Algorithms for the Minimum Maximal Flow Problem

- Optimization over the Efficient Set -

Graduate School of Systems and Information Engineering

University of Tsukuba

March, 2006

Daisuke Zenke

Algorithms for the Minimum Maximal Flow Problem

- Optimization over the Efficient Set -

Daisuke Zenke
(Doctoral Program in Policy and Planning Sciences)

Adviser
Professor Yoshitsugu Yamamoto

Submitted to the Graduate School of
Systems and Information Engineering
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Engineering
at the University of Tsukuba

March, 2006

Abstract

When the system is composed of several subsystems with each own criterion to optimize, one cannot assume that there is a person who can control all the variables of the system. One has to accept one of the solutions at which each subsystem is optimizing its own criterion under constraints that must be observed by the subsystems, i.e., an efficient solution of the multicriteria optimization problem. The solution thus obtained depends on the way of compromise, and can be far apart from the ideal solution that one could attain with full controllability of all the variables. Therefore it counts in designing systems to evaluate how the compromise solution can deteriorate the efficiency of the whole system in the presence of such uncontrollability. This issue is formulated as an optimization problem over the efficient set of a multicriteria optimization problem.

Most of the algorithms for optimization problem over the efficient set anticipate a small number of criteria of the multicriteria optimization problem and convert the problem to a global optimization problem in variables of the number of criteria or so. However, there are interesting and important problems that do not enjoy the low dimensionality of the number of criteria. A typical example is the minimum maximal flow problem, which is the leading motive of this thesis. This problem is to find a maximal flow that attains the minimum flow value. A feasible flow is said to be maximal if there is no feasible flows that is greater than the flow with respect to the partial order of sectors. When one regards each arc as a subsystem which intends to maximize the flow on the arc, the minimum maximal flow problem will furnish an answer about how inefficiently the network can be used. Thus the minimum maximal flow value is of useful for network analysis and network design.

The set of efficient points is usually described by the gap function. It is, however, not defined on the whole set, which often offers computational difficulty. Then we

propose an extension of the gap function so that the function value can be evaluated at any point that is possibly encountered during the computation. We introduce ε-optimal solution to endow the problem with regularity. Then we develop two algorithms for this problem: the cut-and-split method and the outer approximation method. Concerning the convergence of the algorithms, we show by the integrality property of the network flow problems that the outer approximation method terminates after finitely many iterations with the optimal value of the problem. We also show that the cut-and-split method generates a sequence of incumbent solutions that converges to an optimal solution. Exploiting the flow conservation equations of the problem, we also propose an improvement on the algorithms by reducing the number of variables. Finally we propose a heuristics to locate an initial incumbent, we carry out the computational experiment, and report their empirical efficiency. We observe that the outer approximation method surpasses both an application of vertex enumeration method and the algorithm by Shigeno-Takahashi-Yamamoto [48] in computational time, especially as the number of variables grows. We also observe that our heuristics using the modified local search procedure provides a pretty good initial incumbent.

Contents

List of Figures

List of Tables

Notations

Throughout this thesis, we use the following notations.

notation	definition and explanation
$\mathbb{N}$	set of natural numbers
$\mathbb{Z}$	set of integers
$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	set of n-dimensional real column vectors
$\mathbb{R}^n_+$, $\mathbb{R}^n_{++}$	$\mathbb{R}^n_+ = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \geqq \boldsymbol{0} \,\}$, $\mathbb{R}^n_{++} = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} > \boldsymbol{0} \,\}$
$\mathbb{R}_n$	set of n-dimensional real row vectors
$\mathbb{R}_{n+}$, $\mathbb{R}_{n++}$	$\mathbb{R}_{n+} = \{\, \boldsymbol{x} \in \mathbb{R}_n \mid \boldsymbol{x} \geqq \boldsymbol{0} \,\}$, $\mathbb{R}_{n++} = \{\, \boldsymbol{x} \in \mathbb{R}_n \mid \boldsymbol{x} > \boldsymbol{0} \,\}$
I	identity matrix of an appropriate size
O	zero matrix of an appropriate size
$\boldsymbol{e}$	row vector of ones of an appropriate dimension
$\boldsymbol{e}^j$	the jth unit row vector of an appropriate dimension
$\boldsymbol{1}$	column vector of ones of an appropriate dimension
$\boldsymbol{a}^\top$, $A^\top$	transpose vector of $\boldsymbol{a}$, transpose matrix of A
$\mathrm{cl}\, S$	closure of set S
$\mathrm{int}\, S$	interior of set S
$\partial\, S$	relative boundary of set S
P_V	set of vertices of a polyhedron P
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	inner product of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$
$\|\boldsymbol{x}\|$	norm of vector $\boldsymbol{x}$, i.e., $\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$

