論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Optimization over the Efficient Set: A New MIP Approach and Its Applications
著者(和文)	呂寛
Author(English)	Kuan Lu
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第11440号, 授与年月日:2020年3月26日, 学位の種別:課程博士, 審査員:水野 眞治,松井 知己,宮川 雅巳,塩浦 昭義,中田 和秀
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第11440号, Conferred date:2020/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctoral Thesis

Optimization over the Efficient Set: A New MIP Approach and Its Applications

by

Kuan Lu
17D10460

Supervisor: **Prof. Shinji Mizuno**

Department of Industrial Engineering and Economics
Tokyo Institute of Technology

March 2020

Acknowledgments

The completion of this dissertation could not have been possible without the assistance and help of so many people.

First, I wish to express my sincere gratitude to my supervisor Prof. Shinji Mizuno for his professional instruction as well as generous help. During these three years, I have highly benefited and learned a lot from him about doing research and presentation. Those inspiring and happy moments that we pondered and discussed problems together will always shine through my academic career and even the whole life. It is so lucky to have Prof. Mizuno as my mentor, who always stands by my side whenever I am in need.

I am grateful to my fellows. Together we studied, discussed as well as recreated, campus life became much more attractive and colorful with their company. What is more, they continually conveyed excitement regarding studying and a spirit of adventure regarding research, which had a substantial positive impact on me.

Finally, I owe my deepest gratitude to my family and my friends for their unconditional support and love over the years. It is your trust and support that make it possible for me to pursue my academic career freely and devotedly.

This research is supported in part by Grant-in-Aid for Science Research (A) 19H00808 and Grant-in-Aid for Scientific Research (C) 17K01272 of Japan Society for the Promotion of Science.

Abstract

This thesis concerns an optimization problem over the efficient set (OE problem) of a multiobjective linear programming problem and its applications. The optimal solutions to these problems can be obtained by solving equivalent mixed integer programming (MIP) problems with an MIP solver.

In this thesis, we firstly propose a new equivalent MIP problem of the OE problem and obtain an optimal solution with an MIP solver. Compared with the previous MIP approach by Sun, the proposed approach relaxes a strong assumption and reduces the numbers of constraints and binary variables of the MIP problem. By conducting the numerical experiments, we show that the proposed approach is faster than the previous MIP approach.

Second, we propose a new approach to solve the minimum maximal flow (MMF) problem, which is one of the most influential applications of the OE problem. We give an upper bound for the optimal solution to this problem and prove this problem satisfies the assumption of our approach. Based on them, an equivalent simplified MIP problem of the MMF problem is proposed. Then a minimum maximal flow can be obtained with an MIP solver. By performing computational experiments, we find that: the previous MIP approach can solve 3% of the test problems while the proposed MIP approach can solve 100% of them; the proposed approach is mostly faster than the previous research; the growth rate of the running time of our approach is lower than the rates of previous works as the size of the instances increases; the proposed approach is efficient to the MMF problem even for relatively large instances, where the number of edges is up to 15,000.

Third, we show that the least distance problem (LDP) in data envelopment analysis (DEA) can be modeled as an equivalent OE problem, and this model can be optimized by applying the proposed MIP approach. In the implementation, we use a simplified equivalent MIP problem to solve this problem. Compared with the previous research, the proposed approach can obtain a closest efficient target over the whole efficient frontier instead of part of it under the variable returns-to-

scale assumption.

Keywords: Global Optimization, Multiobjective Programming, Efficient Set, Linear Complementarity Conditions, Mixed Integer Programming

Contents

Chapter 1

Introduction

The background is offered in the first section. The basic concepts are introduced in the second section. The previous research and our motivation are addressed in the third section. The structure of this thesis is shown in the fourth section.

1.1 Background

This section provides the background and the overview of our the optimization over the efficient set (OE problem), the minimum maximal flow (MMF) problem, and the least distance problem (LDP). Briefly, the overview of our research themes is shown in Figure 1.1.

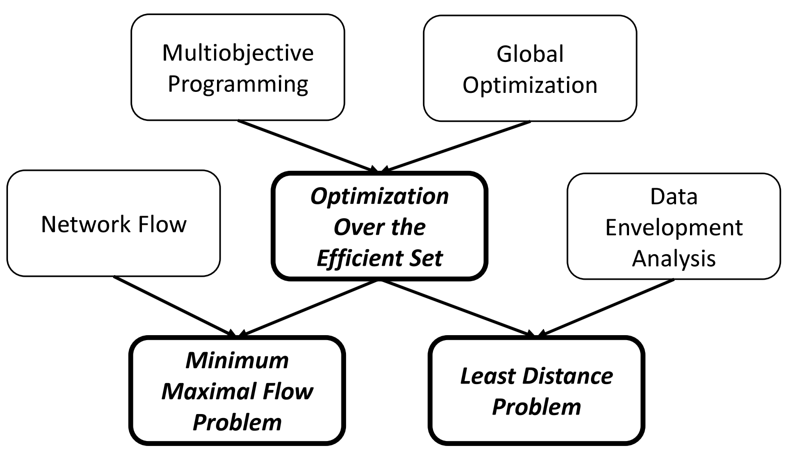


Figure 1.1: Research Overview

The OE problem is a global optimization over a nonconvex feasible region which is defined by the efficient set of a multiobjective programming problem. The background of the OE problem is displayed in the first subsection. Then we focus on two applications of the OE problem in the network flow problem and data envelopment analysis (DEA). There is a kind of network flow problem called the MMF problem which can be solved as an OE problem. We give a brief introduction to this problem in the second subsection[1]. In DEA, there is a problem called the LDP, which can be solved as an OE problem as well. The background of the LDP is introduced in the third subsection.

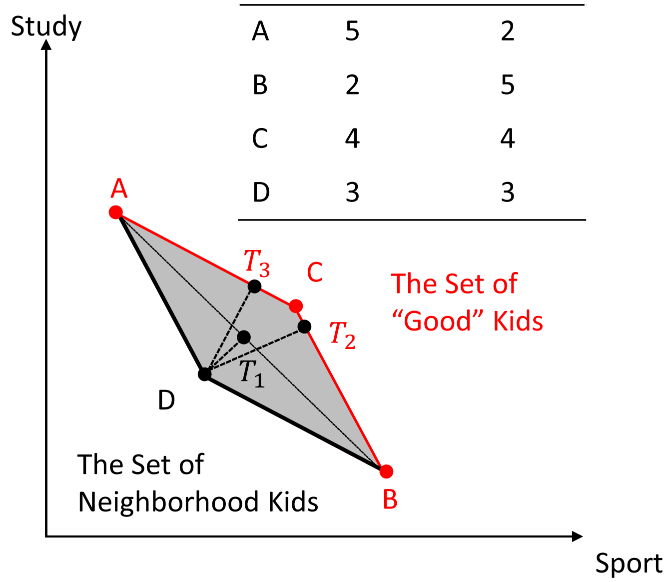
1.1.1 Multiobjective Programming and Optimization over the Efficient Set

In the real world, decision makings are inevitable and expected to be as good as possible, in other words, optimal. Usually, there are some objectives and constraints, and objectives should be optimized under these constraints to make an optimal decision. When there is only one objective, this is a classical optimization. However, the objectives that should be considered are usually not single but multi. For example, when choosing a hotel, the trade-off needs to be taken between the facility and the location under a certain budget. Staying at a luxury hotel in the rural area may have a golf course, but it may be quite far away from the nearest underground station. There are two kinds of methods in this kind of situation. Someone transforms the 'not-that-important' objectives into constraints, and optimizes the most important objective, while others choose to optimize all objectives simultaneously. In the latter method, which is called multiobjective programming, there are usually many solutions with trade-off relationships between the objectives. These solutions are called efficient solutions. Because of its practical significance, multiobjective programming has been widely applied in the real world, such as decision making in economics, the portfolio selection in finance, and the optimal design in engineering. There are many studies on multiobjective programming which focus on searching for an efficient solution. However, even we have a group of efficient solutions, the decision is still difficult to be made because these solutions cannot be compared simply. To choose the most appropriate solution, an additional complex function is usually needed, which means

[1]Chronologically, our MIP approach for the MMF problem in Lu et al. [32] is earlier than our MIP approach for the OE problem in Lu et al. [33, 34]. We generalized the approach for the MMF problem to solving the OE problem. In this thesis, we adjust the order to make our story clearer.

optimizing another objective over the set of efficient solutions, see Example 1.

Example 1 (How to be a good kid?)**.** *Consider a group of kids in the neighborhood in Figure 1.2.*



	Study	Sport
A	5	2
B	2	5
C	4	4
D	3	3

Figure 1.2: Good Kids

In this example, a possible performance (sport and study) is defined in the convex hull of all the known performances (ABCD). We consider a kid as good when there exists no possible performance that is better than his on both sport and study. Suppose D desire to be a good kid, all performances on the segment AC and CB are efficient and can be chosen as a target. The question is how to choose the best one from these infinite efficient solutions. In this case, the closest efficient solution may be the most appropriate choice for its practical advantages. The target can be sufficiently found by comparing $||DT_2||$, $||DT_3||$.

In optimization, this is called an optimization over the efficient set. However, this problem is hard to solve in general. In Example 1, if the number of good kids grows linearly, the computational complexity will grow exponentially. The difficulty comes from the nonconvexity of the set of efficient solutions.

1.1.2 Network Flow and Minimum Maximal Flow Problem

The network flow problem is an important basis of the modern society. It is widely used in transportation systems, electrical engineering, information systems, etc. Lying on the cusp between computer science and operations research, the network flow problem drew its roots back to electrical engineering in the 19th century. In 1950s, with the development of optimization research, the relationship between the network flow problem and linear programming was founded by Dantzig [16]. Over years of development, there are many works on this problem, see [1, 18] One of the most important problems in this field is the maximum flow problem. The maximum flow problem aims to find out how much flow we can send on a network with a capacity over each edge. Most of these works assume that all flows in the network can be controlled, so the maximum value is attainable in use. However, flows do not always satisfy this assumption in the real world. For example, we cannot control every vehicle in a transportation system in reality. So, the real-world case may conflict with the ideal result, see Example 2.

Example 2 (Evacuation Routes)**.** *Consider a city suffering a tsunami in Figure1.3, people on the beach are escaping to a mountain.*

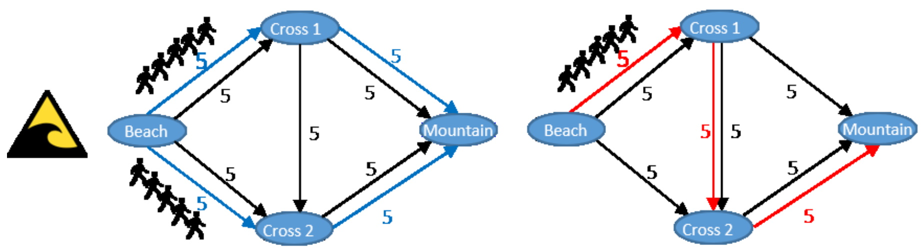


Figure 1.3: Evacuation Routes

The traditional research considers that people will all use the best routes, so 10 of them can be saved. However, if 5 of them use the route ‘Beach→Cross 1→ Cross 2→Mountain’, then the number of the saved people will decrease to 5.

It is meaningful to discuss what would happen without the controllable assumption, especially, what would happen in the worst cases. The MMF problem was proposed by Shi and Yamamoto [41], and lately, Shigeno et al. [42] showed that the MMF problem is a class of the OE problem. So improvements of the approaches to the OE problem can bring benefits to the MMF problem.

1.1.3 Data Envelopment Analysis and Least Distance Problem

There are many organizations or entities in the world, such as banks, supermarkets, car makers, hospitals, schools, and so on. Such organization or entity is called a decision making unit (DMU). A DMU converts several inputs into several outputs. Usually, the performance of one DMU is evaluated relative to others' performances. This kind of evaluation is called DEA. DEA has been widely used in management science to find out whether a DMU is relatively efficient and how to improve an inefficient DMU to be efficient. Most studies in this field focus on the previous part, and they leave the difficulty of the efficiency improvement unconsidered. Since the difficulty of this improvement depends on the difference between an inefficient DMU and the efficient target in general, their similarity should be considered, see Figure 1.4.

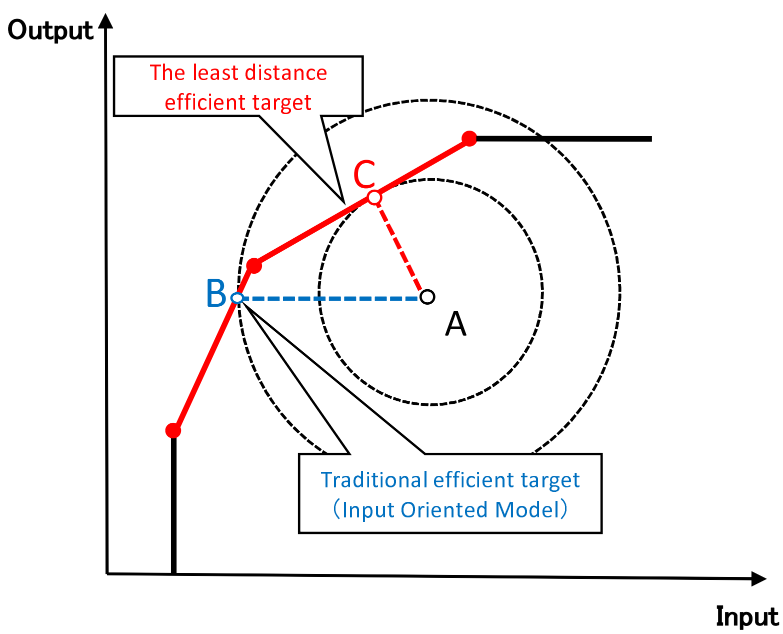


Figure 1.4: The Least Distance Efficient Target

Compared with the traditional efficient target B, the efficient target C is obviously more similar to the assessed DMU A, because C is closer when considering the Euclidean distance as the difference. It is important to offer an efficient target with the least distance for its practical merit. DEA is proposed by Charnes et al.

