neighboring vertex, until the optimal solution or infeasibility is

the feasible region to visit every vertex in the worst-case scenario. For large-scale problems, the feasible region contains numerous extreme points, which can incur a huge number of iterations.

Karmarkar's approach starts with an interior point in the feasible region and moves through the interior region to reach the optimal point. This approach is based on two fundamental insights:

1. If the current interior solution is near the center of the polytope, it makes sense to move in the direction of steepest descent of the objective function to achieve a better value.
2. Without changing the problem in any essential way, an appropriate transformation

can be applied to the solution space so as to place the current interior solution near the center of the transformed solution space.

The basic strategy of Karmarkar's algorithm is: take an interior solution, transform the solution space so as to place the current solution near the center of the transformed space, and then move in the direction of the steepest descent in the transformed space, but not all the way to the boundary in order to remain as an interior solution. Then take the inverse transformation to map the improved solution back to the original solution space as a new interior solution. Repeat this procedure until the stopping criterias are met.

The transformation proposed in the original Karmarkar's algorithm is a projective transformation, thus Karmarkar's algorithm is also referred as projective scaling algorithm. A LP problem must satisfy the following requirements before it can be solved using the projective scaling algorithm:

1. The problem has to be in the following standard form:

Minimize $\mathbf{c}^T\mathbf{x}$ (2.1)

Subject to $\mathbf{Ax} = \mathbf{0}$ (2.2)

$\mathbf{e}^T\mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}$ (2.3)

Also, an initial feasible interior solution (starting point) must be known.

2. The optimal objective function value must be zero.

Since it is relatively cumbersome to transform a standard LP problem to Karmarkar's format, many variants of Karmarkar's algorithm have been developed. Among these methods, the affine scaling algorithm [2] [6] received the widest analysis and experimentation. The interior point method used in this study is the affine scaling algorithm.

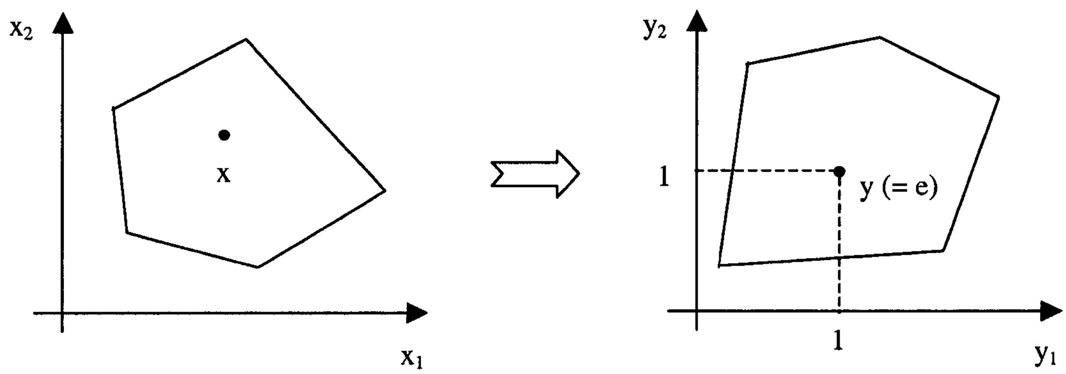
### 2.2.1 Affine Scaling Algorithm

Affine scaling algorithm was named because the transformation used in this algorithm is affine scaling transformation.

For an interior point **x**, we define an n x n diagonal matrix $\mathbf{X}_k$, which has all zero elements except that the diagonal elements $\mathbf{X}_{kii} = x_i$. With $\mathbf{X}_k$, we have the following transformation:

$$\mathbf{y} = \mathbf{X}_k^{-1}\,\mathbf{x} \qquad (2.4)$$

Notice that this transformation does nothing but to rescale $x_i$ by the factor $1/x_i$. It was named the affine scaling transformation because geometrically it maps a straight line in one space to another straight line in another space, as shown in Figure 1:




As we can see from Fig. 1, the point **x** is transformed to a new point $\mathbf{y} = \mathbf{e} = (1\ 1\ \ldots\ 1)^T$, which keeps the same distance from the orthant.

From Eq. (2.4), we have $\mathbf{x} = \mathbf{X}_k\,\mathbf{y}$. Hence the original LP problem

$$\text{Minimize } \mathbf{c}^T\mathbf{x} \qquad (2.5)$$

$$\text{Subject to } \mathbf{A}\mathbf{x} = \mathbf{b} \qquad (2.6)$$

$$(\mathbf{x} \geq \mathbf{0}) \qquad (2.7)$$

is transformed to

$$\text{Minimize } (\mathbf{c}^k)^T\,\mathbf{y} \qquad (2.8)$$

$$\text{Subject to } \mathbf{A}_k\,\mathbf{y} = \mathbf{b} \qquad (2.9)$$

$$(\mathbf{y} \geq \mathbf{0}) \tag{2.10}$$

where $\mathbf{c}^k = \mathbf{X}_k\mathbf{c}$ and $\mathbf{A}_k = \mathbf{A}\,\mathbf{X}_k$

Since $\mathbf{y}^k$ keeps the same distance from the orthant, it is considered "near the center" of the polytope. So we should move along the steepest descent direction $\mathbf{d}_y^k$ to find the new point $\mathbf{y}^{k+1} = \mathbf{y}^k + \alpha_k \mathbf{d}_y^k$, where $\alpha_k$ is the step length.

The steepest descent direction of the objective function is its negative gradient, $-\mathbf{c}^k$. In order to keep feasibility, this direction needs to be projected into the null space of the constraint matrix A. From the linear algebra, we have the null space projection matrix $\mathbf{P}_k = \mathbf{I} - \mathbf{A}_k^T(\mathbf{A}_k\,\mathbf{A}_k^T)^{-1}\mathbf{A}_k = \mathbf{I} - \mathbf{X}_k\mathbf{A}^T(\mathbf{A}\,\mathbf{X}_k^2\,\mathbf{A}^T)^{-1}\mathbf{A}\,\mathbf{X}_k$. The moving direction $\mathbf{d}_y^k = \mathbf{P}_k(-\mathbf{c}^k) = [\mathbf{I} - \mathbf{X}_k\mathbf{A}^T(\mathbf{A}\,\mathbf{X}_k^2\,\mathbf{A}^T)^{-1}\mathbf{A}\,\mathbf{X}_k](-\mathbf{X}_k\mathbf{c}) = -\mathbf{X}_k[\mathbf{c} - \mathbf{A}^T(\mathbf{A}\,\mathbf{X}_k^2\,\mathbf{A}^T)^{-1}\mathbf{A}\,\mathbf{X}_k^2\,\mathbf{c}]$.

If we denote $\mathbf{w}^k = (\mathbf{A}\,\mathbf{X}_k^2\,\mathbf{A}^T)^{-1}\mathbf{A}\,\mathbf{X}_k^2\,\mathbf{c}$,

$$\mathbf{d}_y^k = -\mathbf{X}_k[\mathbf{c} - \mathbf{A}^T\mathbf{w}^k] \tag{2.11}$$

Furthermore, if we denote $\mathbf{r}^k = \mathbf{c} - \mathbf{A}^T\mathbf{w}^k$,

$$\mathbf{d}_y^k = -\mathbf{X}_k\mathbf{r}^k \tag{2.12}$$

$$\mathbf{x}^{k+1} = \mathbf{X}_k\mathbf{y}^{k+1} = \mathbf{X}_k(\mathbf{y}^k + \alpha_k\mathbf{d}_y^k) = \mathbf{x}^k + \alpha_k\mathbf{X}_k\mathbf{d}_y^k \tag{2.13}$$

As for the step length $\alpha_k$, from $\mathbf{y}^{k+1} = \mathbf{y}^k + \alpha_k\mathbf{d}_y^k > 0$, we know that when $(d_y^k)_i < 0$, $\alpha_k$ should be smaller than

$$y_i^k / [-(d_y^k)_i] = 1/[-(d_y^k)_i] \tag{2.14}$$

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

$$\alpha_k = \min\{\alpha/[-(d_y^k)_i],\ \text{for } (d_y^k)_i < 0\} \tag{2.15}$$

to choose an appropriate step length in order to guarantee $\mathbf{y}^{k+1} > 0$.

The iterative procedure of the affine scaling algorithm can be easily derived based on the above discussion. Section 3.3.2 provides the step-by-step calculation for a numerical example problem solved with the affine scaling algorithm.

### 2.2.2 Finding the Starting Interior Point

An initial interior point has to be known beforehand in order to start the interior point method. There are a few methods to find such an initial interior point. The one introduced here is easier to implement [2]. It is also easier to understand due to its similarity to the Big-M method used in the simplex method.

The Big-M method used in the simplex method imposes a large positive number M as a penalty for each artificial variable and transforms the standard LP problem into the following LP problem:

Minimize $z = \mathbf{c}^T\mathbf{x} + Mx_a$

(2.16)

Subject to $\mathbf{Ax} + x_a = \mathbf{b}$ (2.17)

$(\mathbf{x}, x_a \geq 0)$ (2.18)

The starting point (solution) is $\mathbf{x} = 0$ and $x_a = \mathbf{b}$. When M is chosen large enough,

has feasible solution or unbounded solution.

Now we turn back to the interior point method. One artificial variable $x_a$ associated with a "big M" is added to the original problem and transforms it into the following problem:

Minimize $\mathbf{z} = \mathbf{cx} + Mx_a$ (2.19)

Subject to $[\ \mathbf{A} \mid (\mathbf{b}-\mathbf{Ae})\ ] \begin{bmatrix} x \\ x_a \end{bmatrix} = \mathbf{b}$ (2.20)

$(\mathbf{x}, x_a \geq 0)$ (2.21)

where $\mathbf{e} = (1 \quad 1 \quad \ldots \quad 1)^T \in R^n$. (2.22)

Comparing this problem with the big-M problem in the simplex method, we note these differences:

1. Only one artificial variable, $x_a$ (instead of $\mathbf{x}_a$), is added. In total, there are n + 1 variables, instead of n + m.
2. Although the objective function looks the same, the constraint matrix is different. The constraint matrix is manipulated so that $\mathbf{x} = (1 \quad 1 \quad \ldots \quad 1)^T \in R^{n+1}$ satisfies the transformed constraint matrix, which means $\mathbf{x} = (1 \quad 1 \quad \ldots \quad 1)^T$ is a solution to the transformed problem.

In fact, it is not only a solution, but also an interior solution. The reason is as follows:

From the basic theory of the simplex method, we know that graphically, the boundary of the feasible region is a hyperplane defined either by each constraint of $\mathbf{Ax} = \mathbf{b}$, in which all the slack variable and artificial variable equal to zero; or by the constraint $x_i = 0$. Either way, if a point $\mathbf{x}$ is on the boundary of a feasible region, there must be at least one zero in $\mathbf{x}$. Since the point $\mathbf{x} = (1 \quad 1 \quad \ldots \quad 1)^T$ is a solution to the transformed problem, it is either on the boundary of the feasible region or an interior point. And since there is no zero in $\mathbf{x} = (1 \quad 1 \quad \ldots \quad 1)^T$, it is not on the boundary. Hence, it is an interior

Similar to the big-M problem in the simplex method, the solution to the original problem can be derived from the solution to the above big-M problem:

1. If the artificial variable $x_a$ remains positive in the final solution of the big-M problem, the original problem is infeasible.
2. If the artificial variable $x_a$ is equal to zero in the final solution of the big-M problem, the original problem has the same optimal solution as the big-M problem.
3. If the big-M has unbounded solution, the original problem has unbounded solution, too.

# CHAPTER III

# DECOMPOSITION PRINCIPLE

Decomposition principle [1] is an algorithm for efficiently solving large-scale linear programming problems by breaking up the problem into smaller problems. This chapter introduces the theory of the decomposition principle.

## 3.1 Convex Set: Extreme Points, Extreme Directions and Theorems

A few theorems need to be discussed before the introduction to the decomposition principle. These theorems are essential to the derivation of the decomposition principle. And in order to make the explanation of these theorems easier, first we will review a few concepts of the convex theory that are used in these theorems. The first two concepts, convex sets and extreme points, are basic to the linear programming. They are briefly mentioned here in order to introduce a related, but much less well-known concept of extreme direction.

1. Convex sets

For k points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \in R^n$ and k scalars $\lambda_1, \lambda_2, \ldots, \lambda_k \in R$, we know that the expression $\lambda_1\mathbf{x}_1 + \lambda_2\mathbf{x}_2 + \ldots + \lambda_k\mathbf{x}_k$ is called a linear combination. It further becomes a convex combination when

$$\lambda_1 + \lambda_2 + \ldots + \lambda_k = 1 \text{ and } 0 \leq \lambda_1, \lambda_2, \ldots, \lambda_k \leq 1 \quad (3.1)$$

A set X is called a convex set if the convex combination of any two points in X is still in X.

Geometrically, for two points inside a polyhedron defined by a set, if the line segment joining them (which is the convex combination of these two points) is still inside the polyhedron, that set is a convex set.

2. Extreme points

A point in a convex set is called an extreme point if it cannot be represented by a convex combination of two distinct points in that set. In Figure 2, $\mathbf{y}_i$ (i = 1, 2, …, 5) are extreme points.

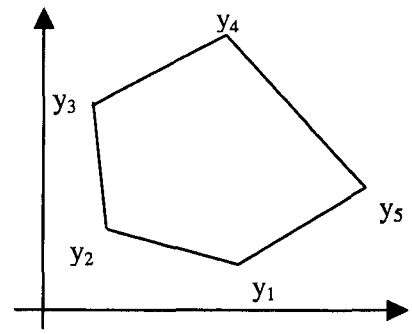



Figure 2: Extreme points of a bounded feasible region

3. Rays and directions

A ray is a set of points with the form

where d is a nonzero vector and is called the direction of the ray.

4. Extreme directions

Direction is nothing but a vector. First, we define the concept of the direction of the set. For a convex set X, a nonzero vector **d** is called a direction of the set if for each point $\mathbf{x} \in X$, the ray $\{ \mathbf{x} + \lambda\, \mathbf{d}: \lambda \geq 0 \} \in X$. It is obvious that for a bounded set as in the Figure 2, there are no directions of the set. From Figure 3, we can see that all the directions between $\mathbf{d}_1$ and $\mathbf{d}_2$ are directions of the set Y defined by the unbounded region, because they all satisfy $\mathbf{y} + \lambda\, \mathbf{d} \in Y$.