Chapter 1

Introduction

In this chapter, we discuss our motivation for considering the minimum maximal flow problem, abbreviated to (mmF), whose precise definition will be given in Section 4.1. We explain what information is obtained from the minimum flow value over the set of maximal flows, how it differs from the conventional maximum flow value, and in what situation such information is useful. We survey the background of the problem to clarify the position of our study. The purpose, results and the organization of the thesis are also stated.

1.1 Motivation

In the field of network flow theory such as maximum flow problem and minimum cost flow problem, we usually take it for granted that we can control arc flows, namely we can freely increase and decrease each arc flow as long as the feasibility is met. However, when we attempt to solve a maximum flow problem on condition that we are not be allowed to decrease arc flows, we often fail to obtain the maximum flow and are obliged to put up with a maximal flow. The minimum flow value attained by a maximal flow, i.e., the optimal value of (mmF), indicates how inefficiently the network can be utilized in the presence of some uncontrollability. Therefore the network should be so designed that the optimal value of (mmF) can be as large as possible to avoid a catastrophic situation on network conjection.

1.2 Background

Global optimization as well as combinatorial optimization is a natural direction to proceed for the research of mathematical programming that has successfully developed algorithms for linear and convex optimization. Then over the past several decades, lots of studies have been made on global optimization problems such as concave minimization problem, reverse convex problem, and D.C. problem. The difficulty shared by these problems is multiextremality, i.e., there are several locally optimal solutions with different objective function values. This difficulty is mainly due to the nonconvexity of the feasible region, the objective and constraint functions. A typical nonconvex feasible region is given as a union of multiple convex sets. If we are given all the convex sets explicitly, the problem simply reduces to multiple convex optimization problems. However, there are problems where the nonconvex feasible region is given as the set of efficient points of some multicriteria optimization problem. Such problem is called the optimization problem over the efficient set, whose precise definition will be given in Section 3.1, and we denoted it by (P_E).

Since Philip first considered (P_E) and proposed an algorithm based on local search and cutting plane technique in [40], a number of papers followed his work. The overview about the efficient set and several algorithms for (P_E) can be found in Yamamoto [66]. For the details about (P_E), the reader should refer to White [63], Sawaragi-Nakayama-Tanino [44] and Steuer [49]. The mathematical structure of the efficient set is studied in Naccache [38], Benson [10] and Hu-Sun [28]. The method enumerating the efficient vertices can be found in Ecker-Kouada [17, 18]. For solution methods for (P_E), see Benson [7–9], Bolintineanu [13], Ecker-Song [19], Fülöp [22], Dauer-Fosnaugh [16], Thach-Konno-Yokota [56], Sayin [46], Phong-Tuyen [41], Thoai [59], Muu-Luc [36] and An-Tao-Thoai [5]. The D.C. optimization algorithm for solving (P_E) can be seen in An-Tao-Muu [3, 4]. Some algorithms on D.C. optimization can be found in Tuy [61, 62] and Horst-Tuy [27].

It is known that (mmF) is a special, relatively difficult, case of (P_E). Shi-Yamamoto [47] first studied (mmF) and proposed an algorithm. After this, several algorithms for (mmF) combining local search and global optimization technique have been proposed in e.g., Gotoh-Thoai-Yamamoto [24] and Shigeno-Takahashi-

Yamamoto [48]. An approach based on D.C. optimization is found in Muu-Shi [37]. The difficulty of the problem is due to the nonconvexity of the set of maximal flows, implying a lot of locally optimal solutions with different objective function values. Indeed, (mmF) embraces the minimum maximal matching problem, which is $\mathcal{NP}$-hard (See e.g., Garey-Johnson [23]), and hence sophisticated algorithms are required. We have neither theoretical evidence for the efficiency of existing algorithms, nor comparative study of them from the computational viewpoint.

1.3 Purpose and Results of the Thesis

The purpose of the thesis is to propose two algorithms for solving (mmF): the cut-and-split method (CS method for short) and the outer approximation method (OA method for short), show their convergence property and demonstrate computational efficiency.