[13] in 1970s, and years later, Briec [12] introduced the concept of distance to measure the similarity. DEA is closely related to multiobjective programming.

1.2 Basic Concepts

In this section, we introduce the basic concepts of the optimization over the efficient set.

This thesis concerns the efficient set of a multiobjective linear programming (MOLP) problem, which is formulated as

$$\left|\begin{array}{ll} \text{minimize} & \boldsymbol{C}\boldsymbol{x} = \begin{pmatrix} \boldsymbol{c}_1^\top \boldsymbol{x} \\ \vdots \\ \boldsymbol{c}_p^\top \boldsymbol{x} \end{pmatrix} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} \leqq \boldsymbol{b}, \boldsymbol{x} \geqq \boldsymbol{0}, \end{array}\right. \tag{1.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of variables, $\boldsymbol{C} = (\boldsymbol{c}_1, \dots, \boldsymbol{c}_p)^\top \in \mathbb{R}^{p\times n}$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, and $\boldsymbol{b} \in \mathbb{R}^m$ are data, $\boldsymbol{0} \in \mathbb{R}^n$ is a vector of zeros, and p, m, and n are positive integers. Let

$$\boldsymbol{X} := \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{A}\boldsymbol{x} \leqq \boldsymbol{b}, \boldsymbol{x} \geqq \boldsymbol{0}\}$$

be the set of feasible solutions. An efficient solution to the MOLP problem (1.1) is defined as follows.

Definition 1 (Efficient Solution)**.** *An $\boldsymbol{x}' \in \boldsymbol{X}$ is called an efficient solution to (1.1) if there is no $\boldsymbol{x} \in \boldsymbol{X}$ such that $\boldsymbol{C}\boldsymbol{x} \leqq \boldsymbol{C}\boldsymbol{x}'$ and $\boldsymbol{C}\boldsymbol{x} \neq \boldsymbol{C}\boldsymbol{x}'$.*

An efficient solution is also called a Pareto optimal solution. Let $\boldsymbol{E}$ denote the set of all efficient solutions to the problem (1.1). Then the OE problem is defined as

$$\left|\begin{array}{ll} \text{minimize} & \phi(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in \boldsymbol{E}, \end{array}\right. \tag{1.2}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ is a convex objective function. To implement our approach with MIP solvers, the convexity (or linearity) of $\phi(\boldsymbol{x})$ is required. In fact, $\phi(\boldsymbol{x})$ is required to be linear or convex quadratic in Gurobi (using the MILP solver and the MIQP solver), and convex in 'fmincon' of Matlab. We note that the convexity of $\phi(\boldsymbol{x})$ is not necessary for our theoretical transformations.

1.3 Previous Research and Motivation

Since efficient sets are generally nonconvex, the OE problem plays an important role in theoretical research on multiobjective programming and global optimization. Since Philip [38] first considered the OE problem in 1972, there have been many works on this problem. There are two kinds of traditional algorithms, namely the decision space algorithms and the outcome space algorithms. The decision space algorithms were surveyed by Yamamoto [50], and they were classified into adjacent vertex search algorithms [11, 38], nonadjacent vertex search algorithms [8], branch-and-bound methods [7], Lagrangian relaxation methods [17], dual approaches [44], and bisection algorithms [39]. For the problem we consider here, the dimension of the outcome space is usually smaller than that of the decision space, so the outcome space algorithms have been proposed to accelerate the computation speed in [9, 10, 30]. Also, the OE problem can be transformed into a kind of difference-of-convex (DC) functions programming problem [24]. In fact, many of these researches are based on the DC optimization or its extensions as shown in Tuy [48]. On the other hand, the OE problem is also important for the DC optimization and has been called a milestone of the DC optimization by Hoai An and Tao [23].

Recently, Sun [43] proposed a new approach to the OE problem which transforms the OE problem into a mixed integer programming (MIP) problem under some assumptions of (1.1) and (1.2). The MIP has been well researched, see [28, 40]. It is promising for solving the OE problem efficiently by using current MIP solvers. However, since this approach needs a strong assumption, its accuracy is quite low for many application problems, see the following Example 3.

Example 3. *Here is a simple experiment with two approaches:*

- *Sun's MIP Approach [43]*
- *Muu and Thuy's DC Approach [37] (as a note)*

The result is shown in Table 1.1.

Table 1.1: Experiments on 10 random instances of the MMF problem

m	$n(=p)$	Sun [43](s)	# of instances (opt. sol. found)	DC [37](s)	# of instances (opt. sol. found)
80	160	**0.3786**	**0**	0.4601	8
100	200	**0.3808**	**0**	0.9351	8
200	400	**0.4556**	**0**	4.7539	8
400	800	**0.5833**	**0**	31.9423	7

The first and second columns are the number of the nodes and edges of tested instances, respectively. The third and fifth columns list the mean running time of 10 different instances in seconds of the two approaches, respectively. The fourth and sixth columns indicate the number of instances in which an exact optimal solution has been obtained in 10 tested instances.

Since state-of-the-art MIP solvers such as Gurobi is usually based on exact algorithms, such as branch and cut method, it is possible to provide solutions with higher accuracy than the DC approach if the strong assumption is relaxed. So in this thesis, we try to relax the strong assumption to improve the accuracy. Because of the advanced computation performance of these MIP solvers, this advantage is desired to be kept. In summary, we aim to develop an MIP approach without the strong assumption of Sun [43].

1.4 Structure

In Chapter 2, we propose a new equivalent MIP problem of the OE problem and compute an optimal solution with an MIP solver. In Chapter 3, by applying the result in Chapter 2, we solve the MMF problem with an MIP solver and show some advanced theoretical results. In Chapter 4, we show that the LDP can be modeled as an OE problem, and can be computed by using the result in Chapter 2. In Chapter 5, we conclude this thesis.

The main contents of this thesis are briefly summarized as follows.

Chapter 2: A Mixed Integer Programming Approach for Optimization over the Efficient Set

Chapter 2 concerns an equivalent MIP problem to compute an optimal solution to the original problem. Compared with the previous MIP approach by Sun, the proposed approach relaxes a strong assumption and reduces the numbers of constraints and binary variables of the MIP problem. By conducting the numerical experiments, we find that the proposed approach is more accurate and faster than the previous MIP approach. The proposed MIP problem can be efficiently solved with current state-of-the-art MIP solvers when the objective function is convex or linear. This chapter is related to [33, 34].

Chapter 3: Application I: the Minimum Maximal Flow Problem

Chapter 3 concerns a minimum maximal flow (MMF) problem, which finds a minimum maximal flow in a given network. This problem is known to be NP-hard. We show that the MMF problem can be formulated as an MIP problem by applying the method in Chapter 2, and we propose to find a minimum maximal flow by solving an equivalent MIP problem. By performing computational experiments, we observe that the proposed approach is efficient to the MMF problem even for relatively large instances, where the number of edges is up to 15,000. Also, the growth rate of the running time of our approach is lower than those of previous works as the size of the instances grows. This chapter is related to [32].

Chapter 4: Application I: the Least Distance Problem in Data Envelopment Analysis

In Chapter 4, we consider a method to solve the least distance problem (LDP) under the variable returns-to-scale (VRS) assumption in data envelopment analysis (DEA). This method aims to find a closest efficient target from the assessed decision making unit (DMU). DEA has been widely applied to evaluating the relative efficiency and providing some efficient targets as goals for the assessed DMU. Because the required modifications are minimized, a closest efficient target is usually more practical. Under this consideration, the least distance problem (LDP) has been actively studied. To find the closest target, we reformulate the LDP into an equivalent OE problem by defining the efficient frontier with a proposed auxiliary problem. The OE problem can be solved by an MIP solver. The closest efficient target can be obtained over the whole efficient frontier instead of part of the frontier under the VRS assumption with a theoretical guarantee. This chapter is related to [35].

Chapter 2

A Mixed Integer Programming Approach for the Optimization over the Efficient Set

In this chapter, we propose a new equivalent MIP problem of the OE problem and compute an optimal solution with the MIP solver. We show Sun's approach and our main result in Section 2.1. The proof of our approach is provided in Section 2.2. In Section 2.3, the numerical experiment result is shown to compare the computational performance. A brief summary and remarks are given in Section 2.4.

2.1 Mixed Integer Programming Approaches

In this section, after giving Sun's main result [43] on the problem (1.2), we show our result and make a comparison between these results.

Sun's result is obtained under following Assumptions 1 and 2 on the problems (1.1) and (1.2), respectively:

Assumption 1. *For the problem (1.1), there is an* $i \in \{1, 2, \ldots, p\}$ *such that* $\boldsymbol{c}_i^\top \boldsymbol{x}$ *is injective on* $\boldsymbol{X}$*, where* $\boldsymbol{c}_i$ *is the i-th row of the matrix* $\boldsymbol{C}$.

Assumption 2. *The problem (1.2) has an optimal solution* $\boldsymbol{x}^*$.

We note that Assumption 1 is very strong, because it is not satisfied if the dimension of the feasible region $\boldsymbol{X}$ is larger than 1. Hence, Sun's result is valid only when the dimension of the feasible region $\boldsymbol{X}$ is at most 1. If Assumption 1 fails, Sun's approach provides a weakly efficient solution[1] instead of an efficient solution. We give examples in Section 3.3 to show what would happen when this assumption fails. Assumption 2 is rather standard and used in most analyses of the problem (1.2). Please note that the problem (1.1) is feasible and the efficient set $\boldsymbol{E}$ is nonempty under Assumption 2.

We state Sun's main result [43] in the next proposition.

Proposition 1. *(Sun [43]) Under Assumptions 1 and 2, an optimal solution to the OE problem (1.2) is computed by solving the following MIP problem:*

$$\left|\begin{array}{ll} \text{minimize} & \phi(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{s} = \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{\beta} \end{pmatrix} - \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{C}_{(i)} \end{pmatrix} \boldsymbol{x}, \\ & \boldsymbol{r} = \boldsymbol{c}_i + \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{C}_{(i)} \end{pmatrix}^\top \boldsymbol{u}, \\ & \boldsymbol{\beta} = \boldsymbol{C}_{(i)}\boldsymbol{y}, \quad \boldsymbol{A}\boldsymbol{y} \leqq \boldsymbol{b}, \\ & \boldsymbol{u} \leqq \theta\boldsymbol{\alpha}_1, \quad \boldsymbol{s} \leqq \theta(\boldsymbol{1} - \boldsymbol{\alpha}_1), \\ & \boldsymbol{x} \leqq \theta\boldsymbol{\alpha}_2, \quad \boldsymbol{r} \leqq \theta(\boldsymbol{1} - \boldsymbol{\alpha}_2), \\ & \boldsymbol{\alpha}_1 \in \{0,1\}^{m+p-1}, \quad \boldsymbol{\alpha}_2 \in \{0,1\}^n, \\ & \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{s}, \boldsymbol{r} \geqq \boldsymbol{0}, \end{array}\right. \tag{2.1}$$

where $\boldsymbol{c}_i^\top \boldsymbol{x}$ is injective on $\boldsymbol{X}$, $\boldsymbol{C}_{(i)} = (\boldsymbol{c}_1^\top, ..., \boldsymbol{c}_{i-1}^\top, \boldsymbol{c}_{i+1}^\top, ..., \boldsymbol{c}_p^\top)^\top$, and $\theta > 0$ is a sufficiently large number.

We state our main result in the next theorem, and the proof is given in the next section.

Theorem 1. *Under Assumption 2, an optimal solution to the OE problem (1.2) is computed by solving the following MIP problem:*

$$\left|\begin{array}{ll} \text{minimize} & \phi(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{z} = \boldsymbol{b}, \\ & \boldsymbol{A}^\top\boldsymbol{u} + \boldsymbol{C}^\top\boldsymbol{v} - \boldsymbol{w} = -\boldsymbol{C}^\top\boldsymbol{1}, \\ & \boldsymbol{z} \leqq \theta\boldsymbol{\alpha}, \quad \boldsymbol{u} \leqq \theta(\boldsymbol{1} - \boldsymbol{\alpha}), \\ & \boldsymbol{x} \leqq \theta\boldsymbol{\beta}, \quad \boldsymbol{w} \leqq \theta(\boldsymbol{1} - \boldsymbol{\beta}), \\ & \boldsymbol{\alpha} \in \{0,1\}^m, \quad \boldsymbol{\beta} \in \{0,1\}^n, \\ & \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \geqq \boldsymbol{0}, \end{array}\right. \tag{2.2}$$

[1] An $\boldsymbol{x}' \in \boldsymbol{X}$ is called a weakly efficient solution to (1.1) if there is no $\boldsymbol{x} \in \boldsymbol{X}$ such that $\boldsymbol{C}\boldsymbol{x} < \boldsymbol{C}\boldsymbol{x}'$.

where θ is a sufficiently large number.