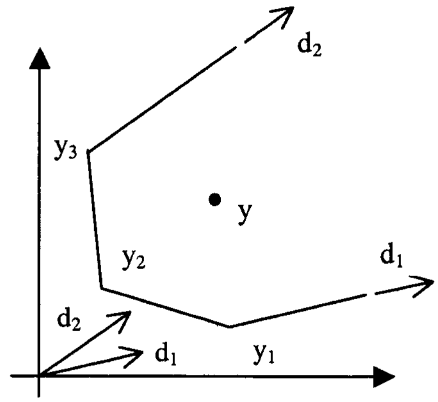



Figure 3: Extreme points and extreme directions of an unbounded feasible region

An extreme direction of a convex set is a direction of the set that cannot be represented as a positive linear combination of two distinct directions of the set. We can see that extreme directions to directions of the set is extreme points to points. In the Fig. 3, from linear algebra, we know that all the directions **d** between $\mathbf{d}_1$ and $\mathbf{d}_2$ are the positive linear combination of $\mathbf{d}_1$ and $\mathbf{d}_2$. However, although $\mathbf{d}_1$ ($\mathbf{d}_2$) can also be represented as linear combination of **d** and $\mathbf{d}_2$ ($\mathbf{d}_1$), the combination is not positive. Hence,

The discussion of LP decomposition algorithm can be greatly facilitated by referring to the following 3 theorems [7]:

<u>Theorem 1</u> (for the bounded region case)

Let $X = \{\mathbf{x}: \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ be a nonempty bounded set. Vector $\mathbf{x} \in X$ if and only if **x** can be represented as a convex combination of the extreme points ($\mathbf{y}_i$) of this set, that is,

$$\mathbf{x} = \sum_{j=1}^{k} \lambda_j \mathbf{y}_j \tag{3.3}$$

where $$\sum_{j=1}^{k} \lambda_j = 1 \tag{3.4}$$

$\lambda_j \geq 0$ ($j = 1, 2, ..., k$). k is the number of extreme points.

As an example, the interior point **x** in Figure 4 can be expressed as a linear combination of points $\mathbf{y}_4$ and **z** (see Figure 5).

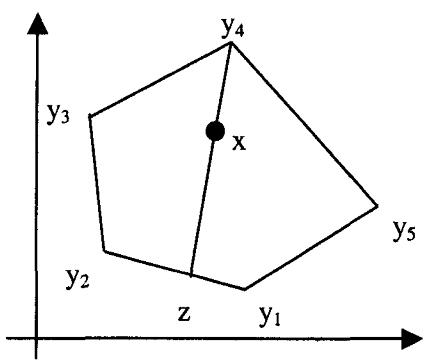



Fig. 4: Point **x** in a bounded region

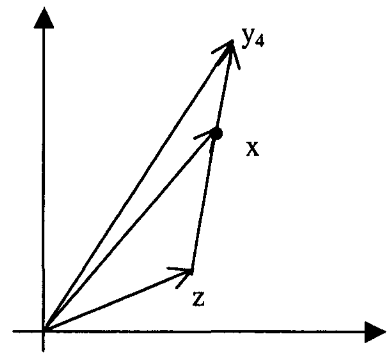



Fig. 5: Point **x** in convex combination

$$x = y_4 - \alpha(y_4 - z) = (1 - \alpha)y_4 + \alpha z \qquad (1 \geq \alpha \geq 0) \qquad (3.5)$$

Similarly, point z can be expressed as (see Figure 4):

$$z = (1 - \beta)y_2 + \beta y_1 \qquad (1 \geq \beta \geq 0) \qquad (3.6)$$

Hence $x = (1 - \alpha)y_4 + \alpha[(1 - \beta)y_2 + \beta y_1] = \alpha\beta y_1 + \alpha(1 - \beta)y_2 + (1 - \alpha)y_4$ (3.7)

Also, $\alpha\beta + \alpha(1 - \beta) + (1 - \alpha) = 1$ (3.9)

Notice that the representation is not unique (see Fig. 6).

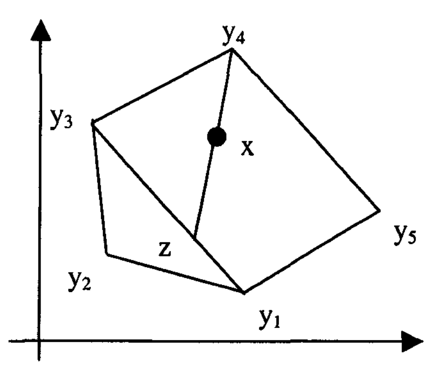



Fig. 6: Point **x** in another convex combination

<u>Theorem 2</u> (for the unbounded case)

Let X = {**x: Ax = b, x≥ 0**} be a nonempty set. Vector **x** ∈ X if and only if x can be represented as a convex combination of the extreme points ($\mathbf{y}_i$) plus a nonnegative linear combination of the extreme directions of this set ($\mathbf{d}_i$), that is,

$$\mathbf{x} = \sum_{j=1}^{k} \lambda_j \mathbf{y}_j + \sum_{j=1}^{l} \mu_j \mathbf{d}_j \tag{3.10}$$

where $\sum_{j=1}^{k} \lambda_j = 1$ (3.11)

$\lambda_j \geq 0$ (j = 1, 2, ..., k), $\mu_j \geq 0$ (j = 1, 2, ..., l), k is the number of extremes points and l is the number of extreme directions.

As an example, the interior point x in Fig. 7 can be expressed as (see Figure 8):

$$x = z + \mu d_1 \quad (\mu \geq 0) \tag{3.12}$$

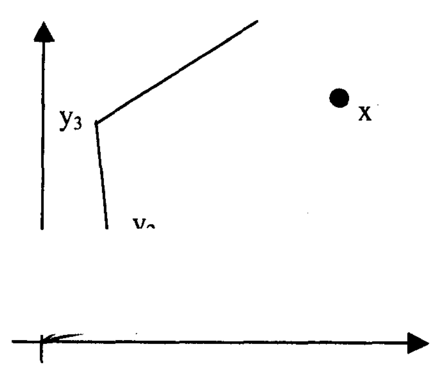



Fig. 7: Point **x** in an unbounded region

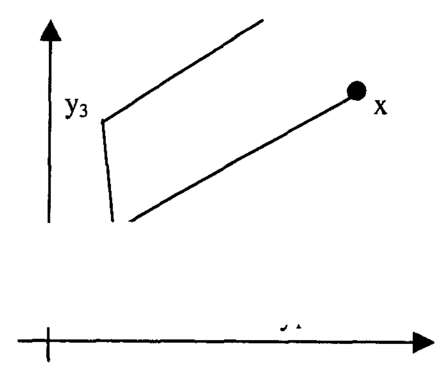



Fig. 8: Points **x** in linear combination

It should be noticed that the extreme direction $d_1$ (see Fig. 7) is parallel to the direction zx (see Fig. 8). Also, point z (in Fig. 8) can be expressed as:

$$z = y_2 + \beta(y_3 - y_2) = (1 - \beta)y_2 + \beta y_3 \qquad (1 \geq \beta \geq 0) \tag{3.13}$$

Hence we have

$$x = (1 - \beta)y_2 + \beta y_3 + \mu d_1 \tag{3.14}$$

Again, this representation is not unique. The point x can be also be represented in terms of $y_i$ and $d_2$ (see Fig. 9).

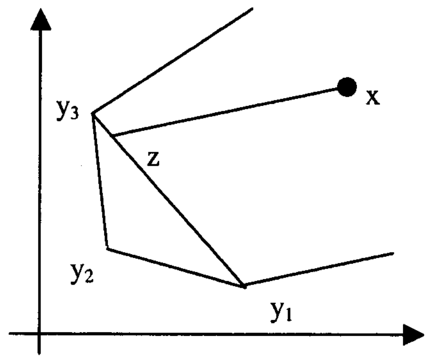



Fig. 9: Point **x** in another linear combination

Theorem 3

For the problem of

Maximize $\mathbf{c}^T\mathbf{x}$ (3.15)

subject to: $\mathbf{Ax} = \mathbf{b}$ (3.16)

$(\mathbf{x} \geq 0)$ (3.17)

(i). It has finite optimal solution if and only if all $\mathbf{cd}_i \leq 0$, where $\mathbf{d}_i$ is an extreme direction

(ii). If it has finite optimal solution, the solution is one of its extreme points.

Proof:

According to theorem 2, the foregoing problem can be transformed to

$$\text{Maximize } \sum_{j=1}^{k} (\mathbf{cy}_j)\lambda_j + \sum_{j=1}^{l} (\mathbf{cd}_j)\mu_j \quad (3.18)$$

$$\text{s.t.} \quad \sum_{j=1}^{k} \lambda_j = 1 \quad (3.19)$$

where $\lambda_j \geq 0$ (j = 1, 2, ..., k), $\mu_j \geq 0$ (j = 1, 2, ..., l) (3.20)

Now,

(1) If one of $\mathbf{cd}_j > 0$, since the corresponding $\mu_j$ can be arbitrarily large, the objective

function $\rightarrow \infty$. Hence there is no finite optimal solution.

(2) If all $\mathbf{cd}_j \leq 0$, in order to maximize the objective function, all $\mu_j$ can be made to be zero. Now the problem becomes

$$\text{Maximize } \sum_{j=1}^{k} (\mathbf{cy}_j)\lambda_j \tag{3.21}$$

$$\text{s.t.} \quad \sum_{j=1}^{k} \lambda_j = 1 \qquad \lambda_j \geq 0 \ (j = 1, 2, ..., k) \tag{3.22}$$

Let $\mathbf{cy}_g = \max \mathbf{cy}_j$ ($j$ = 1, 2, ..., k). Obviously, when $\lambda_g = 1$ and $\lambda_j = 0$ ($j \neq g$), the maximum value is found. Hence, the original problem has finite optimal solution, and the solution ($\mathbf{y}_g$) is one of its extreme points.

### 3.2 The Algorithm of the Decomposition principle

The general form of "block angular" linear programming (LP) problems considered in this work can be expressed as [4]

$$\text{Maximize } \mathbf{z} = \mathbf{c}_1\mathbf{x}_1 + \mathbf{c}_2\mathbf{x}_2 + \ldots + \mathbf{c}_p\mathbf{x}_p \tag{3.23}$$

Subject to

$$\mathbf{B}_1\mathbf{x}_1 = \mathbf{b}_1 \tag{3.25}$$

$$\mathbf{B}_2\mathbf{x}_2 = \mathbf{b}_2 \tag{3.26}$$

$$\ldots$$

$$\mathbf{B}_p\mathbf{x}_p = \mathbf{b}_p \tag{3.27}$$

$$(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p \geq 0) \tag{3.28}$$

where Eq. (3.24) is the common constraint, Eq. (3.25- 3.27) are the block (subproblem) constraints, p is the number of blocks, $\mathbf{x}_i$ and $\mathbf{c}_i$ is an $n_i$ dimensional vector, $\mathbf{b}$ is an m dimensional vector, $\mathbf{A}_i$ is an $m \times n_i$ matrix, $\mathbf{b}_i$ is a $r_i$ dimensional vector, $\mathbf{B}_i$ is a $r_i \times n_i$ matrix.

The following is an example problem in this format:

Maximize $z = - x_1 - 3x_2 - 5x_3 - 2x_4$ (3.29)

Subject to

$$5x_1 + 3x_2 + 4x_3 \leq 10 \quad (3.30)$$

$$x_1 + 2x_2 + 2x_3 + x_4 \leq 100 \quad (3.31)$$

$$5x_1 + x_2 \leq 9 \quad (3.32)$$

$$x_1 + 4x_2 \leq 8 \quad (3.33)$$

$$x_3 - 5x_4 \geq 4 \quad (3.34)$$

$$x_3 + x_4 \geq 10 \quad (3.35)$$

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

Comparing this problem with the "block angular form", we can see that it has two common constraints (Eq. 3.30 - 3.31) and two blocks (i.e., subproblems)

For less than (≤) and/or greater than (≥) type constraints, slack and/or surplus variables can be introduced to convert them into equality (=) type constraints, as indicated in Eq. 3.24 - 3.27. The feasible region (if exist), defined by Eq. 3.24-3.27, can be either bounded or unbounded.

Maximize $\sum_{j=1} (\mathbf{cy}_j)\lambda_j + \sum_{j=1} (\mathbf{cd}_j)\mu_j$ (3.36)

subject to

$$\sum_{j=1}^{k} \lambda_j = 1 \quad (3.37)$$

where $\lambda_j \geq 0$ (j = 1, 2, ..., k) and $\mu_j \geq 0$ (j = 1, 2, ..., l) (3.38)

With the above conclusion, and based upon the 3 theorems discussed before, the "original" LP problem (defined in Eq. 3.23 - 3.28) can be transformed into the following "new" LP problem:

Maximize

$$z = \sum_{j=1}^{k1} (c_1 y_{1j})\lambda_{1j} + \sum_{j=1}^{k2} (c_2 y_{2j})\lambda_{2j} + \dots + \sum_{j=1}^{kp} (c_p y_{pj})\lambda_{pj} + \sum_{j=1}^{l1} (c_1 d_{1j})\mu_{1j} + \sum_{j=1}^{l2} (c_2 d_{2j})\mu_{2j} + \dots + \sum_{j=1}^{lp} (c_p d_{pj})\mu_{pj} \quad (3.39)$$

subject to:

$$\sum_{j=1}^{k1} (A_1 y_{1j})\lambda_{1j} + \sum_{j=1}^{k2} (A_2 y_{2j})\lambda_{2j} + \dots + \sum_{j=1}^{kp} (A_p y_{pj})\lambda_{pj} + \sum_{j=1}^{l1} (A_1 d_{1j})\mu_{1j} + \sum_{j=1}^{l2} (A_2 d_{2j})\mu_{2j} + \dots + \sum_{j=1}^{lp} (A_p d_{pj})\mu_{pj} = b \quad (3.40)$$

$$\sum_{j=1}^{k1} \lambda_{1j} = 1 \quad (3.41)$$

$$\sum_{j=1}^{k2} \lambda_{2j} = 1 \quad (3.42)$$

$$\dots\dots$$

$$\sum_{j=1}^{lp} \lambda_{pj} = 1 \quad (3.43)$$

where $\lambda_{ij} \geq 0$ $(j = 1, 2, \dots, ki)$, $\mu_{ij} \geq 0$ $(j = 1, 2, \dots, l_i)$ (3.44)

It is important to recognize that each block constraints in the "original" LP problem

variables x has been transformed into the new variables $\lambda_{ij}$ and $\mu_{ij}$.

The revised Simplex (product form) algorithm can be applied in the LP problem of Eq. 3.39 – 3.44, with "minor detailed" changes in the steps to select the Entering (and Leaving) variables into (and from) the basic variable group.

**(i) How to choose the entering variable?**

The entering variable in the revised Simplex method corresponds to the global maximum of $c_{ij} - z_{ij}$ (or minimum of $z_{ij} - c_{ij}$, which is the notation format used in chapter 4). Instead of finding the global maximum value, we can find the local maximum value first (corresponding to each block), then choose the maximum amongst these values.