The results of the thesis are the following.

- We extend the gap function characterizing the set of maximal flows.
- Combining the local search procedure, we propose the CS method and the OA method for solving (mmF).
- The convergence of the CS method for (mmF) is discussed.
- Introducing the idea of ε-optimal solution, we study the optimality condition of the OA method for (mmF).
- We show that the OA method for (mmF) terminates after finitely many iterations with the optimal value of the problem.
- We improve both of the CS method and the OA method by reducing the number of variables.
- Some implementation issues are studied.
- We report the result of computational experiments to verify the efficiency of the algorithms.

1.4 Organization of the Thesis

We discuss the global optimization in the next chapter, in which we define a D.C. problem, and explain the cut-and-split method and the outer approximation method for D.C. problems. Chapter 3 is devoted to the summary of optimization problem over the efficient set (P_E). Some known results on an efficient set and (P_E) are presented. In Chapter 4 we define a minimum maximal flow problem (mmF), and then explain some algorithms for (mmF) preceding our study. To make this thesis self-contained, we will outline some of the proofs, which are found in the early studies, in Chapter 2, Chapter 3 and Chapter 4. In Chapter 5, the main chapter of this thesis, we propose the CS method and the OA method for solving (mmF), and discuss their convergence properties. Exploiting the structure of the set of feasible flows, we improve the algorithms in Section 5.5. We report some results of computational experiments in Section 5.6. Conclusion and further works are described in the last chapter.

Chapter 2

Global Optimization

A *general continuous optimization problem* is formulated as

$$(P) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & f(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in D, \end{array} \right.$$

where $f : D \to [-\infty, +\infty]$, called an *objective function*, is a continuous function on a nonempty closed set D, called a *feasible set*, of $\mathbb{R}^n$. A feasible solution $\boldsymbol{x}^* \in D$ is said to be a *globally optimal solution* or simply an *optimal solution* if it satisfies $f(\boldsymbol{x}^*) > -\infty$ and $f(\boldsymbol{x}^*) \leqq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in D$. Here we define a *δ-neighborhood of* $\boldsymbol{x}$, denoted by $N_\delta(\boldsymbol{x})$, as

$$N_\delta(\boldsymbol{x}) = \{\, \boldsymbol{x}' \in \mathbb{R}^n \mid \|\boldsymbol{x}' - \boldsymbol{x}\| < \delta \,\}, \tag{2.1}$$

for $\boldsymbol{x} \in \mathbb{R}^n$ and for $\delta > 0$. A feasible solution $\boldsymbol{x}^* \in D$ is said to be a *locally optimal solution* if $f(\boldsymbol{x}^*) > -\infty$ and there is $\delta > 0$ such that $f(\boldsymbol{x}^*) \leqq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in D \cap N_\delta(\boldsymbol{x}^*)$. The objective function value attained by a (locally) optimal solution is called the *(locally) optimal value*. Note that since

$$\min\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in D\} = -\max\{-f(\boldsymbol{x}) \mid \boldsymbol{x} \in D\},$$

a minimization problem can be reformulated as a maximization problem, and a (locally) optimal solution and the (locally) optimal value are defined for a maximization problem in the similar way.

The above problem (P) is equivalent to

$$(\overline{P}) \qquad \left| \begin{array}{ll} \min\limits_{(\boldsymbol{x},z)} & z \\ \text{s.t.} & (\boldsymbol{x},z) \in S, \end{array} \right.$$

where $\boldsymbol{x}$

$$S = \{\, (\boldsymbol{x},z) \in \mathbb{R}^n \times \mathbb{R} \mid \boldsymbol{x} \in D,\ f(\boldsymbol{x}) \leqq z \,\}.$$

Then, without loss of generality we assume, throughout this chapter, that the objective function of (P) is a linear function, i.e.,

$$f(\boldsymbol{x}) = \boldsymbol{p}\boldsymbol{x}, \tag{2.2}$$

for some cost vector $\boldsymbol{p} \in \mathbb{R}_n$.

We say that problem (P) is *convex* if D is convex, and *nonconvex* otherwise. When D is convex, all locally optimal solutions of (P) have the same objective function value, and hence they are all globally optimal solutions. In this case, local search algorithm provides a globally optimal solution. An optimization problem is said to be *multiextremal* if there are locally optimal solutions with different objective function values. The difficulty of a global optimization problem is mainly due to the multiextremality. When D is nonconvex, (P) may be multiextremal, and hence some locally optimal solutions may fail to be globally optimal (See Figure 2.1). Some examples of a multiextremal problem such as production transportation planning problem and pooling-and-blending problem can be found in Tuy [62].