The MIP problem (2.2) can be efficiently solved by current state-of-the-art MIP solvers when the objective function is convex or linear.

Note that we do not require Assumption 1 in this theorem. Moreover, compared with the problem (2.1), the numbers of constraints and binary variables in (2.2) are less than those in (2.1). So we can solve the problem (2.2) much faster than (2.1) in practical computation, see the results of numerical experiments in Section 2.3.

2.2 Proof of the Main Theorem

In this section, we prove Theorem 1 stated in the previous section. At first, we refer to a linear version of a well-known result, Wendell and Lee's Theorem 1 [49], which gives a necessary and sufficient condition for an efficient solution $\boldsymbol{x} \in \boldsymbol{E}$.

Lemma 1. *$\boldsymbol{x}' \in \boldsymbol{X}$ is an efficient solution to the problem (1.1) if and only if $\boldsymbol{y} = \boldsymbol{x}'$ is an optimal solution to the following auxiliary problem:*

$$\left|\begin{array}{ll} \text{minimize} & (\boldsymbol{C}^\top \mathbf{1})^\top \boldsymbol{y} \\ \text{subject to} & -\boldsymbol{A}\boldsymbol{y} \geqq -\boldsymbol{b}, \\ & -\boldsymbol{C}\boldsymbol{y} \geqq -\boldsymbol{C}\boldsymbol{x}', \\ & \boldsymbol{y} \geqq \mathbf{0}, \end{array}\right. \tag{2.3}$$

where $\boldsymbol{y}$ is a vector of variables.

Since the problem (2.3) is a linear program, we obtain the following result by using the optimality conditions for linear programming (LP).

Lemma 2. *$\boldsymbol{x}' \in \mathbb{R}^n$ is an efficient solution to the problem (1.1) if and only if there exists $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$ such that*

$$\begin{array}{l} -\boldsymbol{A}\boldsymbol{x}' \geqq -\boldsymbol{b}, \\ \boldsymbol{A}^\top \boldsymbol{u} + \boldsymbol{C}^\top \boldsymbol{v} - \boldsymbol{w} = -\boldsymbol{C}^\top \mathbf{1}, \\ (-\boldsymbol{A}\boldsymbol{x}' + \boldsymbol{b})^\top \boldsymbol{u} = 0, \quad \boldsymbol{x}'^\top \boldsymbol{w} = 0, \\ \boldsymbol{x}', \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \geqq \mathbf{0}. \end{array} \tag{2.4}$$

Proof. The dual problem of the auxiliary problem (2.3) at $\boldsymbol{x}' \in \boldsymbol{X}$ is defined as

$$\left|\begin{array}{ll} \text{maximize} & -\boldsymbol{b}^\top \boldsymbol{u} - (\boldsymbol{C}\boldsymbol{x}')^\top \boldsymbol{v} \\ \text{subject to} & -\boldsymbol{A}^\top \boldsymbol{u} - \boldsymbol{C}^\top \boldsymbol{v} + \boldsymbol{w} = \boldsymbol{C}^\top \boldsymbol{1}, \\ & \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \geqq \boldsymbol{0}. \end{array}\right. \tag{2.5}$$

From the optimality conditions for LP, $\boldsymbol{y}$ and $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$ are optimal solutions to (2.3) and (2.5), respectively, if and only if the following conditions hold:

$$\begin{array}{l} -\boldsymbol{A}\boldsymbol{y} \geqq -\boldsymbol{b}, \\ -\boldsymbol{C}\boldsymbol{y} \geqq -\boldsymbol{C}\boldsymbol{x}', \\ \boldsymbol{A}^\top \boldsymbol{u} + \boldsymbol{C}^\top \boldsymbol{v} - \boldsymbol{w} = -\boldsymbol{C}^\top \boldsymbol{1}, \\ (-\boldsymbol{A}\boldsymbol{y} + \boldsymbol{b})^\top \boldsymbol{u} = 0, \quad (-\boldsymbol{C}\boldsymbol{y} + \boldsymbol{C}\boldsymbol{x}')^\top \boldsymbol{v} = 0, \quad \boldsymbol{y}^\top \boldsymbol{w} = 0, \\ \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \geqq \boldsymbol{0}. \end{array}$$

From Lemma 1, $\boldsymbol{x}' \in \boldsymbol{X}$ is an optimal solution to the problem (1.1) if and only if $\boldsymbol{y} = \boldsymbol{x}'$ satisfies the above conditions, that is,

$$\begin{array}{l} -\boldsymbol{A}\boldsymbol{x}' \geqq -\boldsymbol{b}, \\ \boldsymbol{A}^\top \boldsymbol{u} + \boldsymbol{C}^\top \boldsymbol{v} - \boldsymbol{w} = -\boldsymbol{C}^\top \boldsymbol{1}, \\ (-\boldsymbol{A}\boldsymbol{x}' + \boldsymbol{b})^\top \boldsymbol{u} = 0, \quad \boldsymbol{x}'^\top \boldsymbol{w} = 0, \\ \boldsymbol{x}', \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \geqq \boldsymbol{0}. \end{array}$$

Hence, we obtain the result of this lemma. □

From Lemma 1, the OE problem (1.2) is equivalent to the following problem:

$$\left|\begin{array}{ll} \text{minimize} & \phi(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{z} = \boldsymbol{b}, \\ & \boldsymbol{A}^\top \boldsymbol{u} + \boldsymbol{C}^\top \boldsymbol{v} - \boldsymbol{w} = -\boldsymbol{C}^\top \boldsymbol{1}, \\ & \boldsymbol{z}^\top \boldsymbol{u} = 0, \quad \boldsymbol{x}^\top \boldsymbol{w} = 0, \\ & \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \geqq \boldsymbol{0}. \end{array}\right. \tag{2.6}$$

The problem (2.6) is a mathematical program with linear complementarity constraints (MPCC). However, the MPCC is not easy to solve in general because the constraints $\boldsymbol{z}^\top \boldsymbol{u} = 0$ and $\boldsymbol{x}^\top \boldsymbol{w} = 0$ are nonconvex. There are many works solving the MPCC as a nonconvex continuous programming, see [15, 25, 36]. An interior point method for a related problem is also proposed in the author's master's thesis [31]. However, these methods may not always be able to solve the MPCC problem. On the other hand, an MIP solver is promising to solve the MPCC problem by transforming it into an equivalent MIP problem. The previous MIP approach by Sun [43] can only solve limited problems, since the strong Assumption 1 is necessary when proving its Theorem 2.3. So the largest difficulty for applying an MIP solver is how to relax Assumption 1. We use the property of the problem (1.1) and establish the problem (2.6) with Lemmas 1 and 2. As a result of this approach, Assumption 1 is not required.

Next, we show that the problem (2.6) is equivalent to an MIP problem. Since the problem (1.2) has an optimal solution from Assumption 2, the problem (2.6) also has an optimal solution. To circumvent the difficulties when solving the problem (2.6), we make the following assumption.

Assumption 3. *θ is sufficiently large such that*

$$\|(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{w}^*)\|_\infty \leqq \theta$$

for an optimal solution $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*)$ to the problem (2.6).

We can obtain the result of Theorem 1 from the next lemma.

Lemma 3. *Under Assumption 3, the problem (2.6) is equivalent to the following MIP problem:*

$$\left|\begin{array}{ll} \text{minimize} & \phi(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{z} = \boldsymbol{b}, \\ & \boldsymbol{A}^\top \boldsymbol{u} + \boldsymbol{C}^\top \boldsymbol{v} - \boldsymbol{w} = -\boldsymbol{C}^\top \boldsymbol{1}, \\ & \boldsymbol{z} \leqq \theta\boldsymbol{\alpha}, \quad \boldsymbol{u} \leqq \theta(\boldsymbol{1} - \boldsymbol{\alpha}), \\ & \boldsymbol{x} \leqq \theta\boldsymbol{\beta}, \quad \boldsymbol{w} \leqq \theta(\boldsymbol{1} - \boldsymbol{\beta}), \\ & \boldsymbol{\alpha} \in \{0,1\}^m, \quad \boldsymbol{\beta} \in \{0,1\}^n, \\ & \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \geqq \boldsymbol{0}. \end{array}\right. \tag{2.7}$$

Proof. Let $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*)$ be an optimal solution to the problem (2.6). We define

$$I^* = \{i \mid u_i^* = 0\}, \quad J^* = \{j \mid w_j^* = 0\}.$$

Then $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*)$ is an optimal solution to the following problem for $I = I^*$ and $J = J^*$:

$$\left|\begin{array}{ll} \text{minimize} & \phi(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{z} = \boldsymbol{b}, \\ & \boldsymbol{A}^\top \boldsymbol{u} + \boldsymbol{C}^\top \boldsymbol{v} - \boldsymbol{w} = -\boldsymbol{C}^\top \boldsymbol{1}, \\ & z_i = 0, \ i \notin I, \\ & u_i = 0, \ i \in I, \\ & x_j = 0, \ j \notin J, \\ & w_j = 0, \ j \in J, \\ & \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \geqq \boldsymbol{0}. \end{array}\right. \tag{2.8}$$

Let $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$ be a feasible solution to (2.8) for any $I \subset \{1, 2, \ldots, m\}$ and $J \subset \{1, 2, \ldots, n\}$. Then $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$ is a feasible solution to (2.6), so we have that

$$\phi(\boldsymbol{x}^*) \leqq \phi(\boldsymbol{x}).$$

Hence, $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*)$ is a minimum solution to all the problems (2.8) for every $I \subset \{1, 2, \ldots, m\}$ and $J \subset \{1, 2, \ldots, n\}$. Define $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ by

$$\alpha_i^* = \begin{cases} 0, & i \notin I^*, \\ 1, & i \in I^*, \end{cases} \qquad \beta_j^* = \begin{cases} 0, & j \notin J^*, \\ 1, & j \in J^*. \end{cases}$$

Then, from Assumption 3, $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ is an optimal solution to the problem (2.7).

Conversely, suppose that $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ is an optimal solution to the problem (2.7). Then it is not hard to show that $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*)$ is an optimal solution to the problem (2.6). □

Note that the method in [43] uses a similar MIP transformation as well, therefore Sun's result also requires Assumption 3. So the major difference between these two approaches on the assumptions is that the proposed approach does not require Assumption 1.

2.3 Numerical Experiments

In this section, we design a simple numerical experiment to compare the performances of the two approaches on instances that satisfy Assumption 1.

We use INTLINPROG of Gurobi 8.00 and Matlab 2017a to solve instances of the problem on a Windows 10 PC with a 3.20GHz Intel Core i5 6500 CPU and 32GB memory. θ is set to 10^6 in experiments.

The first experiment is based on the following MOLP problem

$$\left| \begin{array}{ll} \text{minimize} & \boldsymbol{C}\boldsymbol{x} \\ \text{subject to} & \begin{pmatrix} \boldsymbol{A} \\ -\boldsymbol{A} \end{pmatrix} \boldsymbol{x} \leqq \begin{pmatrix} \boldsymbol{b} \\ -\boldsymbol{b} \end{pmatrix}, \boldsymbol{0} \leqq \boldsymbol{x} \leqq \theta \boldsymbol{1}, \end{array} \right. \tag{2.9}$$

where $\boldsymbol{A} \in \mathbb{R}^{(n-1)\times n}$ has full row rank, $\boldsymbol{b} \in \mathbb{R}^n$, and $\boldsymbol{C} \in \mathbb{R}^{p\times n}$. To construct a nonempty feasible region, we set $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_0$ where $\boldsymbol{x}_0 \in \mathbb{R}^n$ is a random positive vector. We construct the following OE problem

$$\left| \begin{array}{ll} \text{minimize} & \boldsymbol{d}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{x} \text{ is an efficient solution to (2.9),} \end{array} \right. \tag{2.10}$$

where $\boldsymbol{d} \in \mathbb{R}^p$. All these coefficients and parameters are rational and randomly generated from $[-10, 10]$ with uniform distribution. So Assumption 1 is satisfied with probability 1, and Assumption 2 is constantly satisfied. Since the efficient set of (2.9) is at most one dimensional, it is easy to compute and check the optimal solution to (2.10).

The results of computational experiments are shown in Table 2.1. The first two columns show the sizes of p and n, respectively. The third and the fifth columns list the mean running time of 10 different instances in seconds of the two approaches, respectively. The fourth and sixth columns indicate the number of instances in which an exact optimal solution has been obtained.

Table 2.1: Experiments with Assumption 1

p	$n(= 0.5m)$	Our(s)	# of instances (opt. sol. found)	Sun [43](s)	# of instances (opt. sol. found)
3	5	0.3750	10	0.4126	10
3	10	0.4756	10	0.4967	10
3	50	0.4698	10	0.4923	10
3	100	0.6234	10	0.8375	10
3	500	19.9102	10	36.3735	10
100	5	0.3693	10	0.3795	10
100	10	0.4416	10	0.5053	10
100	50	0.4106	10	0.5427	10
100	100	0.5486	10	1.7158	10
100	500	16.7907	10	39.3602	10
1,000	10	0.4431	10	0.8928	10
1,000	50	0.5463	10	2.5072	10
1,000	100	0.6383	10	4.6795	10
1,000	500	20.4722	10	141.7453	10

We observe from Table 2.1 that both approaches solved all instances while the proposed approach uses less running time with the growth of the problem size. In the largest cases where $n = 500$, it runs 2 to 7 times faster than the previous approach.