**(ii) How to find the local max. $c_{ij} - z_{ij}$?**

According to the revised Simplex method, $c_{ij} - z_{ij} = c_{ij} - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{p}_{ij}$ (3.45)

Let $\mathbf{c}_B\mathbf{B}^{-1} = (\omega_1, \omega_2, \ldots, \omega_m, \alpha_1, \alpha_2, \ldots, \alpha_p) = (\boldsymbol{\omega}, \boldsymbol{\alpha})$ (3.46)

(a) Corresponding to $\lambda_{ij}$, one has $\mathbf{p}_{ij} = \begin{bmatrix} \mathbf{A}_i\mathbf{y}_{ij} \\ \mathbf{e}_i \end{bmatrix}$ (3.47)

where $\mathbf{e}_i$ is a p dimensional vector with the i-th entry equal to 1 and all the other entries equal to 0.

Hence $c_{ij} - z_{ij} = c_{ij} - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{p}_{ij} = \mathbf{c}_i\mathbf{y}_{ij} - (\boldsymbol{\omega}, \boldsymbol{\alpha}) * \begin{bmatrix} \mathbf{A}_i\mathbf{y}_{ij} \\ \mathbf{e}_i \end{bmatrix} = (\mathbf{c}_i - \boldsymbol{\omega}\,\mathbf{A}_i)\mathbf{y}_{ij} - \alpha_i.$ (3.48)

(b) Corresponding to $\mu_{ij}$, one has $\mathbf{p}_{ij} = \begin{bmatrix} \mathbf{A}_i\mathbf{d}_{ij} \\ \mathbf{0} \end{bmatrix}$ (3.49)

Hence $c_{ij} - z_{ij} = c_{ij} - \mathbf{c}_B\mathbf{B}^{-1}\mathbf{p}_{ij} = \mathbf{c}_i\mathbf{d}_{ij} - (\boldsymbol{\omega}, \boldsymbol{\alpha})\begin{bmatrix} \mathbf{A}_i\mathbf{d}_{ij} \\ \mathbf{0} \end{bmatrix} = (\mathbf{c}_i - \boldsymbol{\omega}\,\mathbf{A}_i)\mathbf{d}_{ij}$ (3.50)

For each block, instead of solving Max $[\mathbf{c}_i - \boldsymbol{\omega}\,\mathbf{A}_i]\mathbf{y}_{ij} - \alpha_i]$ and Max $[(\mathbf{c}_i - \boldsymbol{\omega}\,\mathbf{A}_i)\mathbf{d}_{ij}]$ to decide which $\lambda_{ij}$ or $\mu_{ij}$ becomes a candidate of the entering variable, we can just solve the problem Max $(\mathbf{c}_i - \boldsymbol{\omega}\,\mathbf{A}_i)\mathbf{x}_j$, subject to $\mathbf{B}_1\mathbf{x}_1 = \mathbf{b}_1$. The reasons are given below:

= 0. Hence none of $\mu_{ij}$ can be a candidate of the entering variable. Also according to Theorem 3, the optimal solution is one of its extreme points, say, $\mathbf{y}_{ij}$. Obviously, this $\mathbf{y}_{ij}$ also maximize $(\mathbf{c}_i - \boldsymbol{\omega}\,\mathbf{A}_i)\mathbf{y}_{ij} - \alpha_i$. Hence $\lambda_{ij}$, the corresponding variable, becomes a candidate of the entering variable.

(2) If this problem has unbounded solution, according to Theorem 3, there is at least one $\mathbf{d}_{ij}$ which makes $(\mathbf{c}_i - \boldsymbol{\omega}\,\mathbf{A}_i)\mathbf{d}_{ij} > 0$. Notice that this $\mathbf{d}_{ij}$ can be very large, so $(\mathbf{c}_i - \boldsymbol{\omega}\,\mathbf{A}_i)\mathbf{d}_{ij} \to \infty$. Hence $\mu_{ij}$, the corresponding variable, becomes the entering variable.

(3) If this problem has no solution, then this block has no feasible region. Hence, the original problem has no feasible region, meaning that there is no solution to the original problem.

Once the entering variable is found, one has no problems in locating the leaving variable. Hence, the standard revised Simplex procedure can be normally applied afterward.

### 3.3 Finding the Optimal Direction

It should be noted that among multiple extreme directions of an unbounded region, only one makes the value of the objective function increase (or decrease) the fastest. It is similar to the case that in a bounded region, only one of the extreme points makes the objective optimum. Such an extreme point is called the optimal point. Similarly, we call such an extreme direction the optimal direction.

From the discussion in the last section, we know that if a subproblem has unbounded solution, a variable corresponding to the optimal direction will become the entering variable. Also, the value of the optimal direction has to be known for the succeeding calculations. Two examples are given below to show how to find the optimal direction in both the simplex method and the interior point method.

In the following example, the calculation in each simplex iteration is shown in the simplex tableaus.

<u>Example 3.1</u>

Maximize $z = x_1 + 2x_2$

Subject to $-x_1 + x_2 \leq 2$

$-x_1 + 2x_2 \leq 8$

$(x_1, x_2 \geq 0)$

<u>Solution</u>

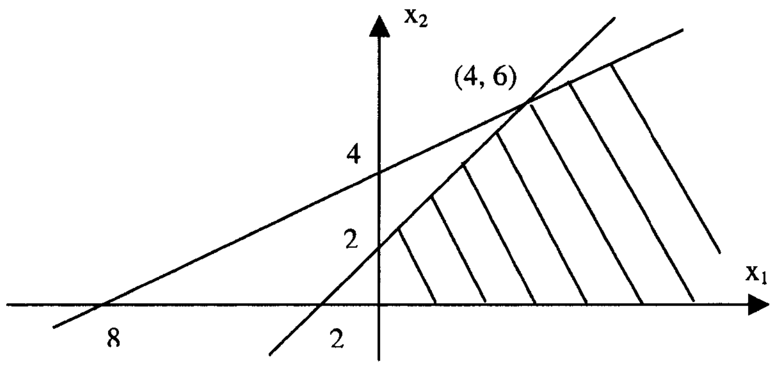



Figure 10: Feasible region of Example 3.1

Iteration 1

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | b |
|---|---|---|---|---|---|
| $x_3$ | -1 | 1 | 1 | 0 | 2 |
| $x_4$ | -2 | 2 | 0 | 1 | 8 |
| | 1 | 2 | 0 | 0 | z |

Iteration 2

| | | | | | |
|---|---|---|---|---|---|
| $x_4$ | 1 | 0 | -2 | 1 | 4 |
| | 3 | 0 | -2 | 0 | Z - 4 |

Iteration 3

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | b |
|---|---|---|---|---|---|
| $x_2$ | 0 | 1 | -1 | 1 | 6 |
| $x_1$ | 1 | 0 | -2 | 1 | 4 |
| | 0 | 0 | 4 | -3 | Z - 16 |

Since the most positive value in the last row is 4, $x_3$ becomes the entering variable. And since all of $x_3$'s coefficients are negative, this problem has unbounded solution. The iteration stops here, but let us keep going a little further to see what will happen after $x_3$ becomes the entering variable. Now that $x_3$ becomes the entering variable, its value will increase from 0. From the last tableau, we can see that $x_1$ and $x_2$ will be increased to

$$x_2 = 4 - (-2)\, x_3 = 4 + 2x_3$$

$$x_2 = 6 - (-1)\, x_3 = 6 + x_3$$

And $x_4$ stays where it is:

$$x_4 = 0$$

The preceding solution can be arranged as

$$\mathbf{x} = \begin{bmatrix} 4 \\ 6 \\ 0 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad (x_3 > 0)$$

Now we can see that actually the solution is a ray.

When $x_3 \rightarrow \infty$, the solution moves along this ray and the objective function

$$z = x_1 + 2x_2 = 4 + 2x_3 + 2(6 + x_3) = 16 + 4x_3$$ [illegible]

problem.

### 3.3.2 Finding the Optimal Direction in the Interior Point Method

In the interior point method, finding the optimal direction is much easier than in the simplex method. From Section 2.2.1, we know that in each iteration of the interior point method, the moving direction $\mathbf{d}_y$ is calculated. Since it is the steepest decent direction, it will become the optimal direction at the last iteration. However, the direction $\mathbf{d}_y$ is in the "**Y**" space. It needs to be projected back to the original "**X**" space. Let the optimal direction in the "**X**" space be denoted as $\mathbf{d}_x$, since $\mathbf{Y} = \mathbf{X}_k^{-1}\, \mathbf{X}$ (see Section 2.2.1), we have

$\mathbf{X} = \mathbf{X}_k\, \mathbf{Y}$. Hence $\mathbf{d}_x = \mathbf{X}_k\, \mathbf{d}_y$.

The example problem 3.2 is the same as example 3.1. Now we solve it with the interior point method. The solved optimal direction can be verified with the result of the simplex method.

Example 3.2

Minimize $z = -x_1 - 2x_2$

Subject to $-x_1 + x_2 \le 2$

$-x_1 + 2x_2 \le 8$

$(x_1, x_2 \ge 0)$

Solution

$$\mathbf{x} = [x_1 \quad x_2 \quad x_3 \quad x_4]^T,\ \mathbf{b} = [2 \quad 8]^T,\ \mathbf{c} = [-1 \quad -2 \quad 0 \quad 0]^T,\ \mathbf{A} = \begin{bmatrix} -1 & 1 & 1 & 0 \\ -1 & 2 & 0 & 1 \end{bmatrix}$$

**Iteration 1**

The starting point can be any point inside the feasible region. By observing Figure 10,

$$\text{Hence } \mathbf{X_0} = \begin{bmatrix} 0 & 4 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

From $\mathbf{A\ X_0^2 A^T\ w_0 = A\ X_0^2\ c}$, we have $\begin{bmatrix} 36 & 48 \\ 48 & 96 \end{bmatrix} \mathbf{w_0} = \begin{bmatrix} -16 \\ -48 \end{bmatrix}$

$$\text{Hence } \mathbf{w_0} = \begin{bmatrix} 0.6667 \\ 0.8333 \end{bmatrix}$$

$$\text{Now } \mathbf{r_0} = \mathbf{c} - \mathbf{A}^T \mathbf{w_0} = \begin{bmatrix} -1.167 \\ -1 \\ -0.6667 \\ 0.8333 \end{bmatrix}$$

Since some components of $\mathbf{r}_0$ are < 0, continue the iteration.

$$\mathbf{d}_{y0} = -\mathbf{X}_0\,\mathbf{r}_0 = \begin{bmatrix} 4.667 \\ 4 \\ 1.333 \\ -3.333 \end{bmatrix}$$

Since some components of $\mathbf{d}_{y0}$ are < 0, continue the iteration.

Take $\alpha = 0.99$, then the step length $\alpha_0 = \min[\,\alpha /- (\mathbf{d}_{y0})_i] = 0.99/3.333 = 0.2970$

$$\text{Hence } \mathbf{x}_1 = \mathbf{x}_0 + \alpha_0\,\mathbf{X}_0\,\mathbf{d}_{y0} = \begin{bmatrix} 9.544 \\ 8.752 \\ 2.792 \\ 0.04 \end{bmatrix}$$

**Iteration 2**

From $\mathbf{A}\,\mathbf{X}_1^2\mathbf{A}^T\,\mathbf{w}_1 = \mathbf{A}\,\mathbf{X}_1^2\,\mathbf{c}$, we have $\begin{bmatrix} 175.5 & 244.3 \\ 244.3 & 397.5 \end{bmatrix}\mathbf{w}_1 = \begin{bmatrix} 62.11 \\ 215.3 \end{bmatrix}$

$$\text{Hence } \mathbf{w}_1 = \begin{bmatrix} 277.0 \\ -224.4 \end{bmatrix}$$

2.244

Since some components of $\mathbf{r}_1$ are < 0, continue the iteration.

$$\mathbf{d}_{y1} = -\mathbf{X}_1\,\mathbf{r}_1 = \begin{bmatrix} 4.525 \\ 2.467 \\ 7.733 \\ -0.08975 \end{bmatrix}$$

Since some components of $\mathbf{d}_{y1}$ are < 0, continue the iteration.

The step length $\alpha_1 = 0.99/0.08975 = 11.03$

Hence $\mathbf{x_2} = \mathbf{x_1} + \alpha_1 \mathbf{X_1} \mathbf{d_{y1}} = \begin{bmatrix} 485.9 \\ 247.0 \\ 241.0 \\ 0 \end{bmatrix}$

**Iteration 3**

From $\mathbf{A}\ \mathbf{X_2}^2\mathbf{A}^T\ \mathbf{w_2} = \mathbf{A}\ \mathbf{X_2}^2\ \mathbf{c}$, we have

$$\begin{bmatrix} 3.552 & 3.581 \\ 3.581 & 4.801 \end{bmatrix} 10^5\ \mathbf{w_1} = \begin{bmatrix} 1.141 \\ -0.07838 \end{bmatrix} 10^5$$

Hence $\mathbf{w_2} = \begin{bmatrix} 1.363 \\ -1.033 \end{bmatrix}$

Now $\mathbf{r_2} = \mathbf{c} - \mathbf{A}^T\ \mathbf{w_2} = \begin{bmatrix} -0.6701 \\ -1.297 \\ -1.363 \\ 1.033 \end{bmatrix}$

Since some components of $\mathbf{r_2}$ are < 0, continue the iteration.

$\begin{bmatrix} 325.6 \\ 320.4 \end{bmatrix}$

Since $\mathbf{d_{y2}}$ is > 0, this problem is unbounded.

$$\mathbf{d_{x2}} = \mathbf{X_2}\ \mathbf{d_{y2}} = \begin{bmatrix} 485.9 & 0 & 0 & 0 \\ 0 & 247.0 & 0 & 0 \\ 0 & 0 & 241.0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 325.6 \\ 320.4 \\ 328.3 \\ 0 \end{bmatrix} = 7.911 \times 10^5 \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Hence, the extreme direction of this problem is $[2 \quad 1 \quad 1 \quad 0]^T$. It is the same as the result of the simplex method, shown in Example 3.1.

# CHAPTER IV

# NUMERICAL STUDIES OF THE DECOMPOSITION PRINCIPLE

As we can see from the last chapter, the theory of the decomposition principle is not that straightforward to understand. This chapter provides numerical examples to illustrate the decomposition principle algorithm step by step. These examples include both cases of LP problems: problems with bounded feasible region and with unbounded feasible region. The step-by-step calculation not only serves the purpose of illustrating the decomposition principle, but also is used to check the result of the parallel algorithm. Debugging the code of a parallel algorithm could be a nightmare for a programmer because the compiler gives very little error message if something is wrong in the code [8]. To make it worse, very often the error message is irrelevant to the actual error. It is essential to compare the computation result of the code with the result of hand calculation step by step to make sure the computers (or more precisely, the processors) are doing what they are supposed to do. In this study, a code of the parallel algorithm of the decomposition principle is

LP problems to test its performance, which is reported at the end of this chapter.

## 4.1 Sequential Algorithm of the Decomposition principle

This section presents the small size numerical examples under the sequential computation environment. In order to make the hand calculation easier, the feasible region of each subproblem is drawn in figure so that the optimal solution to each subproblem can be obtained just by observing the figure instead of by calculation.