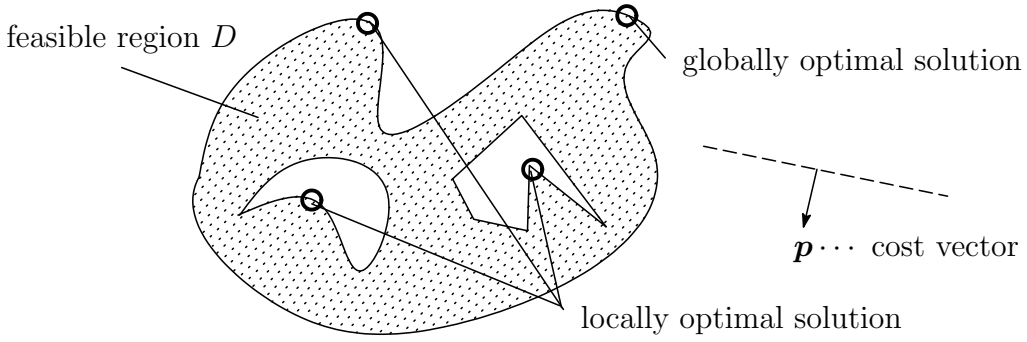


Figure 2.1: A multiextremal problem

In the next section after we define a D.C. problem, we show that (P) is a D.C. problem whenever D is closed. Also we define a canonical form D.C. problem (CDC) and a linear reverse convex problem $(LRCP)$, which are special cases of

D.C. problems. Then a cut-and-split method and an outer approximation method are briefly explained in Section 2.2 and Section 2.3. For the details about these methods, and some other methods for D.C. problems, the reader should refer to Tuy [61, 62] and Horst-Tuy [27]. In the last section of the chapter, we describe the on-line vertex enumeration procedure called the double description method. The procedure will be used in the outer approximation method.

2.1 D.C. Problem

A set S is said to be a *D.C. set* (difference of two convex sets) if there are two convex sets S_1 and S_2 such that $S = S_1 \setminus S_2$. Similarly, a function f is said to be a *D.C. function* (difference of two convex functions) if there are two convex functions f_1 and f_2 such that $f = f_1 - f_2$. An optimization problem described in terms of D.C. sets and/or D.C. functions is called a *D.C. problem*, which is studied in e.g., Tuy [61, 62] and Horst-Tuy [27]. D.C. problem covers many of nonlinear problems such as location planning problem, engineering design problem, multilevel problem, and optimization problem over the efficient set.

2.1.1 Relation between a D.C. problem and a general continuous optimization problem

Given a nonempty closed set $D \subsetneqq \mathbb{R}^n$, let

$$\mathrm{dist}_D(\boldsymbol{x}) = \inf\{\, \|\boldsymbol{x} - \boldsymbol{y}\|^2 \mid \boldsymbol{y} \in D \,\}.$$

Clearly, $\mathrm{dist}_D(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in D$. On the other hand, closedness of D implies that $\mathrm{dist}_D(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \notin D$. Then we have

$$D = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \mathrm{dist}_D(\boldsymbol{x}) \leqq 0 \,\} = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}) \leqq 0 \,\},$$

where $f_1(\boldsymbol{x}) = \|\boldsymbol{x}\|^2$ and $f_2(\boldsymbol{x}) = \|\boldsymbol{x}\|^2 - \mathrm{dist}_D(\boldsymbol{x})$. The function f_2 is a point-wise supremum of infinitely many linear functions, since $f_2(\boldsymbol{x}) = \sup\{\, 2\langle \boldsymbol{x}, \boldsymbol{y} \rangle - \|\boldsymbol{y}\|^2 \mid \boldsymbol{y} \in D \,\}$. Hence f_2 as well as f_1 are convex, which means that dist_D is a D.C. function. Therefore problem (P) can be rewritten as the D.C. problem

$$(P) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{px} \\ \text{s.t.} & \mathrm{dist}_D(\boldsymbol{x}) \leqq 0, \end{array} \right.$$

whenever D is closed. Note that $\mathrm{dist}_D(\boldsymbol{x}) \leqq 0$ is called a *D.C. inequality.*