2.4 Brief Summary and Remarks

In this section, we summarize this chapter briefly, and give remarks about Assumption 1 and θ.

Brief Summary

In this chapter, we showed that an optimal solution to the optimization problem (1.2) over the efficient set of the multiobjective linear programming problem (1.1) can be computed by solving the MIP problem (2.2). Compared with the previous MIP approach by Sun [43], our approach does not require Assumption 1 on the problem (1.1) and reduces the size of the MIP problem. By conducting the preliminary computational experiments, we observed that our approach can solve the OE problem faster. Also, the proposed approach provides an easier implementation to solve the applications of the OE problem that arise from the real world.

Remarks

- **Assumption 1**
  As mentioned before, Assumption 1 is very strong, and necessary for proving Sun's approach. If Assumption 1 fails, Sun's approach provides a weakly efficient solution[2] instead of an efficient solution. We show examples in Table 3.2 of Section 3.3, and show what could happen when this assumption fails.

- **How to estimate $\boldsymbol{\theta}$**
  Both Sun's approach and ours need the value of θ in Assumption 3 in advance. This is a common limitation of both approaches. It is generally not easy to get a good estimate value of θ except for special cases. Lu et al. [32] obtain a good upper bound of θ in the case of the MMF problem. In the case of linear optimization, that is, $\phi(x)$ is linear, it is known that $\theta = 2^L$ is sufficient to satisfy Assumption 3 (Khachiyan [29]), since the optimal solution is a basic feasible solution of (2.8), where L is the sum of the bit lengths of all the data in the problem (2.8). In fact, let $(x^*, z^*, u^*, v^*, w^*)$ be the optimal solution of (2.6) when $\phi(x)$ is linear. Then $(x^*, z^*, u^*, v^*, w^*)$ is the optimal solution of the linear programming problem (2.8) for $I = I^*$ and $J = J^*$.

[2] An $\boldsymbol{x}' \in \boldsymbol{X}$ is called a weakly efficient solution to (1.1) if there is no $\boldsymbol{x} \in \boldsymbol{X}$ such that $\boldsymbol{Cx} < \boldsymbol{Cx}'$.

We can assume that $(x^*, z^*, u^*, v^*, w^*)$ is a basic feasible solution of (2.8). Hence, we have that

$$\|(x^*, z^*, u^*, v^*, w^*)\| \leqq 2^L.$$

See Khachiyan [29] for the detail of this inequality.

Chapter 3

Application I: the Minimum Maximal Flow Problem

In this chapter, we focus on applying our MIP approach to solve the MMF problem. In Section 3.1, the basic concepts and previous researches are introduced. Then we propose an MIP approach for the MMF problem in Section 3.2. In Section 3.3, the numerical experiments are made to check the efficiency of our approach. The brief summary is given in Section 3.4.

3.1 Introduction

The MMF problem is introduced by Shi and Yamamoto [41] in 1997 and it is known to be NP-hard. Let $\boldsymbol{N} = (\boldsymbol{V}, \boldsymbol{E}, s, t, \boldsymbol{c})$ be a network, where $\boldsymbol{V}$ is a set of $m+2$ nodes v_j $(j \in \{1, 2, \ldots, m+2\})$, $\boldsymbol{E}$ is a set of n edges e_k $(k \in \{1, 2, \ldots, n\})$, s is a source, t is a sink, and $\boldsymbol{c} \in \mathbb{R}^n$ is a finite positive vector whose k-th element c_k denotes the capacity of k-th edge e_k. In this paper, $\boldsymbol{V}$ includes both s and t $(s = v_{m+1}$ and $t = v_{m+2})$. Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be the node-edge incidence matrix of the network $\boldsymbol{N}$, that is, each entry a_{jk} $(j \in \{1, 2, \ldots, m\}$ and $k \in \{1, 2, \ldots, n\})$ is defined as

$$a_{jk} = \begin{cases} 1 & \text{if edge } e_k \text{ leaves node } v_j, \\ -1 & \text{if edge } e_k \text{ enters node } v_j, \\ 0 & \text{otherwise.} \end{cases}$$

Then the set $\boldsymbol{X}$ of feasible flows is given by

$$\boldsymbol{X} = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c}\},$$

where $\mathbf{0}$ is a vector whose each element is 0. The value of a flow $\boldsymbol{x}$ is computed by $\boldsymbol{d}^\top \boldsymbol{x}$, where the k-th element of $\boldsymbol{d} \in \mathbb{R}^n$ is

$$d_k = \begin{cases} 1 & \text{if edge } e_k \text{ leaves source } s, \\ -1 & \text{if edge } e_k \text{ enters source } s, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2 (Maximal Flow)**.** *A feasible flow $\boldsymbol{x} \in \boldsymbol{X}$ is maximal if there are no feasible flows $\boldsymbol{y} \in \boldsymbol{X}$ such that $\boldsymbol{y} \geqq \boldsymbol{x}$ and $\boldsymbol{y} \neq \boldsymbol{x}$.*

The minimum maximal flow (MMF) problem is denoted as

$$\left| \begin{array}{ll} \text{minimize} & \boldsymbol{d}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{x} \in \boldsymbol{X} \text{ is a maximal flow.} \end{array} \right. \tag{3.1}$$

To our knowledge, there is no concrete description of this problem as an MIP problem.

In traditional network flow problems, there are many excellent works, see [1, 18]. These works are usually assuming that all flows can be controlled in the network. Then the maximum flow value is attainable. However, a flow may not always satisfy this assumption in the real world. The minimum of the maximal flow values plays an important role in finding what happens when we cannot control the flow. The MMF problem is closely related to an uncontrollable flow raised by Iri [26].

Shi and Yamamoto [41] proposed a global optimization algorithm for the MMF problem. The algorithm increases the lower bound of the flow on each edge and uses a global search method. In 2003, Shigeno et al. [42] showed that the MMF problem is a class of the OE problem. Then algorithms for the OE problem have been applied to solve the MMF problem [22, 42]. In 2007, Yamamoto and Zenke [51] express the set of the maximal flows as a DC set and propose an algorithm that solves the problem with an outer approximation method and a local search method based on DC optimization theory [24]. In 2009, Chen et al. [14, 27] use non-homogeneous Farkas' Theorem to define the set of maximal flows and propose an algorithm. In 2014, Muu and Thuy [37] use a smoothing technique on the DC algorithm to solve the problem locally. In those algorithms, the computational time highly depends on the size of n. Some papers mentioned above report the results of computational experiments for solving the MMF problems, but the size of n is at most 200.

3.2 The Mixed Integer Programming Approach

In this section, we show that the minimum maximal flow of the network $\boldsymbol{N}$ can be computed by solving an MIP problem. More precisely, we start with the next two lemmas to show that the MMF problem is a class of the OE problem by Shigeno et al. [42].

Lemma 4. *A vector $\boldsymbol{x} \in \mathbb{R}^n$ is the maximal flow of the network $\boldsymbol{N} = (\boldsymbol{V}, \boldsymbol{E}, s, t, \boldsymbol{c})$ if and only if $\boldsymbol{x}$ is an efficient solution to the following MOLP problem (3.2):*

$$\left|\begin{array}{ll} \text{maximize} & \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \\ & \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c}. \end{array}\right. \tag{3.2}$$

Since Definitions 1 and 2, it is trivial that a maximal flow is an efficient solution to the problem (3.2). Then we have the next lemma.

Lemma 5. *The MMF problem is equivalent to the following OE problem (3.3):*

$$\left|\begin{array}{ll} \text{minimize} & \boldsymbol{d}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{x} \textit{ is an efficient solution to (3.2).} \end{array}\right. \tag{3.3}$$

To apply our MIP approach, we need to prove that the problem (3.3) satisfies Assumption 2 and 3 firstly. Since there exists a feasible flow whose elements are all zero, and the capacity is finite, the feasible region is bounded and nonempty. So Assumption 2 is satisfied. To make the proof more clear, we offer an MIP approach of the MMF problem in Theorem 2 first, then show the upper bound of variables in Lemma 6 which is related to Assumption 3.

Theorem 2. *A vector $\boldsymbol{x} \in \mathbb{R}^n$ is the minimum maximal flow of the network $\boldsymbol{N} = (\boldsymbol{V}, \boldsymbol{E}, s, t, \boldsymbol{c})$ if and only if there exist $\boldsymbol{z} \in \mathbb{R}^m$, $\boldsymbol{w} \in \mathbb{R}^n$, and $\boldsymbol{\alpha} \in \{0, 1\}^n$ such that $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\alpha})$ is an optimal solution to the following MIP problem:*

$$\left|\begin{array}{ll} \text{minimize} & \boldsymbol{d}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \\ & \boldsymbol{A}^\top \boldsymbol{u} + \boldsymbol{w} = \boldsymbol{1}, \\ & \boldsymbol{c} - \boldsymbol{x} \leqq c_{max}\boldsymbol{\alpha}, \\ & \boldsymbol{w} \leqq n(\boldsymbol{1} - \boldsymbol{\alpha}), \\ & \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c}, \\ & \boldsymbol{\alpha} \in \{0, 1\}^n, \end{array}\right. \tag{3.4}$$

where

$$c_{max} = \max\{c_k \mid k \in \{1, 2, \ldots, n\}\}.$$

Proof. By Lemma 5, similarly with the proof of Lemma 2, we have that a flow $\boldsymbol{x} \in \boldsymbol{X}$ is a maximal flow if and only if there exist $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{z} \in \mathbb{R}^m$, $\boldsymbol{u} \in \mathbb{R}^n$, $\boldsymbol{v} \in \mathbb{R}^n$, and $\boldsymbol{t} \in \mathbb{R}^n$ such that

$$\begin{aligned}
&\boldsymbol{A}\boldsymbol{y} = \boldsymbol{0},\ \boldsymbol{x} \leqq \boldsymbol{y},\ \boldsymbol{0} \leqq \boldsymbol{y} \leqq \boldsymbol{c},\\
&\boldsymbol{A}^\top \boldsymbol{z} + \boldsymbol{u} - \boldsymbol{v} - \boldsymbol{t} = \boldsymbol{1},\\
&(\boldsymbol{c} - \boldsymbol{y})^\top \boldsymbol{u} = 0,\ (\boldsymbol{y} - \boldsymbol{x})^\top \boldsymbol{v} = 0,\ (\boldsymbol{y} - \boldsymbol{0})^\top \boldsymbol{t} = 0,\\
&\boldsymbol{u}, \boldsymbol{v}m \geqq \boldsymbol{0}.
\end{aligned}$$

Since $\boldsymbol{x} \geqq \boldsymbol{0}$, we can relax $\boldsymbol{y} \geqq \boldsymbol{0}$, and simplify the system by deleting $\boldsymbol{t}$. Hence $\boldsymbol{y} = \boldsymbol{x}$, that is, a flow $\boldsymbol{x} \in \boldsymbol{X}$ is maximal if and only if there exist $\boldsymbol{z}$, $\boldsymbol{u}$, and $\boldsymbol{v}$ such that

$$\begin{aligned}
&\boldsymbol{A}^\top \boldsymbol{z} + \boldsymbol{u} - \boldsymbol{v} = \boldsymbol{1},\\
&(\boldsymbol{c} - \boldsymbol{x})^\top \boldsymbol{u} = 0,\\
&\boldsymbol{u}, \boldsymbol{v} \geqq \boldsymbol{0}.
\end{aligned}$$

So the MMF problem (3.1) is equivalent to the following MPCC problem:

$$\left|\begin{array}{ll}
\text{minimize} & \boldsymbol{d}^\top \boldsymbol{x}\\
\text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0},\\
& \boldsymbol{A}^\top \boldsymbol{z} + \boldsymbol{u} - \boldsymbol{v} = \boldsymbol{1},\\
& (\boldsymbol{c} - \boldsymbol{x})^\top \boldsymbol{u} = 0,\\
& \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c},\\
& \boldsymbol{u}, \boldsymbol{v} \geqq \boldsymbol{0}.
\end{array}\right. \tag{3.5}$$

Solving the problem (3.7) is equivalent to find the minimum solution to all the problems (3.7) for $I \subset \{1, 2, \ldots, n\}$, see the similar proof of Lemma 3. By Lemma 6 below, the optimal solution to the problem (3.7) $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$ is upper bounded by n, and so the optimal solution to the problem (3.5) is. So Assumption 3 is satisfied. Then applying the previous MIP approach, solving the MMF problem is equivalent to solve the following MIP problem:

$$\left|\begin{array}{ll}
\text{minimize} & \boldsymbol{d}^\top \boldsymbol{x}\\
\text{subject to} & \boldsymbol{A}^\top \boldsymbol{z} + \boldsymbol{u} - \boldsymbol{v} = \boldsymbol{1},\\
& \boldsymbol{c} - \boldsymbol{x} \leqq c_{max}\boldsymbol{\alpha},\\
& \boldsymbol{u} \leqq n(\boldsymbol{1} - \boldsymbol{\alpha}),\\
& \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0},\\
& \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c},\\
& \boldsymbol{u}, \boldsymbol{v} \geqq \boldsymbol{0},\\
& \boldsymbol{\alpha} \in \{0, 1\}^n.
\end{array}\right. \tag{3.6}$$

Now we show that the problem (3.6) is equivalent to the problem (3.4) in the sense that, for any feasible solution to one problem, we can get a feasible solution to another problem, where their objective function values are the same. For any $\boldsymbol{u} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^n$, define $\boldsymbol{w}(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^n$ by

$$\boldsymbol{w}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u} - \boldsymbol{v}.$$

Then for any feasible solution $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\alpha})$ to (3.6), $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w}(\boldsymbol{u}, \boldsymbol{v}), \boldsymbol{\alpha})$ is a feasible solution to (3.4) and their objective function values are the same. For any $\boldsymbol{w} \in \mathbb{R}^n$, define $\boldsymbol{u}(\boldsymbol{w}) \in \mathbb{R}^n$ and $\boldsymbol{v}(\boldsymbol{w}) \in \mathbb{R}^n$ by

$$\begin{array}{rcll} u_k(\boldsymbol{w}) & = & \max\{w_k, 0\}, & k \in \{1, 2, \ldots, n\}, \\ v_k(\boldsymbol{w}) & = & \max\{-w_k, 0\}, & k \in \{1, 2, \ldots, n\}. \end{array}$$

Then for any feasible solution $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\alpha})$ to (3.4), $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}(\boldsymbol{w}), \boldsymbol{v}(\boldsymbol{w}), \boldsymbol{\alpha})$ is a feasible solution to the problem (3.6) and their objective function values are the same.