### 4.1.1 Bounded Feasible Region Case

Here the "bounded feasible region case" means each subproblem of the original problem has bounded feasible region. Two examples in bounded feasible region case are presented in this section. The second one deserves more attention. We know that if a LP problem has no solution, it has no feasible region. It should be noted here that even though each subproblem of a LP problem has feasible region, the original problem may not have solution at all, as shown by the second example.

#### 4.1.1.1 Example Problem With Optimal Solution

Problem

Maximize $z = x_1 + 3x_2 + 5x_3 + 2x_4$

Subject to

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

$$5x_1 + x_2 \le 9$$

$$x_1 + 4x_2 \le 8$$

$$x_3 - 5x_4 \le 4$$

[illegible] $\le 10$

Solution

The original problem can be transformed to:

$$\text{Maximize } z = \sum_{j=1}^{k1} (\mathbf{c}_1\mathbf{y}_{1j})\lambda_{1j} + \sum_{j=1}^{k2} (\mathbf{c}_2\mathbf{y}_{2j})\lambda_{2j}$$

$$\text{s.t. } \sum_{j=1}^{k1} (\mathbf{A}_1\mathbf{y}_{1j})\lambda_{1j} + \sum_{j=1}^{k2} (\mathbf{A}_2\mathbf{y}_{2j})\lambda_{2j} = 10$$

$$\sum_{j=1}^{k1} \lambda_{1j} = 1$$

$$\sum_{j=1}^{k2} \lambda_{2j} = 1$$

where $\lambda_{ij} \geq 0$ ($j = 1, 2, \ldots, k_i$).

It has the Simplex tableau as follows (using the big-M method):

| | $\lambda_{11}$ $\lambda_{12}$ $\lambda_{13}$ ...... | $\lambda_{21}$ $\lambda_{22}$ $\lambda_{23}$ ...... | $x_5$ | $x_6$ | $x_{11}$ | $x_{12}$ | b |
|---|---|---|---|---|---|---|---|
| Z | $\mathbf{c}_1\mathbf{y}_{11}$ $\mathbf{c}_1\mathbf{y}_{12}$ $\mathbf{c}_1\mathbf{y}_{13}$ ...... | $\mathbf{c}_2\mathbf{y}_{21}$ $\mathbf{c}_2\mathbf{y}_{22}$ $\mathbf{c}_2\mathbf{y}_{23}$ ...... | 0 | -M | -M | -M | |
| $X_6$ | $\mathbf{A}_1\mathbf{y}_{11}$ $\mathbf{A}_1\mathbf{y}_{12}$ $\mathbf{A}_1\mathbf{y}_{13}$ ...... | $\mathbf{A}_2\mathbf{y}_{21}$ $\mathbf{A}_2\mathbf{y}_{22}$ $\mathbf{A}_2\mathbf{y}_{23}$ ...... | -1 | 1 | 0 | 0 | 10 |
| $x_{11}$ | 1 1 1 ...... | 0 0 0 ...... | 0 | 0 | 1 | 0 | 1 |
| $x_{12}$ | 0 0 0 ...... | 1 1 1 ...... | 0 | 0 | 0 | 1 | 1 |

<u>Subproblem 1</u>

$\mathbf{x}_1 = (x_1, x_2)^T$, $\mathbf{c}_1 = (1, 3)^T$, $\mathbf{A}_1 = (5, 3)$

Its constraints are:

$$5x_1 + x_2 \leq 9$$

$$x_1 + 4x_2 \leq 8$$

These two constraints define the following feasible region:

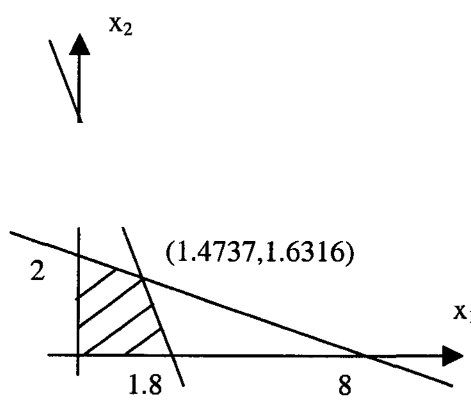



Fig. 11. Feasible region of subproblem 1 of 4.1.1.1

<u>Subproblem 2</u>

$\mathbf{x}_2 = (x_3, x_4)^T$, $\mathbf{c}_2 = (5, 2)^T$, $\mathbf{A}_2 = (4, 0)$

The constraints of this subproblem are:

$$x_3 - 5x_4 \leq 4$$

$$x_3 + x_4 \leq 10$$

These two constraints define the following feasible region:

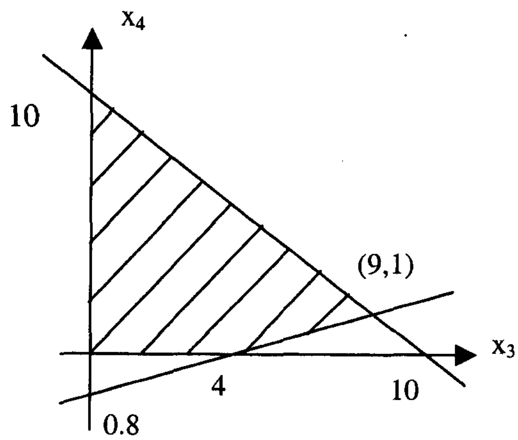



Fig. 12. Feasible region of subproblem 2 of 4.1.1.1

As we mentioned before, the optimal solution to each subproblem will be observed directly from the figure of its feasible region.

**Iteration 0**

$\mathbf{x}_B = (x_6, x_{11}, x_{12})^T = (10, 1, 1)^T$

$\mathbf{c}_B = (- M, - M, - M), \mathbf{B} = \mathbf{B}^{-1} = \mathbf{I}$

**Iteration 1**

$\mathbf{c}_B\mathbf{B}^{-1} = \mathbf{c}_B = (- M, - M, - M)$

<u>Subproblem 1</u>

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (- M, - M, - M)\begin{bmatrix} 5x_1 + 3x_2 \\ 1 \\ 0 \end{bmatrix} - (x_1 + 3x_2)$$

$$= (-5M - 1)x_1 - 3(M + 1)x_2 - M$$

Subject to $\quad x_1 + \ \ x_2 \leq 9$

$\qquad\qquad\quad x_1 + 4x_2 \leq 8$

By observing the figure of its feasible region, we can see that the solution is $\mathbf{y}_{11} = (x_1, x_2)^T = (1.4737, 1.6316)^T$, Min $(z_1 - c_1) = - 13.263M - 6.3685$

Subproblem 2

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_2\mathbf{y}_2 \\ 0 \\ 1 \end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (-M, -M, -M)\begin{bmatrix} 4x_3 \\ 0 \\ 1 \end{bmatrix} - (5x_3 + 2x_4)$$

$$= (-5 - 4M)x_3 - 2x_4 - M$$

Subject to $x_3 - 5x_4 \le 4$

$x_3 + x_4 \le 10$

By observing the figure of its feasible region, we can see that the solution is $\mathbf{y}_{21} = (x_3, x_4)^T = (9, 1)^T$, Min $(z_2 - c_2) = -37M - 47$

$$\text{Also, } z_5 - c_5 = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} - 0 = M$$

The global min. is min. $(z_2 - c_2) = -37M - 47$.

Hence $\lambda_{21}$ becomes the entering variable.

$$\mathbf{P}_{21} = \begin{bmatrix} \mathbf{A}_2\mathbf{y}_{21} \\ 0 \end{bmatrix} = \begin{bmatrix} (4,0)\begin{pmatrix} 9 \\ 1 \end{pmatrix} \\ 0 \end{bmatrix} = \begin{bmatrix} 36 \\ 0 \end{bmatrix}$$

$$\text{Thus } \mathbf{B}^{-1}\mathbf{P}_{21} = \mathbf{I}^*\mathbf{P}_{21} = \begin{bmatrix} 36 \\ 0 \\ 1 \end{bmatrix}$$

Given $\mathbf{x}_B = (x_6, x_{11}, x_{12})^T = (10, 1, 1)^T$, so $q = 1$.

Hence $x_6$ becomes the leaving variable.

$$\text{So } \mathbf{B}^{-1} = \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} * \mathbf{I} = \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix}$$

The new basic solution

$$\mathbf{x}_B = (\lambda_{21}, x_{11}, x_{12})^T = \mathbf{B}^{-1}(10, 1, 1)^T = \begin{bmatrix} \frac{5}{18} \\ 1 \\ \frac{13}{18} \end{bmatrix}$$

$$\mathbf{c}_B = (\mathbf{c}_2\mathbf{y}_{21}, -M, -M) = (47, -M, -M)$$

**Iteration 2**

$$\mathbf{c}_B\mathbf{B}^{-1} = (47, -M, -M)\begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} = \left(\frac{47+M}{36}, -M, -M\right)$$

Subproblem 1

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = \left(\frac{47+M}{36}, -M, -M\right)\begin{bmatrix} 5x_1 + 3x_2 \\ 1 \\ 0 \end{bmatrix} - (x_1 + 3x_2)$$

$$= \frac{47+M}{36}(5x_1 + 3x_2) - (x_1 + 3x_2) - M$$

Subject to $\quad x_1 + x_2 \leq 9$

$\quad x_1 + 4x_2 \leq 8$

The solution is $\mathbf{y}_{12} = (x_1, x_2)^T = (0, 0)^T$, Min $(z_1 - c_1) = -M$

Subproblem 2

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_2\mathbf{y}_2 \\ 0 \\ 1 \end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = \left(\frac{47+M}{36}, -M, -M\right)\begin{bmatrix} 4x_3 \\ 0 \\ 1 \end{bmatrix} - (5x_3 + 2x_4)$$

$$= \frac{47+M}{9}x_3 - (5x_3 + 2x_4) - M$$

Subject to $\quad x_3 - 5x_4 \leq 4$

$\quad x_3 + x_4 \leq 10$

The solution is $\mathbf{y}_{22} = (x_3, x_4)^T = (0, 10)^T$, Min $(z_2 - c_2) = -M - 20$

$$\text{Also, } z_5 - c_5 = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} - 0 = \left(\frac{47+M}{36}, -M, -M\right)\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} - 0 = -\frac{47+M}{36}$$

The global min. is min. $(z_2 - c_2) = -M - 20$.

Hence $\lambda_{22}$ becomes the entering variable.

$$\mathbf{P}_{22} = \begin{bmatrix} \mathbf{A}_2\mathbf{y}_{22} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} (4,0)\begin{pmatrix} 0 \\ 10 \end{pmatrix} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{Thus } \mathbf{B}^{-1}\mathbf{P}_{22} = \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{Since } \mathbf{x}_B = (\lambda_{21}, x_{11}, x_{12})^T = \begin{bmatrix} \frac{5}{18} \\ 1 \\ \frac{13}{18} \end{bmatrix}, q = 3.$$

Hence $x_{12}$ becomes the leaving variable.

$$\text{So } \mathbf{B}^{-1} = \mathbf{I} * \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} = \mathbf{B}^{-1} \text{ of last iteration}$$

The new basic solution

$\frac{13}{18}$

$$\mathbf{c}_B = (47, -M, \mathbf{c}_2\mathbf{y}_{22}) = (47, -M, (5, 2)\begin{pmatrix} 0 \\ 10 \end{pmatrix}) = (47, -M, 20)$$

**Iteration 3**

$$\mathbf{c}_B\mathbf{B}^{-1} = (47, -M, 20) \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} = (\tfrac{3}{4}, -M, 20)$$

Subproblem 1

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (\tfrac{3}{4}, -M, 20)\begin{bmatrix} 5x_1 + 3x_2 \\ 1 \\ 0 \end{bmatrix} - (x_1 + 3x_2) = \tfrac{11}{4}x_1 - \tfrac{3}{4}x_2 - M$$

Subject to $x_1 + x_2 \leq 9$

$x_1 + 4x_2 \leq 8$

The solution is $y_{13} = (x_1, x_2)^T = (0, 2)^T$, $\text{Min } (z_1 - c_1) = -M - 1.5$

<u>Subproblem 2</u>

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_2\mathbf{y}_2 \\ 0 \\ 1 \end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (\tfrac{3}{4}, -M, 20)\begin{bmatrix} 4x_3 \\ 0 \\ 1 \end{bmatrix} - (5x_3 + 2x_4)$$

$$= 3x_3 + 20 - (5x_3 + 2x_4) = -2x_3 - 2x_4 + 20$$

Subject to $x_3 - 5x_4 \leq 4$

$x_3 + x_4 \leq 10$

The solution is $\mathbf{y}_{23} = (x_3, x_4)^T = (0, 10)^T$, $\text{Min } (z_2 - c_2) = 0$

$$\text{Also, } z_5 - c_5 = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} - 0 = (\tfrac{3}{4}, -M, 20)\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} = -\tfrac{3}{4}$$

Hence $\lambda_{13}$ becomes the entering variable.

$$\mathbf{P}_{13} = \begin{bmatrix} \mathbf{A}_1\mathbf{y}_{13} \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} (5,3)\begin{pmatrix} 0 \\ 2 \end{pmatrix} \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 0 \end{bmatrix}$$

$$\text{Thus } \mathbf{B}^{-1}\mathbf{P}_{13} = \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix}\begin{bmatrix} 6 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} \\ 1 \\ -\frac{1}{6} \end{bmatrix}$$

Since $\mathbf{x}_B = (\lambda_{21}, x_{11}, \lambda_{22})^T = \begin{bmatrix} \frac{5}{18} \\ 1 \\ \frac{13}{18} \end{bmatrix}$, q = 2.