2.1.2 Canonical form D.C. problem (CDC) and linear reverse convex problem $(LRCP)$

Definition 2.1 (Canonical form D.C. problem) *A canonical form D.C. problem, abbreviated to* (CDC)*, is*

$$(CDC) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{px} \\ s.t. & \boldsymbol{x} \in D \\ & h(\boldsymbol{x}) \geqq 0, \end{array} \right.$$

where $D \subsetneqq \mathbb{R}^n$ *is a nonempty closed convex set,* $\boldsymbol{p} \in \mathbb{R}_n$ *is a cost vector and* $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *is a convex function.*

Note that the constraint $h(\boldsymbol{x}) \geqq 0$ is called a *reverse convex constraint.* Here we assume that $\mathrm{int}\,\{\, \boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \leqq 0 \,\} = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) < 0 \,\}$, and define the convex set

$$H = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \leqq 0 \,\}. \tag{2.3}$$

Then we can rewrite (CDC) as

$$\left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{px} \\ \text{s.t.} & \boldsymbol{x} \in D \setminus \mathrm{int}\, H. \end{array} \right.$$

It is readily seen that a general D.C. problem defined as

$$\left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in D \\ & g_1(\boldsymbol{x}) - g_2(\boldsymbol{x}) \leqq 0, \end{array} \right.$$

where f_1, f_2, g_1 and g_2 are all convex functions on a closed convex set $D \subsetneqq \mathbb{R}^n$, is equivalent to

$$\left|\begin{array}{ll} \min\limits_{(\boldsymbol{x},z,s,t)} & z \\ \text{s.t.} & \boldsymbol{x} \in D,\ g_1(\boldsymbol{x}) \leqq s,\ f_1(\boldsymbol{x}) - z \leqq t \\ & g_2(\boldsymbol{x}) \geqq s,\ f_2(\boldsymbol{x}) \geqq t. \end{array}\right.$$

The problem is equivalent to

$$\left|\begin{array}{ll} \min\limits_{(\boldsymbol{x},z,s,t)} & z \\ \text{s.t.} & (\boldsymbol{x},z,s,t) \in S \\ & h(\boldsymbol{x},s,t) \geqq 0, \end{array}\right.$$

where $S = \{\, (\boldsymbol{x},z,s,t) \in \mathbb{R}^{n+3} \mid \boldsymbol{x} \in D,\ g_1(\boldsymbol{x}) \leqq s,\ f_1(\boldsymbol{x}) - z \leqq t \,\}$ and $h(\boldsymbol{x},s,t) = \max\{\, g_2(\boldsymbol{x}) - s, f_2(\boldsymbol{x}) - t \,\}$. Since S is a convex set and h is a convex function, the problem is (CDC). Then we can convert almost all D.C. problems to (CDC).

Definition 2.2 (Linear reverse convex problem) *A linear reverse convex problem, abbreviated to $(LRCP)$, is*

$$(LRCP) \qquad \left|\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{p}\boldsymbol{x} \\ s.t. & \boldsymbol{x} \in D \setminus int\, H, \end{array}\right.$$

where H is the same set as in (CDC) and

$$D = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0} \,\}, \tag{2.4}$$

with $A \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$.

Figure 2.2 shows two-dimensional examples of the problems (CDC) and $(LRCP)$, respectively. We see in the figures that the locally optimal solution $\bar{\boldsymbol{x}}$ fails to be global.

The inequality $h(\boldsymbol{x}) \geqq 0$ is said to be *essential* if

$$\min\{\, \boldsymbol{p}\boldsymbol{x} \mid \boldsymbol{x} \in D \,\} < \min\{\, \boldsymbol{p}\boldsymbol{x} \mid \boldsymbol{x} \in D \setminus \operatorname{int} H \,\}. \tag{2.5}$$

If the inequality $h(\boldsymbol{x}) \geqq 0$ is not essential then we can remove it in (CDC). However, it is very difficult to check if the inequality $h(\boldsymbol{x}) \geqq 0$ is essential. When we find a feasible solution $\boldsymbol{x}^* \in D \setminus \operatorname{int} H$ such that $\boldsymbol{p}\boldsymbol{x}^* = \min\{\, \boldsymbol{p}\boldsymbol{x} \mid \boldsymbol{x} \in D \,\}$, we recognize that $h(\boldsymbol{x}) \geqq 0$ is not essential. From the above observation, we do not assume that $h(\boldsymbol{x}) \geqq 0$ is essential.