Since $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{\alpha}^*)$ is an optimal solution to (3.6), $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{w}(\boldsymbol{u}^*, \boldsymbol{v}^*), \boldsymbol{\alpha}^*)$ is an optimal solution to the problem (3.4), that is, there exist $\boldsymbol{z} \in \mathbb{R}^m$, $\boldsymbol{w} \in \mathbb{R}^n$, and $\boldsymbol{\alpha} \in \{0, 1\}^n$ such that $(\boldsymbol{x}^*, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\alpha})$ is an optimal solution to (3.4).

Conversely, suppose that $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{w}^*, \boldsymbol{\alpha}^*)$ is an optimal solution to the problem (3.4). From the discussion above, $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}(\boldsymbol{w}^*), \boldsymbol{v}(\boldsymbol{w}^*), \boldsymbol{\alpha}^*)$ is an optimal solution to the problem (3.6). Hence $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}(\boldsymbol{w}^*), \boldsymbol{v}(\boldsymbol{w}^*))$ is an optimal solution to (3.5) and $\boldsymbol{x}^*$ is a minimum maximal flow of the network $\boldsymbol{N}$. □

Lemma 6. *Fix any $I \subset \{1, 2, \ldots, n\}$. If the linear programming problem (3.7):*

$$\left| \begin{array}{ll} \textit{minimize} & \boldsymbol{d}^\top \boldsymbol{x} \\ \textit{subject to} & \boldsymbol{A}^\top \boldsymbol{z} + \boldsymbol{u} - \boldsymbol{v} = \boldsymbol{1}, \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \\ & \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c}, \\ & x_k = c_k, \quad k \in I, \\ & u_k = 0, \quad k \notin I, \\ & \boldsymbol{u}, \boldsymbol{v} \geqq \boldsymbol{0}. \end{array} \right. \tag{3.7}$$

has an optimal solution, then there exists an optimal solution $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$ which satisfies

$$\|(\boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)\|_\infty \leqq n.$$

Proof. Suppose that the linear programming problem (3.7) has an optimal solution. From the fundamental theorem of linear programming, it has a basic optimal solution $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$. So $(\boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$ is a basic solution to the following linear system of equations:

$$\begin{aligned} &\boldsymbol{A}^\top \boldsymbol{z} + \boldsymbol{u} - \boldsymbol{v} = \mathbf{1}, \\ &u_k = 0, \quad k \notin \boldsymbol{I}. \end{aligned}$$

Since the coefficient matrix of the system above is totally unimodular, we obtain from Cramer's rule that

$$\|(\boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)\|_\infty \leqq \|\mathbf{1}\|_1 = n.$$

□

With Theorem 2, the MMF problem can be solved as an MIP problem by a solver. In this problem, Assumption 1 is not satisfied in general, otherwise there exists an edge $k \in \{1, \ldots, n\}$ and x_k can uniquely determine the flow $\boldsymbol{x}$.

3.3 Numerical Experiments

In this section, we report some results of preliminary computational experiments for finding a minimum maximal flow of the network $\boldsymbol{N}$ by solving the MIP problem (3.4). We use INTLINPROG of Gurobi 8.00 and Matlab 2017a to solve instances of the problem. The OS is Windows 10, CPU is 3.20GHz Intel Core i5-6500, and the memory is 32GB.

We use two kinds of MMF problems. First, we construct networks whose minimum maximal flow values are known, so that we can check whether the exact optimal solutions are computed and compare the accuracy with Muu and Thuy [37] and Sun [43]. Then we use the MMF problems presented in Shigeno et al. [42], Muu and Thuy [37], so that we can compare our approach's running time with theirs. We also conduct experiments on relatively large instances.

Specially Structured Networks

Since the MMF problem is NP-hard, it is usually difficult to find an exact optimal solution. To check whether we find an optimal solution, we construct specially structured networks whose optimal solutions can be computed easily. We use networks of the following structure. The network has n edges $\{e_1, e_2, \ldots, e_n\}$ and $m + 2$ nodes $\{v_1, v_2, \ldots, v_m, s, t\}$. First three edges are generated as (v_1, s), (t, v_m), (s, t), and the other $n - 3$ edges are randomly generated by choosing 2 different nodes from $\boldsymbol{V}/\{s, t\}$ for each edge. The capacity c_k of each edge is randomly

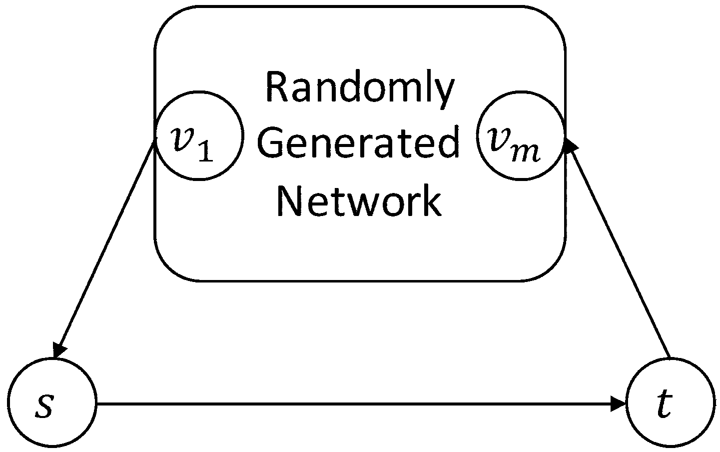


Figure 3.1: A Specially Structured Network

chosen from $\{1, 2, ..., 10\}$. See Figure 3.1 for the illustration of this network. The minimum maximal flow value of this network is equal to the maximum flow value from s to t minus the maximum flow value from t to s.

The results of computational experiments are shown in Table 3.1. The first column and the second column show the sizes of m and n, the third column shows the mean running time of 500 different instances in seconds, the fourth column shows the standard deviation of the running time, and the fifth column shows the rate of instances for which we find the exact optimal solution. We can see from

Table 3.1: Experiments on specially structured networks

m	$n(=p)$	Running time(s)	Standard Deviation	Accuracy(%)
10	20	0.0901	0.0154	100
20	40	0.0997	0.0172	100
40	80	0.1121	0.0147	100
80	160	0.1188	0.0136	100
100	200	0.1192	0.0127	100
200	400	0.1352	0.0136	100
400	800	0.1923	0.0207	100

Table 3.1 that the proposed approach find an optimal solution in a short running time.

As mentioned before, Assumption 1 is not satisfied in the MMF problem in general. To find what happens in this case, we conduct the following experiments. [1]

The results of computational experiments are shown in Table 3.2. The first and the second column show the sizes of m and n, respectively. The third, the fifth, and the seventh column show the mean running time of 10 different instances in seconds of the two approaches, respectively. The fourth, the sixth, and the eighth column show the number of instances in which an exact optimal solution has been obtained.

Table 3.2: Experiments without Assumption 1

m	$n(=p)$	Our(s)	# of instances (opt. sol. found)	Sun [43](s)	# of instances (opt. sol. found)	DC [37](s)	# of instances (opt. sol. found)
10	20	0.2467	10	0.3038	2	0.0663	10
20	40	0.3099	10	0.3363	0	0.0774	9
40	80	0.3874	10	0.4037	0	0.1994	8
80	160	0.3964	10	0.3786	0	0.4601	8
100	200	0.3844	10	0.3808	0	0.9351	8
200	400	0.3846	10	0.4556	0	4.7539	8
400	800	0.4808	10	0.5833	0	31.9423	7

In this experiment, our approach obtained the exact optimal solution in all tested cases. However, Sun's approach only solved 2 instances with an exact optimal solution in all 70 tested instances, see Table 3.2, though most of the obtained solutions are weakly efficient. Sun's approach almost failed to find any exact optimal solution in this application because the MMF problem does not satisfy Assumption 1 in general. Since Assumption 1 is too strong, the application of Sun's approach is limited. Moreover, the running times of our approach are usually less than those of Sun's.

[1] To implement Sun's approach, we use a random $i \in \{1, \ldots, n\}$ and $\theta = 10^6$.
To the fairness of this experiment, we set $\theta = 10^6$ in our approach instead of an exact θ which are different from the experiment in Table 3.1.
Here we use a DC approach based on Muu and Thuy [37] as a note and rewrite the code of it to deal with the relatively large instances. Since the difference of the programming technique, for fairness, the experiments use the original experiment data when comparing the running time.

Comparison with Previous Works

We use a totally randomly generated networks presented in Shigeno et al. [42]. The network has n edges $\{e_1, e_2, \ldots, e_n\}$ and $m+2$ nodes $\{v_1, v_2, \ldots, v_m, s, t\}$. Each edge is randomly generated by choosing 2 different nodes. The capacity c_k of each edge is randomly chosen from $\{1, 2, ..., 10\}$. See Figure 3.2 for the illustration of this network.

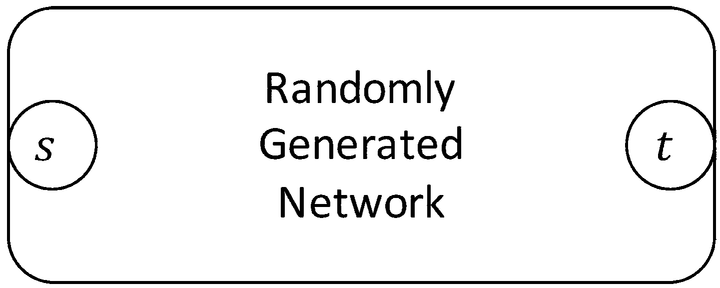


Figure 3.2: A Generally Structured Network

The results of computational experiments are shown in Table 3.3. The first 3 columns are the same as Table 3.1, and the fourth column shows the running time in [42]. In this table, the sizes of generated networks are based on previous work's experiments [42]. In Figure 3.3, we show the natural logarithm of the running time by Shigeno et al.'s algorithm and ours. From the figure, our running times are much less than Shigeno et al.'s in all tested instances. By regression analysis, we obtain that

$$\log(\text{MIP Running Time}) = 0.0083n - 2.2222, \quad R^2 = 0.8017,$$
$$\log(\text{Shigeno et al. [42] Running Time}) = 0.1032n - 0.7809, \quad R^2 = 0.91815,$$

where R^2 denotes the coefficient of determination. The slope of our approach is about 1/12 of previous work (0.0083 vs 0.1032). We observe that the running times and its growth rate are both much less than those in the previous work [42].

Note that our computational environments are different from Shigeno et al.'s. Although the constant term of the linear regression highly depends on the environments, the slope is less affected by them.

We use the problems with the same size of Muu and Thuy [37] to compare with theirs. The results of the experiments are shown in Table 3.4, where the sizes of generated networks are based on their experiments [37]. In Figure 3.4, we show the natural logarithm of the running time by Muu and Thuy's algorithm and ours.