Hence $x_{11}$ becomes the leaving variable.

$$\text{So } \mathbf{B}^{-1} = \begin{bmatrix} 1 & -\frac{1}{6} & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{6} & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{36} & -\frac{1}{6} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & \frac{1}{6} & 1 \end{bmatrix}$$

The new basic solution

$$\mathbf{x}_B = (\lambda_{21}, \lambda_{13}, \lambda_{22})^T = \mathbf{B}^{-1}(10, 1, 1)^T = \begin{bmatrix} \frac{1}{9} \\ 1 \\ \frac{8}{9} \end{bmatrix}$$

$$\mathbf{c}_B = (47, \mathbf{c}_1\mathbf{y}_{13}, 20) = (47, (1, 3)\begin{pmatrix} 0 \\ 2 \end{pmatrix}, 20) = (47, 6, 20)$$

**Iteration 4**

$$\mathbf{c}_B\mathbf{B}^{-1} = (47, 6, 20)\begin{bmatrix} \frac{1}{36} & -\frac{1}{6} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & \frac{1}{6} & 1 \end{bmatrix} = (0.75, 1.5, 20)$$

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (0.75, 1.5, 20)\begin{bmatrix} 5x_1 + 3x_2 \\ 1 \\ 0 \end{bmatrix} - (x_1 + 3x_2)$$

$$= 2.75x_1 - 0.75x_2 + 1.5$$

Subject to $x_1 + x_2 \leq 9$

$x_1 + 4x_2 \leq 8$

The solution is $\mathbf{y}_{14} = (x_1, x_2)^T = (0, 2)^T$, Min $(z_1 - c_1) = 0$

<u>Subproblem 2</u>

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_2\mathbf{y}_2 \\ 0 \\ 1 \end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (0.75, 1.5, 20)\begin{bmatrix} 4x_3 \\ 0 \\ 1 \end{bmatrix} - (5x_3 + 2x_4)$$

$$= 3x_3 + 20 - (5x_3 + 2x_4) = -2x_3 - 2x_4 + 20$$

Subject to $x_3 - 5x_4 \le 4$

$x_3 + x_4 \le 10$

The solution is $\mathbf{y}_{24} = (x_3, x_4)^T = (0, 10)^T$, Min $(z_2 - c_2) = 0$

$$\text{Also, } z_5 - c_5 = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} - 0 = (0.75, 1.5, 20)\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} = -0.75$$

The global min. is $z_5 - c_5 = -0.75$

<u>Hence $x_5$ becomes the entering variable.</u>

$$\mathbf{P}_{54} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

[illegible]

$$\text{Since } \mathbf{x}_B = (\lambda_{21}, \lambda_{13}, \lambda_{22})^T = \begin{bmatrix} \frac{1}{9} \\ 1 \\ \frac{8}{9} \end{bmatrix}, q = 3.$$

<u>Hence $\lambda_{22}$ becomes the leaving variable.</u>

$$\text{So } B^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 36 \end{bmatrix}\begin{bmatrix} \frac{1}{36} & -\frac{1}{6} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & \frac{1}{6} & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 6 & 36 \end{bmatrix}$$

The new basic solution

$$\mathbf{x}_B = (\lambda_{21}, \lambda_{13}, x_5)^T = \mathbf{B}^{-1}(10, 1, 1)^T = \begin{bmatrix} 1 \\ 1 \\ 32 \end{bmatrix}$$

$\mathbf{c}_B = (47, 6, 0)$

**Iteration 5**

$$\mathbf{c}_B\mathbf{B}^{-1} = (47, 6, 0)\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 6 & 36 \end{bmatrix} = (0, 6, 47)$$

Subproblem 1

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (0, 6, 47)\begin{bmatrix} 5x_1 + 3x_2 \\ 1 \\ 0 \end{bmatrix} - (x_1 + 3x_2) = -x_1 - 3x_2 + 6$$

Subject to $x_1 + x_2 \le 9$

$x_1 + 4x_2 \le 8$

The solution is $\mathbf{y}_{15} = (x_1, x_2)^T = (0, 2)^T$, Min $(z_1 - c_1) = -6.3685 + 6 = -0.3685$

Subproblem 2

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \text{[illegible]} \\ 0 \\ 1 \end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (0, 6, 47)\begin{bmatrix} \text{[illegible]} \\ 0 \\ 1 \end{bmatrix} - (5x_3 + 2x_4) = -5x_3 - 2x_4 + 47$$

Subject to $x_3 - 5x_4 \le 4$

$x_3 + x_4 \le 10$

The solution is $\mathbf{y}_{25} = (x_3, x_4)^T = (9, 1)^T$, Min $(z_2 - c_2) = 0$

The global min. is min $(z_1 - c_1) = -0.3685$

Hence $\lambda_{15}$ becomes the entering variable.

$$\mathbf{P}_{15} = \begin{bmatrix} \mathbf{A}_1\mathbf{y}_{15} \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} (5,3)\begin{pmatrix} 1.4737 \\ 1.6316 \end{pmatrix} \\ 1 \\ 0 \\ \\ \end{bmatrix} = \begin{bmatrix} 12.263 \\ 1 \\ 0 \end{bmatrix}$$

$$\text{Thus } \mathbf{B}^{-1}\mathbf{P}_{15} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 6 & 36 \end{bmatrix} \begin{bmatrix} 12.263 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -6.263 \end{bmatrix}$$

$$\text{Since } \mathbf{x}_B = (\lambda_{21}, \lambda_{13}, x_5)^T = \begin{bmatrix} 1 \\ 1 \\ 32 \end{bmatrix}, q = 2.$$

Hence $\lambda_{13}$ becomes the leaving variable.

$$\text{So } \mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -6.263 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 6 & 36 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 12.263 & 36 \end{bmatrix}$$

The new basic solution

$$\mathbf{x}_B = (\lambda_{21}, \lambda_{15}, x_5)^T = \mathbf{B}^{-1}(10, 1, 1)^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{c}_B = (47, \mathbf{c}_1\mathbf{y}_{15}, 0) = (47, (1,3)\begin{pmatrix} \\ 1.6316 \end{pmatrix}, 0) = (47, 6.368, 0)$$

**Iteration 6**

$$\mathbf{c}_B\mathbf{B}^{-1} = (47, 6.368, 0) \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 12.263 & 36 \end{bmatrix} = (0, 6.368, 47)$$

Subproblem 1

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1} \begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (0, 6.368, 47) \begin{bmatrix} 5x_1 + 3x_2 \\ 1 \\ 0 \end{bmatrix} - (x_1 + 3x_2)$$

$= - x_1 - 3x_2 + 6.368$

Subject to $x_1 + x_2 \le 9$

$x_1 + 4x_2 \le 8$

The solution is $\mathbf{y}_{16} = (x_1, x_2)^T = (1.4737, 1.6316)^T$, Min $(z_1 - c_1) = 0$

Subproblem 2

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_2\mathbf{y}_2 \\ 0 \\ 1 \end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (0, 6.368, 47)\begin{bmatrix} 4x_3 \\ 0 \\ 1 \end{bmatrix} - (5x_3 + 2x_4) = - 5x_3 - 2x_4 + 47$$

Subject to $x_3 - 5x_4 \le 4$

$x_3 + x_4 \le 10$

The solution is $\mathbf{y}_{26} = (x_3, x_4)^T = (9, 1)^T$, Min $(z_2 - c_2) = 0$

$$\text{Also, } z_5 - c_5 = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} - 0 = (0, 6.368, 47)\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} = 0$$

Since all the $z_j - c_j = 0$, iteration 5 reaches the optimal solution, which is

$\mathbf{x}_1 = (x_1, x_2)^T = \lambda_{15}\mathbf{y}_{15} = 1 * (1.4737, 1.6316)^T = (1.4737, 1.6316)^T$

$\mathbf{x}_2 = (x_3, x_4)^T = \lambda_{21}\mathbf{y}_{21} = 1 * (9, 1)^T = (9, 1)^T$

Max $z = x_1 + 3x_2 + 5x_3 + 2x_4 = 1.4737 + 3 * 1.6316 + 5 * 9 + 2 * 1 = 53.368$

### 4.1.1.2 Example Problem With No Solution

Problem

Maximize $z = - x_1 - 3x_2 - 5x_3 - 2x_4$

Subject to
$$\begin{aligned} 5x_1 + 3x_2 + 4x_3 &\le 10 \\ 5x_1 + x_2 &\le 9 \\ x_1 + 4x_2 &\le 8 \\ x_3 - 5x_4 &\ge 4 \\ x_3 + x_4 &\ge 10 \end{aligned}$$

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

Solution

The original problem can be transformed to:

$$\text{Maximize } z = \sum_{j=1}^{k1} (\mathbf{c}_1\mathbf{y}_{1j})\lambda_{1j} + \sum_{j=1}^{k2} (\mathbf{c}_2\mathbf{y}_{2j})\lambda_{2j}$$

s.t.

$$\sum_{j=1}^{k1}(\mathbf{A}_1\mathbf{y}_{1j})\lambda_{1j} + \sum_{j=1}^{k2} (\mathbf{A}_2\mathbf{y}_{2j})\lambda_{2j} = 10$$

$$\sum_{j=1}^{k1}\lambda_{1j} = 1$$

$$\sum_{j=1}^{k2} \lambda_{2j} = 1$$

where $\lambda_{ij} \geq 0$ ($j = 1, 2, ..., k_i$).

It has the Simplex tableau as follows (using the big-M method):

| | $\lambda_{11}$ $\lambda_{12}$ $\lambda_{13}$ ...... | $\lambda_{21}$ $\lambda_{22}$ $\lambda_{23}$ ...... | $x_5$ | $x_6$ | $x_{11}$ | $x_{12}$ | b |
|---|---|---|---|---|---|---|---|
| z | $\mathbf{c}_1\mathbf{y}_{11}$ $\mathbf{c}_1\mathbf{y}_{12}$ $\mathbf{c}_1\mathbf{y}_{13}$ ...... | $\mathbf{c}_2\mathbf{y}_{21}$ $\mathbf{c}_2\mathbf{y}_{22}$ $\mathbf{c}_2\mathbf{y}_{23}$ ...... | 0 | -M | -M | -M | |
| $x_{12}$ | 0 0 0 ...... | 1 1 1 ...... | 0 | 0 | 0 | 1 | 1 |

Subproblem 1

$\mathbf{x}_1 = (x_1, x_2)^T$, $\mathbf{c}_1 = (1, 3)^T$, $\mathbf{A}_1 = (5, 3)$. The constraints are:

$$5x_1 + x_2 \leq 9$$

$$x_1 + 4x_2 \leq 8$$

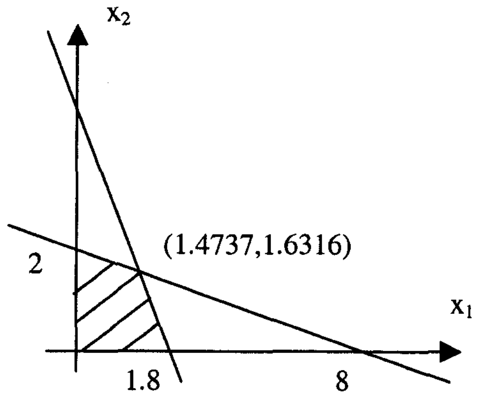



Fig. 13. Feasible region of subproblem 1 of 4.1.1.2

Subproblem 2

$\mathbf{x}_2 = (x_3, x_4)^T$, $\mathbf{c}_2 = (5, 2)^T$, $\mathbf{A}_2 = (4, 0)$. The constraints are:

$$x_3 - 5x_4 \geq 4$$

$$x_3 + x_4 \geq 10$$

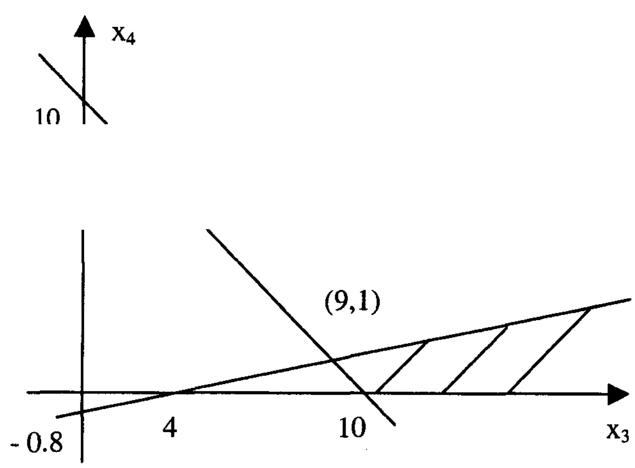



Fig. 14. Feasible region of subproblem 2 of 4.1.1.2

**Iteration 0**

$\mathbf{x}_B = (x_6, x_7, x_8)^T = (10, 1, 1)^T$

$\mathbf{c}_B = (0, -M, -M)$, $\mathbf{B} = \mathbf{B}^{-1} = \mathbf{I}$

**Iteration 1**

$\mathbf{c}_B\mathbf{B}^{-1} = \mathbf{c}_B = (0, -M, -M)$

<u>Subproblem 1</u>

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (0, -M, -M)\begin{bmatrix} 5x_1 + 3x_2 \\ 1 \\ 0 \end{bmatrix} - (-x_1 - 3x_2)$$

$$= x_1 + 3x_2 - M$$

Subject to $x_1 + x_2 \leq 9$

$x_1 + 4x_2 \leq 8$

The solution is $\mathbf{y}_{11} = (x_1, x_2)^T = (0, 0)^T$, Min $(z_1 - c_1) = -M$

<u>Subproblem 2</u>

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_2\mathbf{y}_2 \\ 0 \end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (0, -M, -M)\begin{bmatrix} 4x_3 \\ 0 \end{bmatrix} - (-5x_3 - 2x_4)$$

$$= 5x_3 + 2x_4 - M$$

Subject to $x_3 - 5x_4 \geq 4$

$x_3 + x_4 \geq 10$

The solution is $\mathbf{y}_{21} = (x_3, x_4)^T = (9, 1)^T$, Min $(z_2 - c_2) = -M + 47$

The global min. is min. $(z_1 - c_1) = -M$.

<u>Hence $\lambda_{11}$ becomes the entering variable.</u>

$$\mathbf{P}_{11} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{y}_{11} \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} (5,3)\begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Thus $\mathbf{B}^{-1}\mathbf{P}_{11} = \mathbf{I}^*\mathbf{P}_{11} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

Given $\mathbf{x}_B = (x_6, x_7, x_8)^T = (10, 1, 1)^T$, so q = 2.

Hence $x_7$ becomes the leaving variable.

So $\mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

The new basic solution

$$\mathbf{x}_B = (x_6, \lambda_{11}, x_8)^T = \mathbf{B}^{-1}(10, 1, 1)^T = \begin{bmatrix} 10 \\ 1 \\ 1 \end{bmatrix}$$

$\mathbf{c}_B = (0, \mathbf{c}_1\mathbf{y}_{11}, -M) = (0, 0, -M)$

**Iteration 2**

$\mathbf{c}_B\mathbf{B}^{-1} = (0, 0, -M) * \mathbf{B}^{-1} = (0, 0, -M)$

Subproblem 1

$\begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \end{bmatrix}$ $\begin{bmatrix} 5x_1 + 3x_2 \end{bmatrix}$

Subject to $x_1 + x_2 \le 9$

$x_1 + 4x_2 \le 8$

The solution is $\mathbf{y}_{12} = (x_1, x_2)^T = (0, 0)^T$, Min $(z_1 - c_1) = 0$.