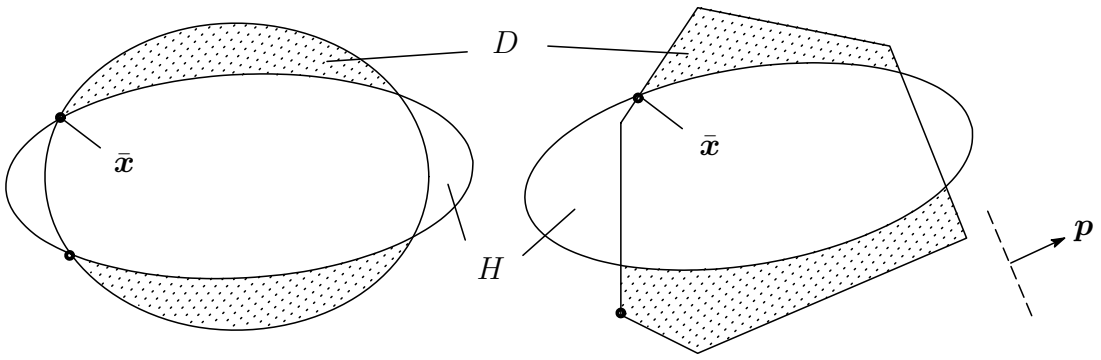


Figure 2.2: Problems (CDC) and $(LRCP)$

2.2 Cut-and-Split Method for $(LRCP)$

In this section, we explain the *cut-and-split (CS for short) method for* $(LRCP)$:

$$(LRCP) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{px} \\ \text{s.t.} & \boldsymbol{x} \in D \setminus \operatorname{int} H, \end{array} \right.$$

where D and H are given by (2.4) and (2.3).

Assumption 2.3 (Assumption for of the CS method) *We assume throughout this section that*

$$D \subsetneqq \mathbb{R}^n_+ \quad \textit{and} \quad \mathbf{0} \in D_V \cap \textit{int}\, H, \tag{2.6}$$

where D_V denotes the set of vertices of D.

In the first step of the CS method for $(LRCP)$, we find an initial feasible solution $\bar{\boldsymbol{x}} \in D \setminus \operatorname{int} H$, which serves as an initial incumbent, and set up the family $\mathcal{S}$ of polyhedral cones, initially $\mathcal{S} := \{\mathbb{R}^n_+\}$. In each iteration, we calculate a lower bound for the subproblem with respect to each cone $K \in \mathcal{S}$. We then split the cone whose lower bound is minimum until the optimality condition is met. To calculate the lower bound we define a concavity cut for $K \setminus H$ in the next subsection.

2.2.1 Concavity cut

A vector $\boldsymbol{r}$ is said to be a *ray* of the polyhedral cone K if it satisfies that $\boldsymbol{r} \neq \mathbf{0}$ and $\alpha \boldsymbol{r} \in K$ for all $\alpha > 0$. A ray $\boldsymbol{r}$ of K is said to be *extreme* if it is not a nonnegative

combination of two distinct rays of K. Given a polyhedral cone $K \subsetneqq \mathbb{R}^n$ with a vertex at $\mathbf{0}$ and exactly n extreme rays, let $\boldsymbol{u}^i$ denote the intersection point of ∂H and the ith extreme ray of K for $i = 1, \ldots, n$. The *concavity cut* $l_K(\boldsymbol{x}) \geqq 0$ *for* $K \setminus H$ is defined by the linear function $l_K : \mathbb{R}^n \to \mathbb{R}$ such that

$$l_K(\boldsymbol{x}) = \boldsymbol{e}U^{-1}\boldsymbol{x} - 1, \tag{2.7}$$

where $U = \begin{bmatrix} \boldsymbol{u}^1 & \boldsymbol{u}^1 & \cdots & \boldsymbol{u}^n \end{bmatrix}$. Clearly $l_K(\mathbf{0}) = -1$ and $l_K(\boldsymbol{u}^i) = 0$ for $i = 1, \ldots, n$. Figure 2.3 shows an example of the concavity cut. As seen in the figure, adding the concavity cut $l_K(\boldsymbol{x}) \geqq 0$ to K as a linear constraint does not cut any points in $K \setminus \operatorname{int} H$, since H is convex. Note that the assumption $\mathbf{0} \in \operatorname{int} H$ ensures that we can define the concavity cut for any cones.