Table 3.3: Comparison of running time (s) with Shigeno et al. [42]

m	$n(=p)$	MIP (s)	Shigeno et al. [42](s)
14	20	0.1171	1.07
14	24	0.1191	4.07
14	28	0.1285	6.19
14	32	0.1412	12.89
14	36	0.1538	41.29
14	40	0.1633	42.64
14	44	0.1729	91.16
14	48	0.1760	113.36
14	52	0.1778	166.84
14	56	0.1835	172.63
14	60	0.1800	195.84
14	64	0.1817	344.29
14	68	0.1853	407.18
14	72	0.1836	504.10
14	76	0.1820	623.54

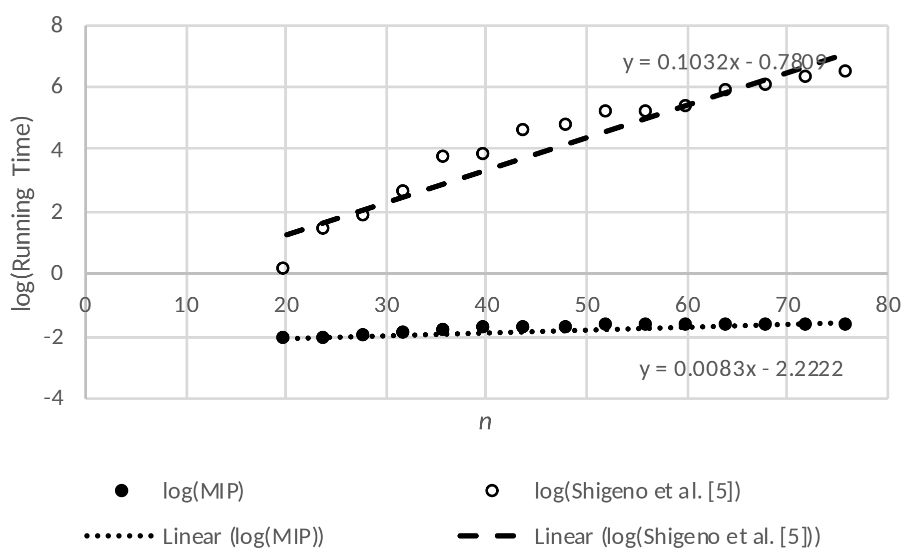


Figure 3.3: Comparison of log(Running Time) with Shigeno et al. [42]

It can be seen that our running times are less than theirs for all the instances. By regression analysis, we obtain that

$$\log(\text{MIP Running Time}) = 0.0022n - 2.129, \quad R^2 = 0.6685,$$
$$\log(\text{Muu and Thuy [37] Running Time}) = 0.0248n - 0.2729, \quad R^2 = 0.86433.$$

The slope of our approach is estimated as about 1/11 of previous work (0.0022 vs 0.0248). We can find that the running times and its growth rate are both much less than those in the previous work.

Table 3.4: Comparison of running time (s) with Muu and Thuy [37]

m	$n(=p)$	MIP (s)	Muu and Thuy [37](s)
6	10	0.1141	0.516
16	20	0.1147	0.982
30	70	0.1706	15.362
100	200	0.1747	74.254

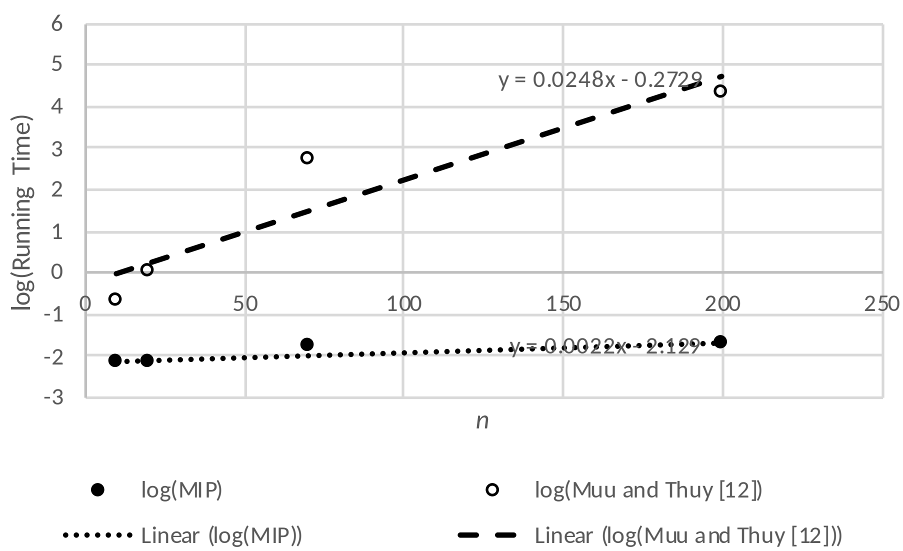


Figure 3.4: Comparison of log(Running Time) with Muu and Thuy [37]

Relatively Large Instances

As far as we know, the largest-scaled instance of the MMF problem is solved in Muu and Thuy [37] where $n = 200$. The results of computational experiments for large-scaled instances are shown in Table 3.5. We solve the MMF problems of two kinds of networks. In the table, the column 'S RT' and the column 'G RT' show the mean running time in seconds of 50 different instances of specially structured networks and generally structured networks, and the fourth column and the seventh column show the standard deviation. Accuracy shows the rate of instances for which we find the exact optimal solution.

Table 3.5: Experiments on large instances

m	$n(= p)$	S RT(s)	S SD	Accuracy(%)	G RT(s)	G SD
100	1,000	0.2862	0.0240	100	0.3259	0.0263
100	2,000	0.5566	0.0672	100	0.6071	0.0538
100	3,000	1.0794	0.1535	100	1.0642	0.0957
100	4,000	1.8428	0.3187	100	1.9177	0.2144
100	5,000	2.7916	0.4727	100	2.8733	0.4026
500	1,000	0.2452	0.0435	100	0.2787	0.0365
500	2,000	1.3790	0.8169	100	2.1319	1.0890
500	3,000	3.0705	1.1969	100	3.6991	1.7350
500	4,000	4.2061	1.6904	100	6.1973	3.2700
500	5,000	5.3831	1.8795	100	6.6702	2.3094
1,000	2,000	0.5811	0.1313	100	0.6088	0.0846
1,000	3,000	2.9635	1.9302	100	3.8681	2.6233
1,000	4,000	7.6708	5.6761	100	13.6337	8.6696
1,000	5,000	19.7831	14.1366	100	23.6851	18.5367
1,000	10,000	64.6637	6.7976	100	64.5310	6.3798
1,000	15,000	205.0781	18.2760	100	203.9493	16.6314

From Table 3.5, we see that our approach is capable to compute the exact solution to the MMF problems up to the size of $m = 1,000$ and $n = 15,000$ in a short time and the accuracy is 100%.

3.4 Brief Summary

We show that the MMF problem (3.1) of the network $\boldsymbol{N}$ can be formulated as the MIP problem (3.4) in Theorem 2, and we proposed to find the minimum maximal flow by solving the problem (3.4). Also, we show that in special cases, the value θ can be exactly established in Lemma 6. We observed from our preliminary computational experiments that: when comparing with Sun's MIP approach or Muu and Thuy's DC approach, the accuracy of our approach has been improved to 100%; Our approach showed a computational advantage to the previous researches and able to solve the relatively large instances; Furthermore the growth rate of the running time was slowed down much more than those in previous works.

Chapter 4

Application II: the Least Distance Problem in Data Envelopment Analysis

4.1 Introduction

Data Envelopment Analysis (DEA) has been extensively studied since Charnes et al. [13] proposed their Charnes, Cooper, and Rhodes (CCR) model. DEA has been widely applied to evaluating the relative efficiency of the assessed Decision Making Units (DMU) such as schools, banks, hospitals, and business units. In DEA, we assume that there are n DMUs, and each DMU$_j$ (for $j = 1, 2, \dots, n$) uses m inputs $\boldsymbol{x} \in \mathbb{R}^m$ to produce s outputs $\boldsymbol{y} \in \mathbb{R}^s$. A pair of inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$ is called an activity which can be expressed by the notation $(\boldsymbol{x}, \boldsymbol{y})$. The production possibility set $\boldsymbol{T}$ under the variable returns-to-scale (VRS) assumption is defined as

$$\boldsymbol{T} = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{m \times s} | \exists \boldsymbol{\lambda} \geqq 0 \text{ s.t. } \boldsymbol{X}\boldsymbol{\lambda} \leqq \boldsymbol{x},\ \boldsymbol{Y}\boldsymbol{\lambda} \geqq \boldsymbol{y},\ ,\boldsymbol{1}^\top \boldsymbol{\lambda} = 1,\ \boldsymbol{x}, \boldsymbol{y} > 0\},$$

where $\boldsymbol{\lambda} \in \mathbb{R}^n$, $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ is the input data matrix, and $\boldsymbol{Y} = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_n)$ is the output data matrix.

Definition 3 (Efficient Activity)**.** $(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{T}$ *is called an efficient activity, if there is no activity* $(\boldsymbol{x}', \boldsymbol{y}') \in \boldsymbol{T}$ *such that* $(\boldsymbol{x}, -\boldsymbol{y}) \geqq (\boldsymbol{x}', -\boldsymbol{y}')$ *and* $(\boldsymbol{x}, -\boldsymbol{y}) \neq (\boldsymbol{x}', -\boldsymbol{y}')$.

DEA provides not only the efficiency level, but also a kind of suggestive information (e.g. efficient target) for the inefficient DMUs to improve its efficiency level. Thus, providing such information is also an important part of DEA [12]. The classical DEA models, such as the CCR model, Banker, Charnes, and Cooper (BCC) model by Banker et al. [6] and Slacks-Based Measure (SBM) model by Tone [45], find an efficient target to compare with the assessed DMU. However, the required modifications when an inefficient DMU improves its efficiency level are not considered in the previous literature. Because these modifications are closely related to the necessary costs, the traditional research may not provide the most appropriate target for the assessed DMU. For example, suppose there are two kinds of healthy lifestyles, people can imitate to improve their health conditions, the lifestyle of a centenarian in the neighborhood and that of a top athlete. Imitating the lifestyle of a centenarian is more practical for common people, because we may only need to change the menu for dinner instead of living in a totally different way as an athlete. Under this consideration, Briec [12] estimated the required modifications by the distance, introduced the identification of the closest efficient target, and proposed the least distance problem to find an efficient target that needs the least modifications. Since the closest efficient target plays a growing important role in DEA research, the properties of the least distance problem (LDP) have been actively researched in recent years, see [2, 3, 20, 21, 19, 46, 47]. The LDP can be defined as:

$$\left|\begin{array}{ll} \text{minimize} & \|(\boldsymbol{x}_0, \boldsymbol{y}_0) - (\boldsymbol{x}, \boldsymbol{y})\|_p \\ \text{subject to} & (\boldsymbol{x}, \boldsymbol{y}) \text{ is an efficient activity.} \end{array}\right. \tag{4.1}$$

where $\|.\|_p$ is the l_p-norm. In the view of computation, there are two main difficulties of the problem (4.1) which minimizes the distance from the assessed DMU_0 to an efficient target over the whole efficient frontier which is defined by the set of efficient activities in DEA:

- the definition of the efficient frontier is given in an implicit form;
- the efficient frontier is nonconvex.

Aparicio et al. [5] introduced a set $\boldsymbol{D}_0$ which is the intersection of the efficient frontier and the set dominating the assessed DMU and proposed an MIP approach to search the closest efficient target from $\boldsymbol{D}_0$. In Aparicio et al. [4], they proposed a similar approach to solve the problem under the VRS assumption. However, the approaches of Aparicio et al. [4, 5] may not always find the closest efficient target.

Example 4. *Consider the LDP of l_1-distance for DMU A under the VRS assumption as the following Table 4.*

Table 4.1: Illustrative example

DMU	x_1	x_2	y	Efficiency
A	5	0.8	1.6	No
B	4	0.8	2.4	Yes
C	5	1	3	Yes

The closest efficient target for A by Aparicio et al. [5] is B. However, the l_1-distance here is 1.8 while the l_1-distance from A to C is 1.6. This is because the closest efficient target could be any points lying on the whole efficient frontier instead of the intersection.

In this chapter, we show that the efficient frontier is the efficient set to our proposed auxiliary problem, so the problem (4.1) can be written as an equivalent OE problem. Then an MIP problem is used for solving the OE problem. The main contribution of this chapter is that we provide a method to obtain the closest efficient target over the whole efficient frontier under the VRS assumption.

This chapter is organized as follows. In Section 4.2, we show that the LDP can be reformulated as an equivalent OE problem. By applying a reformulation and our MIP approach of the OE problem, a simplified equivalent MIP problem is shown as the implementation. In Section 4.3, the conclusions are given, and the limitations are discussed in remarks.

4.2 Model and Implementation

In this section, we show that the LDP can be reformulated as an equivalent OE problem with the auxiliary problem (4.2). The reformulated problem can be solved by a simplified equivalent MIP problem. First, we have the following Lemma 7 to define the efficiency.

Lemma 7. *An activity $(\boldsymbol{x}, \boldsymbol{y})$ is efficient, if and only if $(\boldsymbol{x}, \boldsymbol{y})$ is an efficient solution to the following MOLP problem (4.2):*

$$\left|\begin{array}{ll} \text{minimize} & \begin{pmatrix} \boldsymbol{x} \\ -\boldsymbol{y} \end{pmatrix} \\ \text{subject to} & \boldsymbol{X}\boldsymbol{\lambda} \leqq \boldsymbol{x}, \\ & \boldsymbol{Y}\boldsymbol{\lambda} \geqq \boldsymbol{y}, \\ & \boldsymbol{1}^\top \boldsymbol{\lambda} = 1, \\ & \boldsymbol{x}, \boldsymbol{y} > \boldsymbol{0},\ \boldsymbol{\lambda} \geqq \boldsymbol{0}. \end{array}\right. \tag{4.2}$$

Lemma 7 is obviously true since this lemma can be simply reformulated from Definition 3. Then we have the following Corollary 1.