Subproblem 2

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_2\mathbf{y}_2 \\ 0 \\ 1 \end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (0, 0, -M)\begin{bmatrix} 4x_3 \\ 0 \\ 1 \end{bmatrix} - (-5x_3 - 2x_4) = 5x_3 + 2x_4 - M$$

Subject to $x_3 - 5x_4 \ge 4$

$x_3 + x_4 \ge 10$

The solution is $\mathbf{y}_{22} = (x_3, x_4)^T = (9, 1)^T$, Min $(z_2 - c_2) = 47 - M$

The global min. is min. $(z2 - c2) = 47 - M$

<u>Hence $\lambda_{22}$ becomes the entering variable.</u>

$$\mathbf{P}_{22} = \begin{bmatrix} \mathbf{A}_2\mathbf{y}_{22} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} (4,0)\begin{pmatrix} 9 \\ 1 \end{pmatrix} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 36 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{Thus } \mathbf{B}^{-1}\mathbf{P}_{22} = \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{Since } \mathbf{x}_B = (x_6, \lambda_{11}, x_8)^T = \begin{bmatrix} 10 \\ 1 \\ 1 \end{bmatrix}, q = 1.$$

<u>Hence $x_6$ becomes the leaving variable.</u>

$$\text{So } \mathbf{B}^{-1} = \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} * \mathbf{I} = \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix}$$

$$\mathbf{x}_B = (\lambda_{22}, \lambda_{11}, x_8)^T = \mathbf{B}^{-1}(10, 1, 1)^T = \begin{bmatrix} 18 \\ 1 \\ \frac{13}{18} \end{bmatrix}$$

$$\mathbf{c}_B = (\mathbf{c}_2\mathbf{y}_{22}, 0, -M) = (-47, 0, -M)$$

**Iteration 3**

$$\mathbf{c}_B\mathbf{B}^{-1} = (-47, 0, -M) \begin{bmatrix} \frac{1}{36} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{36} & 0 & 1 \end{bmatrix} = (-\tfrac{47}{36} + \tfrac{M}{36}, 0, -M)$$

Subproblem 1

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix}\mathbf{A}_1\mathbf{y}_1\\ 1\\ 0\end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (-\tfrac{47}{36} + \tfrac{M}{36}, 0, -M)\begin{bmatrix}5x_1 + 3x_2\\ 1\\ 0\end{bmatrix} - (-x_1 - 3x_2)$$

$$= [\tfrac{5}{36}(M - 47) + 1]\, x_1 + [\tfrac{1}{12}(M - 47) + 3]\, x_2$$

Subject to $x_1 + x_2 \le 9$

$x_1 + 4x_2 \le 8$

The solution is $\mathbf{y}_{13} = (x_1, x_2)^T = (0, 0)^T$, Min $(z_1 - c_1) = 0$.

Subproblem 2

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix}\mathbf{A}_2\mathbf{y}_2\\ 0\\ 1\end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (-\tfrac{47}{36} + \tfrac{M}{36}, 0, -M)\begin{bmatrix}4x_3\\ 0\\ 1\end{bmatrix} - (-5x_3 - 2x_4)$$

$$= \tfrac{M-2}{9}x_3 + 2x_4 - M$$

Subject to $x_3 - 5x_4 \ge 4$

$x_3 + x_4 \ge 10$

The solution is $\mathbf{y}_{23} = (x_3, x_4)^T = (9, 1)^T$, Min $(z_2 - c_2) = 0$

Hence the iteration stops.

From $\mathbf{x}_B = (\lambda_{22}, \lambda_{11}, x_8)^T = \begin{bmatrix}\frac{5}{18}\\ 1\\ \frac{13}{18}\end{bmatrix}$, $x_8 = \frac{13}{18}$. Since $x_8 > 0$ and is an artificial variable, there is

NO feasible solution to this problem.

### 4.1.2 Unbounded Feasible Region Case: Example Problem With Optimal Solution

Here the "unbounded feasible region case" means that at least one of the subproblems of the original problem has unbounded feasible region. In the following example, subproblem 1 has unbounded feasible region. In the first iteration, subproblem 1 has unbounded solution. Hence its extreme direction is calculated and in the original problem the variable corresponding to subproblem 1 becomes the entering variable. In the rest of the iterations, subproblem 1 has multiple solutions, which is treated as the optimal solution case by picking anyone of these multiple solutions as the optimal solution.

Same as the discussion in Section 4.1.1, even if every subproblem of an original LP problem has unbounded feasible region, the original problem may not have solution. The computation procedure for this case is the same as the Section 4.1.1.2, hence the numerical example is not provided.

Problem

Maximize $z = x_1 + 2x_2 + x_3$

Subject to $x_1 + x_2 + x_3 \leq 12$

$-x_1 + x_2 \leq 2$

$-x_1 + 2x_2 \leq 8$

$(x_1, x_2, x_3 \geq 0)$

Solution

The original problem can be transformed to:

$$\text{Maximize } z = \sum_{j=1}^{k1} (\mathbf{c}_1\mathbf{y}_{1j})\lambda_{1j} + \sum_{j=1}^{k2} (\mathbf{c}_2\mathbf{y}_{2j})\lambda_{2j} + \sum_{j=1}^{l1} (\mathbf{c}_1\mathbf{d}_{1j})\mu_{1j} + \sum_{j=1}^{l2} (\mathbf{c}_2\mathbf{d}_{2j})\mu_{2j}$$

$$\text{s.t. } \sum_{j=1}^{k1} (\mathbf{A}_1\mathbf{y}_{1j})\lambda_{1j} + \sum_{j=1}^{k2} (\mathbf{A}_2\mathbf{y}_{2j})\lambda_{2j} + \sum_{j=1}^{l1} (\mathbf{A}_1\mathbf{d}_{1j})\mu_{1j} + \sum_{j=1}^{l2} (\mathbf{A}_2\mathbf{d}_{2j})\mu_{2j} = 12$$

$$\sum_{j=1}^{k1} \lambda_{1j} = 1$$

$$\sum_{j=1}^{k2} \lambda_{2j} = 1$$

where $\lambda_{ij} \geq 0$ ($j = 1, 2, \ldots, k_i$), $\mu_{ij} \geq 0$ ($j = 1, 2, \ldots, l_i$)

It has the Simplex tableau as follows (using the big-M method):

| | $\lambda_{11}$ $\lambda_{12}$ $\lambda_{13}\ldots$ | $\lambda_{21}$ $\lambda_{22}$ $\lambda_{23}\ldots$ | $\mu_{11}$ $\mu_{12}$ $\mu_{13}\ldots$ | $\mu_{21}$ $\mu_{22}$ $\mu_{23}\ldots$ | $x_4$ | $x_5$ | $x_6$ | b |
|---|---|---|---|---|---|---|---|---|
| Z | $c_1\mathbf{y}_{11}$ $c_1\mathbf{y}_{12}$ $c_1\mathbf{y}_{13}\ldots$ | $c_2\mathbf{y}_{21}$ $c_2\mathbf{y}_{22}$ $c_2\mathbf{y}_{23}\ldots$ | $c_1\mathbf{d}_{11}$ $c_1\mathbf{d}_{12}$ $c_1\mathbf{d}_{13}\ldots$ | $c_2\mathbf{d}_{21}$ $c_2\mathbf{d}_{22}$ $c_2\mathbf{d}_{23}\ldots$ | 0 | -M | -M | |
| $x_4$ | $A_1\mathbf{y}_{11}$ $A_1\mathbf{y}_{12}$ $A_1\mathbf{y}_{13}\ldots$ | $A_2\mathbf{y}_{21}$ $A_2\mathbf{y}_{22}$ $A_2\mathbf{y}_{23}\ldots$ | $A_1\mathbf{d}_{11}$ $A_1\mathbf{d}_{12}$ $A_1\mathbf{d}_{13}\ldots$ | $A_2\mathbf{d}_{21}$ $A_2\mathbf{d}_{22}$ $A_2\mathbf{d}_{23}\ldots$ | 1 | 0 | 0 | 12 |
| $x_5$ | 1 1 1 ... | 0 0 0 ... | 0 0 0 ... | 0 0 0 ... | 0 | 1 | 0 | 1 |
| $x_6$ | 0 0 0 ... | 1 1 1 ... | 0 0 0 ... | 0 0 0 ... | 0 | 0 | 1 | 1 |

<u>Subproblem 1</u>

$\mathbf{x}_1 = (x_1, x_2)^T$, $\mathbf{c}_1 = (1, 2)^T$, $\mathbf{A}_1 = (1, 1)$

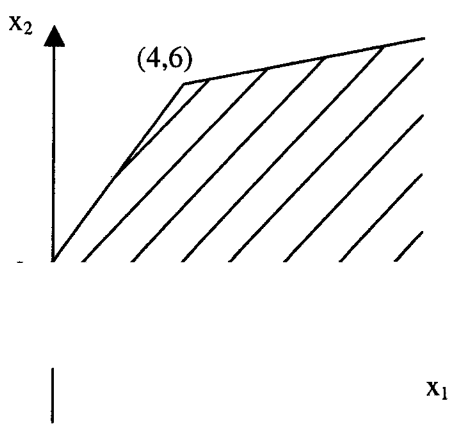



Fig. 15. Feasible region of subproblem 1 of 4.1.2

<u>Subproblem 2</u>

$\mathbf{x}_2 = (x_3)$, $\mathbf{c}_2 = (1)$, $\mathbf{A}_2 = (1)$

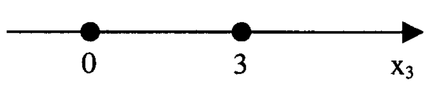



Fig. 16. Feasible region of subproblem 2 of 4.1.1.2

Having these two figures, we will be able to find the local optimal value corresponding to each subproblem very easily, just by observing the figures instead of by using the simplex iterations.

**Iteration 0**

$$\mathbf{x}_B = (x_4, x_5, x_6)^T = \begin{bmatrix} 12 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{c}_B = (0, -M, -M),\ \mathbf{B} = \mathbf{B}^{-1} = \mathbf{I}$$

**Iteration 1**

$$\mathbf{c}_B\mathbf{B}^{-1} = \mathbf{c}_B = (0, -M, -M)$$

Subproblem 1

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (0, -M, -M)\begin{bmatrix} x_1 + x_2 \\ 1 \\ 0 \end{bmatrix} - (x_1 + 2x_2)$$

$$= -x_1 - 2x_2 - M$$

Subject to $\quad -x_1 + x_2 \le 2$

Notice that this problem is the same as the problem in section 3.3.2. Hence, this problem has **unbounded solution** and the optimal direction

$$\mathbf{d}_{11} = (x_1, x_2)^T = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

Hence $\mu_{11}$ directly becomes the entering variable.

$$\mathbf{P}_{11} = \begin{bmatrix} \mathbf{A}_1\mathbf{d}_{11} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} (1,1)\begin{pmatrix} 2 \\ 1 \end{pmatrix} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$$

Thus $\mathbf{B}^{-1}\mathbf{P}_{11} = \mathbf{I}\,\mathbf{P}_{11} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$

Given $\mathbf{x}_B = (x_4, x_5, x_6)^T = \begin{bmatrix} 12 \\ 1 \\ 1 \end{bmatrix}$, so q = 1.

Hence $x_4$ becomes the leaving variable.

$$\text{So } \mathbf{B}^{-1} = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \mathbf{I} = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The new basic solution

$$\mathbf{x}_B = (\mu_{11}, x_5, x_6)^T = \mathbf{B}^{-1}\begin{bmatrix} 12 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{c}_B = (\mathbf{c}_1\mathbf{d}_{11}, -M, -M) = ((1, 2)\begin{pmatrix} 2 \\ 1 \end{pmatrix}, -M, -M) = (4, -M, -M)$$

**Iteration 2**

$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

Subproblem 1

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (\tfrac{4}{3}, -M, -M)\begin{bmatrix} x_1 + x_2 \\ 1 \\ 0 \end{bmatrix} - (x_1 + 2x_2)$$

$$= \tfrac{1}{3}x_1 - \tfrac{2}{3}x_2 - M$$

Subject to $-x_1 + x_2 \le 2$

$-x_1 + 2x_2 \le 8$

This problem has multiple solutions, one of them is $\mathbf{y}_{12} = (x_1, x_2)^T = (4, 6)^T$.

Min $(z_1 - c_1) = -M - \frac{4}{3}$

<u>Subproblem 2</u>

$$\text{Min}\ (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix}\mathbf{A}_2\mathbf{y}_2\\ 0\\ 1\end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (\tfrac{4}{3}, -M, -M)\begin{bmatrix}x_3\\ 0\\ 1\end{bmatrix} - (x_3) = \tfrac{1}{3}x_3 - M$$

Subject to $x_3 \leq 3$

The solution is $\mathbf{y}_{22} = (x_3) = (0)^T$, Min $(z_2 - c_2) = -M$

$$\text{Also, } z_4 - c_4 = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix}1\\ 0\\ 0\end{bmatrix} - 0 = \tfrac{4}{3}$$

The global min. is min. $(z_1 - c_1) = -M - \frac{4}{3}$

<u>Hence $\lambda_{12}$ becomes the entering variable.</u>

$$\mathbf{P}_{12} = \begin{bmatrix}\mathbf{A}_1\mathbf{y}_{12}\\ 1\\ 0\end{bmatrix} = \begin{bmatrix}(1,1)\begin{pmatrix}4\\ 6\end{pmatrix}\\ 1\\ 0\end{bmatrix} = \begin{bmatrix}10\\ 1\\ 0\end{bmatrix}$$

$$\begin{bmatrix}1 & 0 & 0\\ & & \\ & & \end{bmatrix}\begin{bmatrix}10\\ \\ \end{bmatrix} \quad \begin{bmatrix}10\\ \\ \end{bmatrix}$$

$$\text{Since } \mathbf{x}_B = (\mu_{11}, x_5, x_6)^T = \begin{bmatrix}4\\ 1\\ 1\end{bmatrix}, q = 2.$$

<u>Hence $x_5$ becomes the leaving variable.</u>

So $\mathbf{B}^{-1} = \begin{bmatrix} 1 & -\frac{10}{3} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & -\frac{10}{3} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

The new basic solution

$$\mathbf{x}_B = (\mu_{11}, \lambda_{12}, x_6)^T = \mathbf{B}^{-1} \begin{bmatrix} 12 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{c}_B = (4, \mathbf{c}_1\mathbf{y}_{12}, -M) = (4, (1, 2)\begin{pmatrix} 4 \\ 6 \end{pmatrix}, -M) = (4, 16, -M)$$

**Iteration 3**

$$\mathbf{c}_B\mathbf{B}^{-1} = (4, 16, -M) \begin{bmatrix} \frac{1}{3} & -\frac{10}{3} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = (\tfrac{4}{3}, \tfrac{8}{3}, -M)$$

Subproblem 1

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1} \begin{bmatrix} \mathbf{A}_1\mathbf{y}_1 \\ 1 \\ 0 \end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (\tfrac{4}{3}, \tfrac{8}{3}, -M) \begin{bmatrix} x_1 + x_2 \\ 1 \\ 0 \end{bmatrix} - (x_1 + 2x_2)$$

Subject to $-x_1 + x_2 \leq 2$

$-x_1 + 2x_2 \leq 8$

This problem has multiple solutions, one of them is $\mathbf{y}_{13} = (x_1, x_2)^T = (4, 6)^T$.