Corollary 1. *The LDP problem is equivalent to the following OE problem:*

$$\left|\begin{array}{ll} \text{minimize} & \|(\boldsymbol{x}_0, \boldsymbol{y}_0) - (\boldsymbol{x}, \boldsymbol{y})\|_p \\ \text{subject to} & (\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{E}, \end{array}\right. \tag{4.3}$$

where $\boldsymbol{E}$ is the efficient set of the problem (4.2).

With Corollary 1, an MIP approach is almost able to be applied to solve the LDP. However, the strict inequality constraints and some variables make the problem unnecessarily complex. So we propose a new equivalent feasible set $\boldsymbol{H}$ to simplify the problem in following Lemmas 8 and 9.

Let k be the number of the efficient DMUs. Let DMU_j for $j = 1, \ldots, k$ be efficient, and DMU_j for $j = k+1, \ldots, n$ be inefficient without loss of generality. Let $\boldsymbol{X}_E$ and $\boldsymbol{Y}_E$ denote $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ and $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$, respectively. We have the following Lemma 8 to show the feasibility.

Lemma 8. *Let $\boldsymbol{H}$ be defined as*

$$\boldsymbol{H} = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{m \times s} | \exists \boldsymbol{\lambda} \geqq 0 \text{ s.t. } \boldsymbol{X}_E \boldsymbol{\lambda} = \boldsymbol{x},\ \boldsymbol{Y}_E \boldsymbol{\lambda} = \boldsymbol{y},\ \mathbf{1}^\top \lambda = 1\},$$

then

$$\boldsymbol{H} \subset \boldsymbol{T}.$$

Proof. The proof is simple, first we construct

$$\boldsymbol{T}' = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{m \times s} | \exists \boldsymbol{\lambda} \geqq 0 \text{ s.t. } \boldsymbol{X} \boldsymbol{\lambda} = \boldsymbol{x},\ \boldsymbol{Y} \boldsymbol{\lambda} = \boldsymbol{y},\ \mathbf{1}^\top \lambda = 1,\ \lambda \geqq \mathbf{0}\}.$$

Since all elements in $\boldsymbol{X}$ and $\boldsymbol{Y}$ are positive, any convex combination of them is constantly positive. So $\boldsymbol{T}' \subset \boldsymbol{T}$.

Also, since all elements in $\boldsymbol{X}_E$ and $\boldsymbol{Y}_E$ are contained in $\boldsymbol{X}$ and $\boldsymbol{Y}$, we have

$$\boldsymbol{H} \subset \boldsymbol{T}' \subset \boldsymbol{T}.$$

□

Then we show that the efficient frontier is contained in $\boldsymbol{H}$ by Lemma 9.

Lemma 9. *If an activity $(\boldsymbol{x}', \boldsymbol{y}') \in \boldsymbol{T}$ is efficient, then $(\boldsymbol{x}', \boldsymbol{y}') \in \boldsymbol{H}$.*

Proof. Suppose the contrary that $(\boldsymbol{x}', \boldsymbol{y}') = (\boldsymbol{x}'_1, \dots, \boldsymbol{x}'_m, \boldsymbol{y}'_1, \dots, \boldsymbol{y}'_s) \in \boldsymbol{T} \setminus \boldsymbol{H}$ is efficient. First, we show that, $(\boldsymbol{x}', \boldsymbol{y}') \in \boldsymbol{T}'$, otherwise, there exists i such that $(\boldsymbol{X}\boldsymbol{\lambda})_i < x'_i$ or $(\boldsymbol{Y}\boldsymbol{\lambda})_i > y'_i$, which is a contradiction for that there is an activity $(\boldsymbol{x}'_1, \dots, (\boldsymbol{X}\boldsymbol{\lambda})_i, \dots, \boldsymbol{x}'_m, \boldsymbol{y}'_1, \dots, \boldsymbol{y}'_s) \in \boldsymbol{T}$ or $(\boldsymbol{x}'_1, \dots, \boldsymbol{x}'_m, \boldsymbol{y}'_1, \dots, (\boldsymbol{Y}\boldsymbol{\lambda})_i, \dots, \boldsymbol{y}'_s) \in \boldsymbol{T}$. Thus, we have $(\boldsymbol{x}', \boldsymbol{y}') \in \boldsymbol{T}' \setminus \boldsymbol{H}'$. Since $(\boldsymbol{x}', \boldsymbol{y}') \in \boldsymbol{T}'$, without loss of generality, we assume the following:

$$\boldsymbol{x}' = \boldsymbol{X}_E\boldsymbol{\lambda}' + \boldsymbol{x}_{k+1}\lambda_{k+1},\ \boldsymbol{y}' = \boldsymbol{Y}_E\boldsymbol{\lambda}' + \boldsymbol{y}_{k+1}\lambda_{k+1},\ \boldsymbol{1}^\top\lambda' + \lambda_{k+1} = 1,\ \lambda \geqq \boldsymbol{0}, \lambda_{k+1} > 0,$$

where $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1})$ is inefficient. By the definition of efficient, there exists an activity $(\boldsymbol{x}'_{k+1}, \boldsymbol{y}'_{k+1}) \in \boldsymbol{T}$, such that $(\boldsymbol{x}_{k+1}, -\boldsymbol{y}_{k+1}) \geqq (\boldsymbol{x}'_{k+1}, -\boldsymbol{y}'_{k+1})$ and $(\boldsymbol{x}_{k+1}, -\boldsymbol{y}_{k+1}) \neq (\boldsymbol{x}'_{k+1}, -\boldsymbol{y}'_{k+1})$. Let $(\boldsymbol{x}'', \boldsymbol{y}'')$ defined as

$$\boldsymbol{x}'' = \boldsymbol{X}_E\boldsymbol{\lambda}' + \boldsymbol{x}'_{k+1}\lambda_{k+1},\ \boldsymbol{y}'' = \boldsymbol{Y}_E\boldsymbol{\lambda}' + \boldsymbol{y}'_{k+1}\lambda_{k+1},\ \boldsymbol{1}^\top\lambda' + \lambda_{k+1} = 1,\ \lambda \geqq \boldsymbol{0}, \lambda_{k+1} > 0.$$

Obviously, $(\boldsymbol{x}'', \boldsymbol{y}'') \in \boldsymbol{T}$, we have that $(\boldsymbol{x}', -\boldsymbol{y}') \geqq (\boldsymbol{x}'', -\boldsymbol{y}'')$ and $(\boldsymbol{x}', -\boldsymbol{y}') \neq (\boldsymbol{x}'', -\boldsymbol{y}'')$, which is a contradiction. □

With Lemmas 8 and 9, the problem (4.2) is equivalent to the following problem:

$$\left|\begin{array}{ll} \text{minimize} & \begin{pmatrix} \boldsymbol{x} \\ -\boldsymbol{y} \end{pmatrix} \\ \text{subject to} & \boldsymbol{X}\boldsymbol{\lambda} = \boldsymbol{x}, \\ & \boldsymbol{Y}\boldsymbol{\lambda} = \boldsymbol{y}, \\ & \boldsymbol{1}^\top\boldsymbol{\lambda} = 1, \\ & \boldsymbol{\lambda} \geqq \boldsymbol{0}. \end{array}\right. \tag{4.4}$$

Since the problem (4.4) is an MOLP problem over $\boldsymbol{H}$, we have an MIP approach of the LDP problem under VRS assumption with Corollary 1 and Theorem 2.

Theorem 3. *An optimal solution of the LDP problem (4.1) can be computed by solving the following MIP problem:*

$$\left|\begin{array}{ll} \text{minimize} & \|(\boldsymbol{x}_0, \boldsymbol{y}_0) - (\boldsymbol{x}, \boldsymbol{y})\|_p \\ \text{subject to} & \boldsymbol{X}_E\boldsymbol{\lambda} = \boldsymbol{x}, \\ & \boldsymbol{Y}_E\boldsymbol{\lambda} = \boldsymbol{y}, \\ & \boldsymbol{1}^\top\boldsymbol{\lambda} = 1, \\ & -\boldsymbol{u} - \boldsymbol{X}_E^\top\boldsymbol{v}_1 + \boldsymbol{Y}_E^\top\boldsymbol{v}_2 + \boldsymbol{1}v_3 = \boldsymbol{0} \\ & \boldsymbol{0} \leqq \boldsymbol{u} \leqq M\boldsymbol{b}, \boldsymbol{0} \leqq \boldsymbol{\lambda} \leqq M(\boldsymbol{1} - \boldsymbol{b}) \\ & \boldsymbol{v}_1 \leqq -\boldsymbol{1}, \boldsymbol{v}_2 \geqq \boldsymbol{1}, \boldsymbol{b} \in \{0, 1\}^k, \end{array}\right. \tag{4.5}$$

where M is a large number and k is the number of the efficient DMUs.

When $p \geqq 1$, the objective function of the problem (4.5) is convex, the problem (4.5) can be efficiently solved by current state-of-the-art MIP solvers. The proposed MIP approach can obtain C instead of B as the closest efficient target for A in Example 4. We note that the major difference between previous research and ours that coursed the different efficient targets. The efficient frontier is defined by two different MOLP problems, with or without the VRS assumption. The original MOLP problem in [5, 4] is equivalent to the following problem (4.6):

$$\left|\begin{array}{ll} \text{minimize} & \begin{pmatrix} -\boldsymbol{s}^- \\ -\boldsymbol{s}^+ \end{pmatrix} \\ \text{subject to} & \boldsymbol{X}\boldsymbol{\lambda} \leqq \boldsymbol{x}_0 - \boldsymbol{s}^-, \\ & \boldsymbol{Y}\boldsymbol{\lambda} \geqq \boldsymbol{y}_0 + \boldsymbol{s}^+, \\ & \boldsymbol{1}^\top \boldsymbol{\lambda} = 1, \\ & \boldsymbol{x}_0 - \boldsymbol{s}^- > \boldsymbol{0}, \boldsymbol{y}_0 + \boldsymbol{s}^+ > \boldsymbol{0}, \\ & \boldsymbol{s}^-, \boldsymbol{s}^+, \boldsymbol{\lambda} \geqq \boldsymbol{0}. \end{array}\right. \tag{4.6}$$

The set $\boldsymbol{D}_0$ is the efficient set of the problem 4.6. Since the efficient frontier is constant while the set $\boldsymbol{D}_0$ is different when assessing a different DMU, our proposed frontier is more intuitive in economics. In the LDP, there is no evidence that shows the closest efficient target always lies in $\boldsymbol{D}_0$, see Example 4. Since $\boldsymbol{D}_0 \subset \boldsymbol{E}$, (4.2)'s frontier can always provide an efficient target that is no further than the efficient target (4.6)'s frontier provides. To show this point more intuitively, we make a bird-view figure of Example 4 as the following Figure 4.1. In this example, the feasible region $\boldsymbol{D}_0$ of previous research is $\{B\}$ while the frontier is the line segment BC.

On the other hand, many classical DEA researches (e.g. maximizing the sum of slacks) under constant returns-to-scale assumption are defined based on (4.6)'s frontier since the frontier of (4.2) could be unbounded in that case.

4.3 Brief Summary and Remarks

In this section, we offer a brief summary and several remarks of this chapter.

Brief Summary

In this chapter, we defined the efficient frontier with the MOLP problem (4.2). Then, the LDP (4.1) is reformulated into the OE problem (4.3). Finally, we proposed an equivalent simplified problem (4.5) which can be solved by an MIP

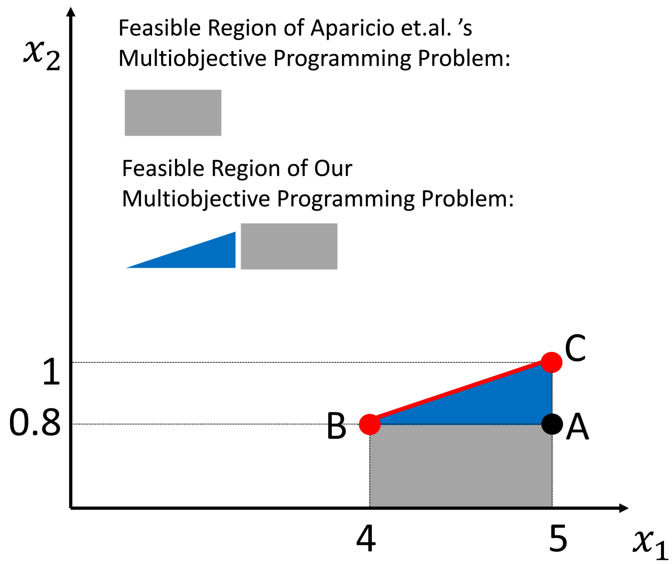


Figure 4.1: Bird's-eye View of the Example

solver. Compared with previous research, the proposed approach obtains the closest efficient target over the whole efficient frontier instead of part of it under the VRS assumption with the theoretical guarantee.

Remarks

There are still two questions remained to be discussed.