Min $(z_1 - c_1) = 0$

Subproblem 2

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1} \begin{bmatrix} \mathbf{A}_2\mathbf{y}_2 \\ 0 \\ 1 \end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (\tfrac{4}{3}, \tfrac{8}{3}, -M) \begin{bmatrix} x_3 \\ 0 \\ 1 \end{bmatrix} - (x_3) = \tfrac{1}{3}x_3 - M$$

Subject to $x_3 \leq 3$

The solution is $\mathbf{y}_{23} = (x_3) = (0)$, Min $(z_2 - c_2) = -M$

Also, $z_4 - c_4 = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix}1\\0\\0\end{bmatrix} - 0 = \frac{4}{3}$

The global min. is min. $(z_2 - c_2) = -M$

<u>Hence $\lambda_{23}$ becomes the entering variable.</u>

$$\mathbf{P}_{23} = \begin{bmatrix}\mathbf{A}_2\mathbf{y}_{23}\\0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$\text{Thus } \mathbf{B}^{-1}\mathbf{P}_{12} = \begin{bmatrix}\frac{1}{3} & -\frac{10}{3} & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

Since $\mathbf{x}_B = (\mu_{11}, \lambda_{12}, x_6)^T = \begin{bmatrix}\frac{2}{3}\\1\\1\end{bmatrix}$, $q = 3$.

<u>Hence $x_6$ becomes the leaving variable.</u>

$$\text{So } \mathbf{B}^{-1} = \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{3} & -\frac{10}{3} & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} = \begin{bmatrix}\frac{1}{3} & -\frac{10}{3} & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$$

$$\mathbf{x}_B = (\mu_{11}, \lambda_{12}, \lambda_{23})^T = \mathbf{B}^{-1}\begin{bmatrix}\text{[illegible]}\\1\\1\end{bmatrix} = \begin{bmatrix}3\\1\\1\end{bmatrix}$$

$\mathbf{c}_B = (4, 16, \mathbf{c}_2\mathbf{y}_{23}) = (4, 16, 0)$

**Iteration 4**

$$\mathbf{c}_B\mathbf{B}^{-1} = (4, 16, 0)\begin{bmatrix}\frac{1}{3} & -\frac{10}{3} & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} = \left(\frac{4}{3}, \frac{8}{3}, 0\right)$$

Subproblem 1

$$\text{Min } (z_1 - c_1) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix}\mathbf{A}_1\mathbf{y}_1\\ 1\\ 0\end{bmatrix} - \mathbf{c}_1\mathbf{y}_1 = (\tfrac{4}{3}, \tfrac{8}{3}, 0)\begin{bmatrix}x_1 + x_2\\ 1\\ 0\end{bmatrix} - (x_1 + 2x_2)$$

$$= -\tfrac{1}{3}(-x_1 + 2x_2) + \tfrac{8}{3}$$

Subject to $-x_1 + x_2 \leq 2$

$-x_1 + 2x_2 \leq 8$

This problem has multiple solutions, one of them is $\mathbf{y}_{13} = (x_1, x_2)^T = (4, 6)^T$.

$\text{Min } (z_1 - c_1) = 0$

Subproblem 2

$$\text{Min } (z_2 - c_2) = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix}\mathbf{A}_2\mathbf{y}_2\\ 0\\ 1\end{bmatrix} - \mathbf{c}_2\mathbf{y}_2 = (\tfrac{4}{3}, \tfrac{8}{3}, 0)\begin{bmatrix}x_3\\ 0\\ 1\end{bmatrix} - (x_3) = \tfrac{1}{3}x_3$$

Subject to $x_3 \leq 3$

The solution is $\mathbf{y}_{23} = (x_3) = (0)$, $\text{Min } (z_2 - c_2) = 0$

$$\text{Also, } z_4 - c_4 = \mathbf{c}_B\mathbf{B}^{-1}\begin{bmatrix}1\\ 0\\ 0\end{bmatrix} - 0 = \tfrac{4}{3}$$

$$\mathbf{x}_1 = (x_1, x_2)^T = \mu_{11}\mathbf{d}_{11} + \lambda_{12}\mathbf{y}_{12} = \tfrac{2}{3}\begin{pmatrix}2\\ 1\end{pmatrix} + 1\begin{pmatrix}4\\ 6\end{pmatrix} = \begin{pmatrix}\tfrac{16}{3}\\ \tfrac{20}{3}\end{pmatrix}$$

$\mathbf{x}_2 = (x_3) = \lambda_{23}\mathbf{y}_{23} = 1 * 0 = 0$

The optimal objective solution

$$\text{Max } z = \mathbf{c}_1\mathbf{x}_1 + \mathbf{c}_2\mathbf{x}_2 = (1, 2)\begin{pmatrix}\tfrac{16}{3}\\ \tfrac{20}{3}\end{pmatrix} + 1 * 0 = \tfrac{56}{3} = 18.667$$

### 4.2 Parallel Algorithm of the Decomposition Principle

From the preceding numeric examples done in the sequential computation procedure, we can write the flow chart of its parallel algorithm as follows:

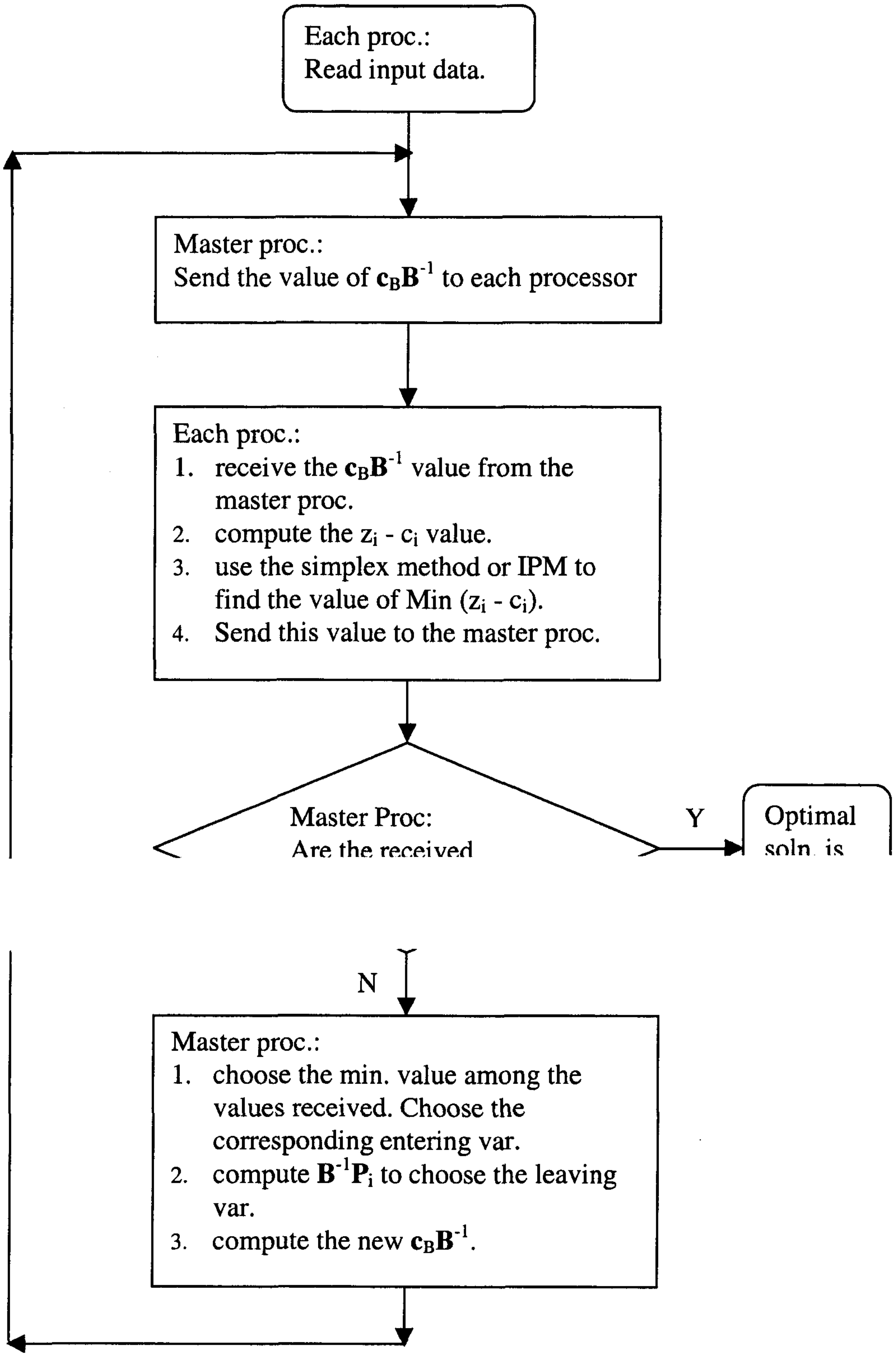



Fig. 17. Flow chart of the parallel algorithm of the decomposition principle

A code of parallel algorithm of the decomposition principle is implemented in MPI/Fortran based on the above flow chart. Different size of large-scale LP problems and different number of processors are used to test for its performance [9]. Description of problems' sizes, number of processors (= np) used, computational time (in seconds, including I/O), parallel speed-up and efficiency factors (on Sun/Sparc Rhino workstation in the CEE department) is described and tabulated in Tables 1 – 3. The definitions for speed-up and efficiency factor are:

$$\text{Speed-up} = \frac{\text{computation time by 1 processor}}{\text{computation time by n processors}}$$

$$\text{Efficiency} = \frac{\text{speed - up}}{\text{number of processors (used to test the speed - up)}}$$

In all these tables, 1 common constraint is used, and the following notations are defined:

nblksize = the size of each block

nblocks = number of blocks

nconviter = number of converged iterations

The total number of constraints (= ntotcon) and the total number of design variables (= ndv) can be given as:

| np | time | speedup | Efficiency |
|---|---|---|---|
| 1 | 122 | | |
| 2 | 62 | 1.97 | 99% |
| 3 | 43 | 2.84 | 95% |
| 4 | 35 | 3.49 | 87% |

Table 1: Numerical results of the parallel decomposition principle, Case 1 (nblksize = 20, nblocks = 80, nconviter = 132)

| np | time | speedup | Efficiency |
|---|---|---|---|
| 1 | 147 | | |
| 2 | 76 | 1.93 | 97% |
| 3 | 53 | 2.77 | 92% |
| 4 | 43 | 3.42 | 85% |

Table 2: Numerical results of the parallel decomposition principle, Case 2 (nblksize = 40, nblocks = 40, nconviter = 42)

| np | time | speedup | efficiency |
|---|---|---|---|
| 1 | 266 | | |
| 2 | 135 | 1.97 | 99% |
| 3 | 95 | 2.80 | 93% |
| 4 | 75 | 3.55 | 89% |

Table 3: Numerical results of the parallel decomposition principle, Case 3

The above result shows that the parallel MPI/FORTRAN implementation has resulted in good parallel speedup, and efficiency factors. The MPI/FORTRAN used in the developed code will facilitate the porting of this parallel code to different computer platforms. The developed parallel MPI/FORTRAN LP decomposition code also offers computer memory advantages, since large number of independent constraints can be stored by different number of processors. Thus, large-scale (block diagonal constraints) LP problems that cannot be solved by a single processor (due to computer memory restrictions) can be "quickly" solved by the developed parallel MPI code.

# CHAPTER V

# A NEW DECOMPOSITION ALGORITHM: DIVISION BY THE INTERIOR POINT

The numerical study of the last chapter shows that the decomposition principle can be used for effective parallel computation. However, one major problem is that it only achieves the satisfactory result for the LP problems with the block angular structure. This chapter discusses a new parallel decomposition algorithm that saves time and can be used for general LP problems [10]. Basically, this algorithm divides the feasible region of a LP problem into multiple subregions (subproblems) based on the found interior point. Then multiple processors are used to solve these subproblems.

## 5.1 Introduction

For a linear programming problem, the collection of extreme points of the feasible

the problem is nothing but an extreme point with the optimal objective value. The simplex method is a procedure that moves from one extreme point to another extreme point with a better objective. Hence, roughly speaking, the number of iterations of the simplex method is proportional to the number of extreme points of the problem.

The idea of the "division by the interior point" algorithm is to decrease the number of extreme points by dividing the feasible region into multiple subregions. If we can divide the feasible region into multiple subregions, the number of the extreme points of each subregion will be greatly decreased, compared to the original feasible region. For example, if the feasible region is a regular octagon, it has 8 extreme points. If we draw a horizontal line and a vertical line passing through the centroid of area, the feasible region is divided into 4 same polygons, each with 5 extreme points. Let subproblem be a LP problem with such a subregion as its feasible region and the original objective function as its objective function. It is obvious that the optimal solution of the original LP problem is the maximum/minimum value of the optimal solutions of all the subproblems. When each subproblem is solved by an individual processor for parallel computation, the original problem can be solved much faster.

To divide the feasible region into subregions, we need at least one interior point inside the feasible region as a base point for the dividing hyperplanes. There are existing algorithms to find the initial interior point, such as the algorithm discussed in Section 2.2.2 of Chapter 2. We will see later that the algorithm discussed in Chapter 2 is perfect for our purpose. For real world optimization problems, an interior point near the center of the feasible region can be reasonably derived directly from the context of the problem, as demonstrated in the numerical example of Section 5.3.

Last, but not least, it should be noted that in order to decrease the iteration number, extra constraints are added into the original problem, making the problem become even "larger". This is the contrary of the common concept that the more constraints, the more

[illegible]

n (number of variables) and m (number of constraints). Indeed, n and m decide the size of the problem. However, is size everything? Imagine two problems with the same value of n and the same value m. If one problem has much less extreme points than the other, it is conceivable that its number of converged iterations, and hence the computational time, will be much less. Now let's take a look at a numerical example:

<u>Problem:</u>

Max. $x_1 + 2x_2$

subject to:

$$3.7321\ x_1 + x_2 \leq 1635.1$$

$x_1 + x_2 \leq 650.27$

$0.26795\ x_1 + x_2 \leq 438.13$

$-\ 0.26795\ x_1 + x_2 \leq 334.60$

$-\ x_1 + x_2 \leq 263.90$

$-\ 3.7321\ x_1 + x_2 \leq 193.19$

$0.26795\ x_1 - x_2 \leq 51.764$

$x_1 - x_2 \leq 263.90$

$3.7321\ x_1 - x_2 \leq 1248.8$

$3.7321\ x_1 + x_2 \geq 193.19$

$x_1 + x_2 \geq 122.47$

$0.26795\ x_1 + x_2 \geq 51.764$

$(x_1, x_2 \geq 0)$

If we draw a figure of the above problem, it will show that the feasible region of this example is a regular polygon with 12 sides, with each constraint as one of the sides. Using the Simplex method, it takes 7 iterations to find the optimal solution ($x_1 = 289.78$

[illegible]

divide the feasible region into 2 subregions. Correspondingly, the original problem is decomposed into 2 subproblems. These two subproblems are exactly the same as the original problem, except that each with a new constraint added. Let the subproblem with the added constraint $x_1 \leq 95.341$ be subproblem 1 and the subproblem with the added constraint $x_1 \geq 95.341$ be subproblem 2. It takes 5 iterations for the subproblem 1 to find the optimal solution ($x_1 = 95.341$ and $x_2 = 359.24$) with the optimal objective value 813.82. It also takes 5 iterations for the subproblem 2 to find the optimal solution ($x_1 = 289.78$, $x_2 = 360.49$) with the optimal objective value 1010.76. Since 1010.76 > 813.82, 1010.76 is the solution of the original problem. And the optimal solution to the original problem is $x_1 = 289.78$, $x_2 = 360.49$.

Although the iteration number is only decreased from 7 to 5 for this small LP example, it can be greatly decreased for large scale LP problem. For example, if we extend the above problem to a regular polygon with 8000 sides, it takes 3002 iterations for the Simplex method to solve it. If we solve the two subproblems divided by the interior point $x_1 = x_2 = 7002.82$, it takes 1597 iterations to solve subproblem 1 and 1408 iterations to solve subproblem 2.

The above division method of the feasible region in 2 dimensional space can be extended into multi-dimensional space, using the following strategy: let the known (solved) starting point be $x_1 = x_1'$, $x_2 = x_2'$, ..., $x_n = x_n'$. The original feasible region can be divided into $2^m$ regions (subproblems) by adding the following constraints into the original problem, respectively:

$x_1 \le x_1'$ $\quad\quad$ $x_1 \ge x_1'$ $\quad\quad$ (2 regions)

/ \ $\quad\quad$ / \

+ $\quad$ $x_2 \le x_2'$ $\quad$ $x_2 \ge x_2'$ $\quad$ $x_2 \le x_2'$ $\quad$ $x_2 \ge x_2'$ $\quad\quad$ (4 regions)

[illegible]

/ \ $\quad$ / \ $\quad$ / \ $\quad$ / \ $\quad$ / \ $\quad$ / \ $\quad$ / \ $\quad$ / \

### 5.2 Parallelizing the Division by the Interior Point Algorithm

The parallelization of the division by the interior point algorithm is straightforward, as shown by the flow chart (Figure 18).

As we know, communication between the master processor and the other processors is nothing but an overhead for the effectiveness of a parallel algorithm. In the flow chart, the words "send" and "receive" are underlined to show the communication between the master processor and the other processors. We can see that very little information needs to be exchanged: only two vectors of size n and one scalar.

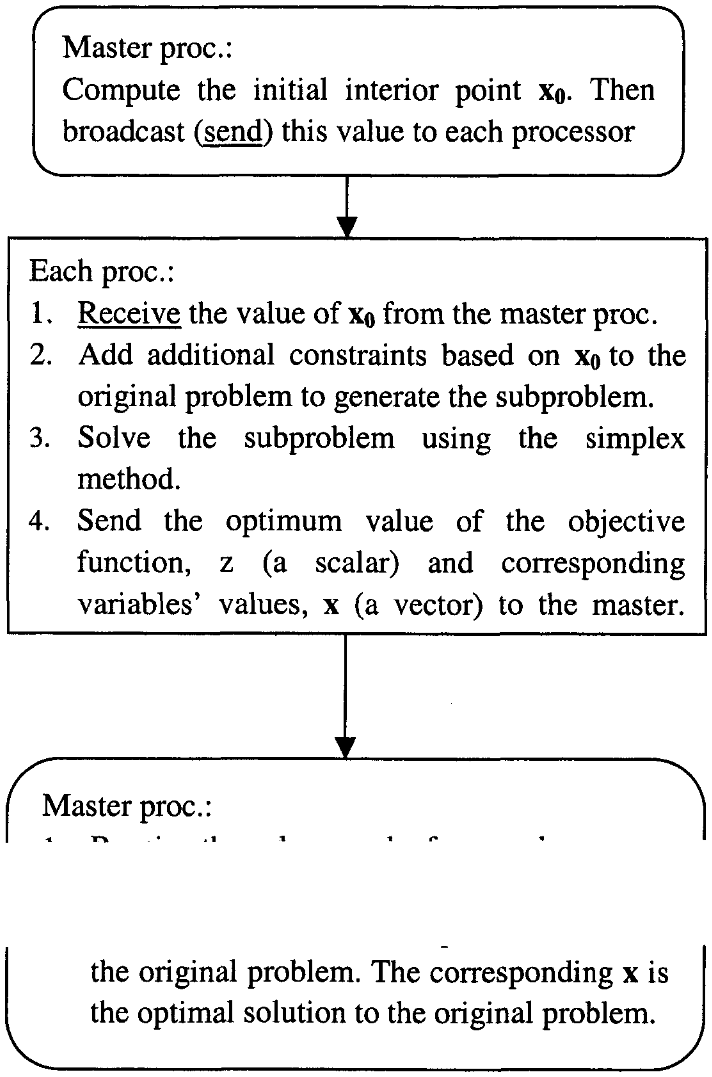



Fig.18: Flow chart of the parallel algorithm of Division by the Interior Point

To make a parallel algorithm effective, another important point is to make the computation work divided as equally as possible for each processor. However, before we actually solve a LP problem, we have no idea what its feasible region looks like, not to mention to divide the feasible region in the way that each subregion has the same amount of extreme points. If we can find an interior point as close to the center of the feasible region as possible, that would be our best bet to divide the feasible region as equally as possible. As we discussed in Section 2.2.2, the algorithm introduced there is such an algorithm. Another advantage of that algorithm is that very little computation needs to be done to "find" the interior point. From the preceding flow chart we can see that since the work to find the interior point cannot be parallelized, it is important to keep its computation as little as possible to make the whole parallel algorithm more effective.

Actually, the initial interior point in the Section 2.2.2 is not calculated, but "given" as $\mathbf{x}_0 = (1 \quad 1 \quad \ldots \quad 1)^T$. This brings another benefit: there is no need for the master processor to "send" its value of the interior point to each processor because they have this information from the very beginning. Hence the communication time is saved.

**5.3 Numerical studies**

A code of parallel algorithm of the "division by the interior point" is implemented in MPI/Fortran and the optimization problem of school desegregation is used as the large-scale test problem. The objective of the school desegregation problem is to

range must be satisfied, and school's capacities in different school districts need to be satisfied also. For this optimization problem, the number of variables (NVAR) = NI x NJ x NK, and the number of constraints (NCON) = NI x NJ + NK + 2 x NK X NI, where NI = number of ethnic groups, NJ = number of school districts, and NK = number of schools.

Based on the context of this problem, we can see that there are some obvious interior points. For example, the number of students of ethnic group i living in district j divided by the number of schools is such a point. It is used as the dividing base point for the test problem.

The Different size of large-scale school desegregation problems and different number of processors (denoted as np in the following tables) are used to test for the code's performance [11]. The results are tabulated as follows:

| np | NVARxNCON | No. of iterations | Time (sec) | Speedup |
|---|---|---|---|---|
| 1 | 2500 x 375 | 1764 | 198 | |
| 2 | 2501 x 376 | 996 | 144 | 1.38 |
| 4 | 2501 x 377 | 959 | 134 | 1.48 |
| 8 | 2501 x 378 | 996 | 126 | 1.58 |

Table 4: Numerical results of the parallel division by the interior point procedure , Case 1 (NI=5, NJ = 20, NK = 25)

| np | NVARxNCON | No. of iterations | Time (sec) | Speedup |
|---|---|---|---|---|
| 1 | 3750 x 475 | 2672 | 575 | |
| 2 | 3751 x 476 | 1040 | 271 | 2.12 |
| 8 | 2501 x 478 | 1040 | 248 | 2.32 |

Table 5: Numerical results of the parallel division by the interior point procedure , Case 2 (NI=6, NJ = 25, NK = 25)

# CHAPTER VI
# CONCLUSIONS AND FUTURE WORKS

## 6.1 Conclusions

In this study, two linear programming decomposition procedures are examined, then implemented and tested under the parallel computation environment. The first decomposition procedure, the decomposition principle, is custom-made for the linear programming problems in the special block-angular structure, while the second decomposition procedure can be applied to any linear programming problems. Both of the simplex method and the Interior Point Method are used in this study as subroutines to solve LP problems.

In the decomposition principle procedure, the unbounded solution case has been paid special attention since its solution procedure is different. The related concept of extreme direction is explained. Methods to find the extreme direction in both the simplex [illegible] the method to find an initial interior point is discussed.

Small numerical examples with step-by-step calculations are included in this study to illustrate both of the two parallel decomposition procedures. The tabulated test results of these two parallel decomposition algorithms show satisfactory efficiency in solving large-scale linear programming problems.

## 6.2 Future Research

The algorithm of the decomposition principle procedure requires the problems in the block-angular format. If a general linear programming problems can be manipulated and transformed into this special format, it can be solved with this algorithm. Future work should be done on how to transform a general linear programming problem into this format efficiently so that the computation time saved by the decomposition principle procedure will not be wasted by the extra effort of the transformation.

The decomposition procedure of the "division by the interior point" idea can be applied to general linear programming problems. However, nothing is free. The generality of this algorithm is paid by the price of its efficiency: the performance of this procedure is subject to many factors such as the "shape" of the problem's feasible region; the location of the interior base point; the method of dividing the feasible region, etc. In short, it is difficult, if not impossible, to divide the feasible region "equally" into subregions. Future work should be done on the methods of dividing the feasible region.

So far, the research on computational complexity of the simplex method is focused on the size of the LP problem, i.e, the value of n (number of variables) and m (number of constraints). As discussed in Chapter V, it is not only the size, but also the number of extreme points that directly links to the computational complexity. This knowledge is the foundation of the "division by the interior point" decomposition idea. In order to make [illegible]

and exploited, which would be a very interesting future research topic.

# Bibliography

[1] Dantzig, G. B., <<Linear Programming and Extensions>>, Princeton University Press, Princeton, N. J., 1963

[2] Fang, Shu-Cherng and Puthenpura, Sarat, <<Linear Optimization and Extensions: Theory and Algorithms>>, Prentice Hall, 1993

[3] Karmarka, N., "A new polynominal-time algorithm for linear programming", Proceedings of the 16th Annual ACM Symposium on the Theory of Computing, pp. 302-311, 1984

[4] Dantzig, G. B. and Wolfe, Philip, "A Decomposition Principle For Linear Programs", the RAND Corporation, P-1544, Dec. 10, 1959

[5] Nguyen, D. T.; Bai, Y.; Qin, J.; Han, B. and Hu, Y., "Computational Aspects of Linear Programming Simplex Method", Advances in Engineering Software, Vol. 31 pp. 539-545, Elsevier, 2000

[6] Nguyen, D. T.; Runesha, H.; Belegunda, A.D. and Chandrupatla, T. R., "Interior point method & indefinite sparse solver for linear programming problems", Proceedings of

Performance Computers and Workstations, Williamsburg, VA, Oct. 15-17, 1997

[7] Bazaraa, M.; Jarvis, J. and Sherali, H., "Linear Programming and Network Flows", 2nd ed., Wiley, New York, 1990

[8] Pacheco, P. S., "Parallel Programming with MPI", Morgan Kaufmann, 1997

[9] Hu, Y.; Nguyen, D. T. and Gould, Kevin, "Large Scale Linear Programming Problems by Decomposition and Parallel Procedures", the 8-th AIAA / USAF / NASA / ISSMO Symposium on Multidisciplinary Optimization, Westin Long Beach, Long Beach, California, September 6-8, 2000

[10] Hu, Y.; Gould, K. and Nguyen, D. T., "Linear Programming Domain Decomposition Solutions Using Simplex and Interior Point Methods Under Parallel Computer

Environments", 9th International Conference on Numerical Methods and Computational Mechanics, Miskolc, Hungary, July 15-19, 2002

[11] Private communication with Prof. Duc T. Nguyen and his Ph.D student Siroj Tungkahotara who has developed the computer program to test large-scale school desegregation problem

## Supplemental Source

[1] Taha, H. A., "Operations Research: An Introduction", Prentice Hall, 1997

[2] Baumol, W. J. and T. Fabian, "Decomposition Pricing for Decentralization and External Economies", Management Science, Vol. 11, No. 1, 1964

[3] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T. and Flannery, B. P., " Numerical Recipes in Fortran 77: The Art of Scientific Computing", 2nd edition, Cambridge University Press, 1992

[4] Wolfe, P., "The Composite Simplex Algorithm", SIAM Review, Vol. 7, No. 1, 1965

[5] Wolfe, P., "A Technique for Resolving Degeneracy in Linear Programming", RAND Report RM-2995-PR. The RAND Corporation, Santa Monica, CA, May 1962

memory computers", Proceedings of the ICES '95 (International Conference on Computational Engineering Science), the Ritz-Carlton Hotel, Mauna Lani, Big Island of Hawaii, July 30-August 3, 1995

[7] "Parallel Decomposition of Linear Programs", Dept. of Operation Research, Stanford University, CA, Nov. 1989. (Thesis, Technical Report)

[8] Hooker, J., "Karmarkar's Linear Programming Algorithm", Interfaces, Vol. 16, No. 4, pp. 75-90, 1986

# VITA

Yusong Hu
Email: yusonghu@hotmail.com

**Education**

- **Ph.D.**, Civil Engineering, Old Dominion University, Norfolk, VA (5/2004).
- **M.S.**, Structural Engineering, Beijing Jiaotong University, Beijing, China (4/1992).
- **B.S.**, Structural Engineering, Hefei United University, Hefei, China (7/1989)

**Experience**

- 08/98 – 08/02: **Research Assistant**, Dept. of Civil and Environmental Engineering, Old Dominion University, Norfolk, VA. Research topic: high performance scientific computation with application in structural engineering.
- 04/92 - 05/98: **Assistant Professor,** Dept. of Civil Engineering, Beijing Polytechnic University, Beijing, China.
- 09/89 - 04/92: **Research Assistant**, Dept. of Civil Engineering, Northern Jiaotong University, Beijing, China. Research topic: LISP program implementation of developing mechanism of expert system for structural engineering design.

**Publications and conference papers**

- **Linear Programming Domain Decomposition Solutions Using Simplex and Interior Point Methods Under Parallel Computer Environments**, Y. Hu, K.E. Gould and D.T. Nguyen, 9th International Conference on Numerical Methods and

Bai, J. Qin, and Yusong Hu, Advances in Engineering Software, pages 539-546, V31, 2000, Elsevier
- **Large Scale Linear Programming Problems by Decomposition and Parallel Procedures**, Hu, Yusong, Nguyen, D.T. and Gould, Kevin, the 8-th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Optimization, September 6-8, 2000, Westin Long Beach, Long Beach, California
- **Decomposition Procedures For Linear Programming Problems: Sequential computer Environments,** Yusong Hu, D.T. Nguyen, and K.E. Gould, Virginia Academy of Science Conference, May 24-26, 2000, Radford, Virginia
- **Computational Aspects of Linear Programming Simplex Method,** Nguyen, D.T.; Bai, Y.; Qin, J. and Hu, Y., 5 th National Symposium on Large-Scale Analysis, Design and Intelligent Synthesis Environment, October 12-15, 1999, Williamsburg, VA.
- **Computational Issues In the Revised Simplex Method For LP Problems,** D.T. Nguyen, Yu Bai, J. Qin and Yusong Hu, Virginia Academy of Science Conference, May 25- 28, 1999, Norfolk, VA.