- **Monotonicity**
  When considering the distance, an important property is necessary: when the assessed DMU has a longer distance, the assessed DMU is more inefficient. Ando et al. [2] considered this question first, and proposed to search the least distance between an activity which is dominated by the assessed DMU and the efficient frontier. The problem they considered can be written as the following:

$$\left|\begin{array}{ll} \text{minimize} & \|(\boldsymbol{x}', \boldsymbol{y}') - (\boldsymbol{x}, \boldsymbol{y})\|_p \\ \text{subject to} & (\boldsymbol{x}, \boldsymbol{y}) \text{ is an efficient activity,} \\ & (\boldsymbol{x}_0, -\boldsymbol{y}_0) \geqq (\boldsymbol{x}', -\boldsymbol{y}'), \\ & \boldsymbol{x}', \boldsymbol{y}'. \end{array}\right. \tag{4.7}$$

When applying the proposed approach, the problem can be solved by the following MIP problem:

$$
\left|
\begin{array}{ll}
\underset{\boldsymbol{x},\boldsymbol{y},}{\text{minimize}} & \|(\boldsymbol{x}',\boldsymbol{y}')-(\boldsymbol{x},\boldsymbol{y})\|_p \\
\text{subject to} & \boldsymbol{X}_E\boldsymbol{\lambda}=\boldsymbol{x}, \\
& \boldsymbol{Y}_E\boldsymbol{\lambda}=\boldsymbol{y}, \\
& \mathbf{1}^\top\boldsymbol{\lambda}=1, \\
& -\boldsymbol{u}-\boldsymbol{X}_E^\top\boldsymbol{v}_1+\boldsymbol{Y}_E^\top\boldsymbol{v}_2+\mathbf{1}v_3=\mathbf{0}, \\
& \mathbf{0}\leqq\boldsymbol{u}\leqq M\boldsymbol{b},\mathbf{0}\leqq\boldsymbol{\lambda}\leqq M(\mathbf{1}-\boldsymbol{b}), \\
& \boldsymbol{v}_1\leqq-\mathbf{1},\boldsymbol{v}_2\geqq\mathbf{1},\boldsymbol{b}\in\{0,1\}^k, \\
& (\boldsymbol{x}_0,-\boldsymbol{y}_0)\leqq(\boldsymbol{x}',-\boldsymbol{y}'), \\
& \boldsymbol{x}',\boldsymbol{y}'\geqq\mathbf{0}.
\end{array}
\right.
\tag{4.8}
$$

However, either the problem (4.7) or the problem (4.8) faces a same conflict. The reduction of the efficiency level needs no cost in the feasible region while it needs in the objective function.

- **Distance Function**
  Since the weight is different from one input to another, a simple l_p-norm cannot describe it. Another problem is that since the real case is so complex, the function of l_p-norm may not fit for every cost. The effort of increasing one unit of the labor cost may be different from the effort to decrease it. Of course, we can optimize the following problem (4.9):

$$
\left|
\begin{array}{ll}
\underset{\boldsymbol{x},\boldsymbol{y},}{\text{minimize}} & c_{(\boldsymbol{x}_0,\boldsymbol{y}_0)}(\boldsymbol{x},\boldsymbol{y}) \\
\text{subject to} & \boldsymbol{X}_E\boldsymbol{\lambda}=\boldsymbol{x}, \\
& \boldsymbol{Y}_E\boldsymbol{\lambda}=\boldsymbol{y}, \\
& \mathbf{1}^\top\boldsymbol{\lambda}=1, \\
& -\boldsymbol{u}-\boldsymbol{X}_E^\top\boldsymbol{v}_1+\boldsymbol{Y}_E^\top\boldsymbol{v}_2+\mathbf{1}v_3=\mathbf{0} \\
& \mathbf{0}\leqq\boldsymbol{u}\leqq M\boldsymbol{b},\mathbf{0}\leqq\boldsymbol{\lambda}\leqq M(\mathbf{1}-\boldsymbol{b}) \\
& \boldsymbol{v}_1\leqq-\mathbf{1},\boldsymbol{v}_2\geqq\mathbf{1},\boldsymbol{b}\in\{0,1\}^k,
\end{array}
\right.
\tag{4.9}
$$

where $c_{(\boldsymbol{x}_0,\boldsymbol{y}_0)}(\boldsymbol{x},\boldsymbol{y})$ is a general cost function. Still, we need to discuss the cost function case by case.

Chapter 5

Conclusion

In this thesis, we showed that an optimal solution to the optimization problem (1.2) over the efficient set of the multiobjective linear programming problem (1.1) can be computed by solving the MIP problem (2.2). Compared with the previous MIP approach by Sun [43], our approach does not require Assumption 1 on the problem (1.1) and reduces the size of the MIP problem. By conducting the preliminary computational experiments, we observed that our approach can solve an OE problem faster and more accurately.

The proposed approach provides a more practical implementation to solve the applications of the OE problem that arise from the real world. Since the MMF problem, which is a class of network flow problem, is an important application of the OE problem, it is valuable to propose an efficient algorithm. We proposed an MIP approach that can efficiently solve the MMF problem with an MIP solver. In the numerical experiments, it performed more accurately than the previous MIP approach and the DC approach. Compared with the global optimization approach and the DC approach, the running time and the growth of the running time are improved as well.

Another application is the LDP in DEA. We showed that this problem is a class of OE problem and proposed a simplified equivalent MIP problem to solve it. Compared with previous approaches, the model of our approach is more accurate to reflect the economic activities by computing on the whole efficient frontier. Also, the MIP approach of the LDP can be efficiently solved by an MIP solver.

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice hall, 1993.

[2] K. Ando, A. Kai, Y. Maeda, and K. Sekitani. Least distance based inefficiency measures on the pareto-efficient frontier in DEA. *Journal of the Operations Research Society of Japan*, 55(1):73–91, 2012.

[3] K. Ando, M. Minamide, K. Sekitani, and J. Shi. Monotonicity of minimum distance inefficiency measures for data envelopment analysis. *European Journal of Operational Research*, 260(1):232–243, 2017.

[4] J. Aparicio, J. M. Cordero, and J. T. Pastor. The determination of the least distance to the strongly efficient frontier in data envelopment analysis oriented models: modelling and computational aspects. *Omega*, 71:1–10, 2017.

[5] J. Aparicio, J. L. Ruiz, and I. Sirvent. Closest targets and minimum distance to the pareto-efficient frontier in DEA. *Journal of Productivity Analysis*, 28(3):209–218, 2007.

[6] R. D. Banker, A. Charnes, and W. W. Cooper. Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management science*, 30(9):1078–1092, 1984.

[7] H. P. Benson. An algorithm for optimizing over the weakly-efficient set. *European Journal of Operational Research*, 25(2):192–199, 1986.

[8] H. P. Benson. A finite, nonadjacent extreme-point search algorithm for optimization over the efficient set. *Journal of Optimization Theory and Applications*, 73(1):47–64, 1992.

[9] H. P. Benson. An outcome space algorithm for optimization over the weakly efficient set of a multiple objective nonlinear programming problem. *Journal of Global Optimization*, 52(3):553–574, 2012.

[10] H. P. Benson and D. Lee. Outcome-based algorithm for optimizing over the efficient set of a bicriteria linear programming problem. *Journal of Optimization Theory and Applications*, 88(1):77–105, 1996.

[11] S. Bolintineanu. Minimization of a quasi-concave function over an efficient set. *Mathematical Programming*, 61(1-3):89–110, 1993.

[12] W. Briec. Hölder distance function and measurement of technical efficiency. *Journal of Productivity Analysis*, 11(2):111–131, 1999.

[13] A. Charnes, W. W. Cooper, and E. Rhodes. Measuring the efficiency of decision making units. *European journal of operational research*, 2(6):429–444, 1978.

[14] W. Chen, S. Masahiro, and J. Shi. The effective algorithm for minimum maximal network flow. *IPSJ SIG technical reports*, 2009(2):1–6, 2009, (in Japanese).

[15] R. W. Cottle, J.-S. Pang, and R. E. Stone. *The Linear Complementarity Problem*, volume 60. SIAM, 1992.

[16] G. B. Dantzig. Application of the simplex method to a transportation problem. *Activity Analysis and Production and Allocation*, 1951.

[17] J. P. Dauer and T. A. Fosnaugh. Optimization over the efficient set. *Journal of Global Optimization*, 7(3):261–277, 1995.

[18] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[19] H. Fukuyama, J. Leth Hougaard, K. Sekitani, and J. Shi. Efficiency measurement with a non-convex free disposal hull technology. *Journal of the Operational Research Society*, 67(1):9–19, 2016.

[20] H. Fukuyama, Y. Maeda, K. Sekitani, and J. Shi. Input-output substitutability and strongly monotonic p-norm least distance DEA measures. *European Journal of Operational Research*, 237(3):997–1007, 2014.

[21] H. Fukuyama, H. Masaki, K. Sekitani, and J. Shi. Distance optimization approach to ratio-form efficiency measures in data envelopment analysis. *Journal of Productivity Analysis*, 42(2):175–186, 2014.

[22] J. Gotoh, N. V. Thoai, and Y. Yamamoto. Global optimization method for solving the minimum maximal flow problem. *Optimization Methods and Software*, 18(4):395–415, 2003.

[23] L. T. Hoai An and P. D. Tao. Dc programming and dca: thirty years of developments. *Mathematical Programming*, 169(1):5–68, 2018.

[24] L. T. Hoai An, P. D. Tao, and L. D. Muu. Numerical solution for optimization over the efficient set by dc optimization algorithms. *Operations Research Letters*, 19(3):117–128, 1996.

[25] J. Hu, J. E. Mitchell, J.-S. Pang, and B. Yu. On linear programs with linear complementarity constraints. *Journal of Global Optimization*, 53(1):29–51, 2012.

[26] M. Iri. An essay in the theory of uncontrollable flows and congestion. *Department of Information and System Engineering, Chuo University, TRISE*, pages 94–03, 1994.

[27] Y. Ji and J. Shi. An algorithm for minimum maximal network flow. *IPSJ SIG technical reports*, 2011(1):1–6, 2011, (in Japanese).

[28] M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey. *50 Years of integer programming 1958-2008: From the early years to the state-of-the-art*. Springer Science & Business Media, 2009.

[29] L. G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

[30] Z. Liu and M. Ehrgott. Primal and dual algorithms for optimization over the efficient set. *Optimization*, 67(10):1661–1686, 2018.

[31] K. Lu. Efficient algorithms for solving the minimum maximal flow problem. Master's thesis, Graduate School of Management, 2017.

[32] K. Lu, S. Mizuno, and J. Shi. A mixed integer programming approach for the minimum maximal flow. *Journal of the Operations Research Society of Japan*, 64(4):261–271, 2018.

[33] K. Lu, S. Mizuno, and J. Shi. A numerical study on mip approaches over the efficient set. In *Optimization of Complex Systems: Theory, Models, Algorithms and Applications. WCGO 2019. Advances in Intelligent Systems and Computing*, pages 611–616. Springer, 2020.

[34] K. Lu, S. Mizuno, and J. Shi. A new mixed integer programming approach for optimization over the efficient set of a multiobjective linear programming problem. *Optimization Letters*, (Accepted).

[35] K. Lu, S. Mizuno, J. Shi, and X. Wang. Solving the least distance problem under the vrs assumption in dea as a bilevel programming problem. In *Proceedings of 2019 International Conference on Business, Big-Data, and Decision Sciences*, pages 305–311. National Taipei University of Technology, 2019.

[36] Z. Q. Luo, J. S. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

[37] L. D. Muu and L. Q. Thuy. On dc optimization algorithms for solving minmax flow problems. *Mathematical Methods of Operations Research*, 80(1):83–97, 2014.

[38] J. Philip. Algorithms for the vector maximization problem. *Mathematical Programming*, 2(1):207–229, 1972.

[39] T. Q. Phong and J. Q. Tuyen. Bisection search algorithm for optimizing over the efficient set. *Vietnam Journal of Mathematics*, 28:217–226, 2000.

[40] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[41] J. Shi and Y. Yamamoto. A global optimization method for minimum maximal flow problem. *Acta Mathematica Vietnamica*, 22(1):271–287, 1997.

[42] M. Shigeno, I. Takahashi, and Y. Yamamoto. Minimum maximal flow problem: an optimization over the efficient set. *Journal of Global Optimization*, 25(4):425–443, 2003.

[43] E. Sun. On optimization over the efficient set of a multiple objective linear programming problem. *Journal of Optimization Theory and Applications*, 172(1):236–246, 2017.

[44] P. T. Thach, H. Konno, and D. Yokota. Dual approach to minimization on the set of pareto-optimal solutions. *Journal of Optimization Theory and Applications*, 88(3):689–707, 1996.

[45] K. Tone. A slacks-based measure of efficiency in data envelopment analysis. *European journal of operational research*, 130(3):498–509, 2001.

[46] K. Tone. Variations on the theme of slacks-based measure of efficiency in DEA. *European Journal of Operational Research*, 200(3):901–907, 2010.

[47] K. Tone. Data envelopment analysis as a kaizen tool: SBM variations revisited. *Bulletin of Mathematical Sciences and Applications*, 16:49–61, 8 2016.

[48] H. Tuy. *Convex Analysis and Global Optimization*. Springer, 2016.

[49] R. E. Wendell and D. Lee. Efficiency in multiple objective optimization problems. *Mathematical Programming*, 12(1):406–414, 1977.

[50] Y. Yamamoto. Optimization over the efficient set: overview. *Journal of Global Optimization*, 22(1-4):285–317, 2002.

[51] Y. Yamamoto and D. Zenke. Outer approximation method for the minimum maximal flow problem. *Journal of the Operations Research Society of Japan*, 50(1):14–30, 2007.