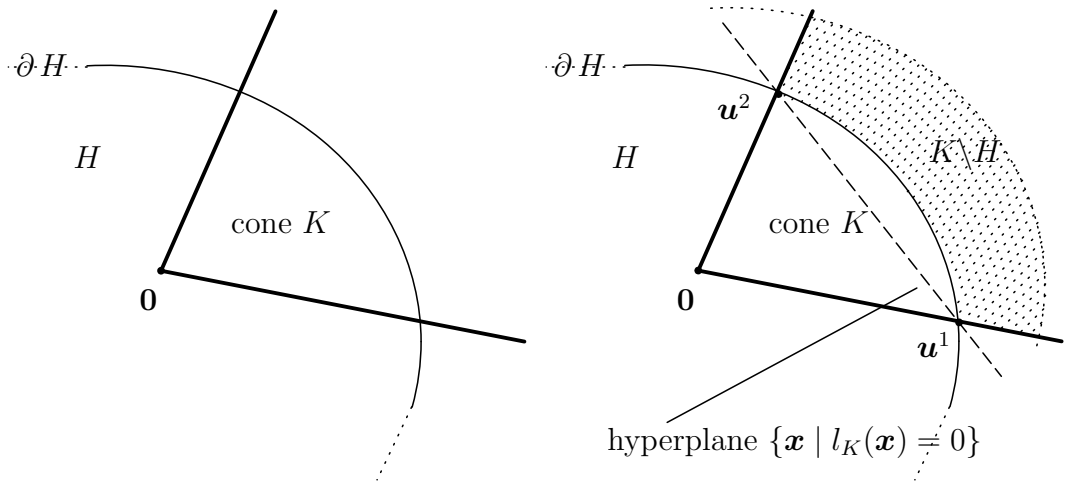


Figure 2.3: Concavity cut $l_K(\boldsymbol{x}) \geqq 0$ for $K \setminus H$

For each cone $K \in \mathcal{S}$, we solve the linear programming problem

$$(LP(K)) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{p}\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in D \cap K \\ & l_K(\boldsymbol{x}) \geqq 0, \end{array} \right.$$

to obtain a solution $\boldsymbol{x}^K$ with the optimal value $\beta_K = \boldsymbol{p}\boldsymbol{x}^K$, which is a lower bound for the subproblem

$$(SP(K)) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{p}\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in (D \setminus \operatorname{int} H) \cap K. \end{array} \right.$$

Note that since l_K is a linear function, the feasible set $\{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \in D \cap K,\ l_K(\boldsymbol{x}) \geqq 0 \,\}$ of $(LP(K))$ is a polyhedron. Let $\bar{\boldsymbol{x}} \in D \setminus \operatorname{int} H$ be an incumbent solution. If $h(\boldsymbol{x}^K) \geqq 0$, meaning $\boldsymbol{x}^K \in D \setminus \operatorname{int} H$, and $\beta_K < \boldsymbol{p}\bar{\boldsymbol{x}}$ for some K, then we update $\bar{\boldsymbol{x}}$ to $\boldsymbol{x}^K$. When there remain cones after discarding the cones K with $\boldsymbol{p}\boldsymbol{x}^K \geqq \boldsymbol{p}\bar{\boldsymbol{x}}$ if any, then we choose one of them and perform the $\boldsymbol{\omega}$-subdivision on the cone, which will be defined in the next subsection, and go to the next iteration.

2.2.2 Subdivision on cone

Let $Z \subsetneqq \mathbb{R}^n$ be a simplex with a vertex set $\{\boldsymbol{v}^1, \cdots, \boldsymbol{v}^n\}$ and let

$$\boldsymbol{\omega} = \sum_{j \in J} \theta_j \boldsymbol{v}^j, \quad \text{with} \quad \sum_{j \in J} \theta_j = 1 \text{ and } \theta_j > 0 \text{ for all } j \in J, \tag{2.8}$$

where $J \subsetneqq \{1, \ldots, n\}$ is the index set of at least two elements, i.e., $|J| \geqq 2$. Let Z_j denote the simplex with a vertex set $\{\boldsymbol{v}^1, \cdots, \boldsymbol{v}^n\} \setminus \{\boldsymbol{v}^j\} \cup \{\boldsymbol{\omega}\}$ for each $j \in J$. The simplex Z is then split into $|J|$ simplices Z_j. This splitting is called the *$\boldsymbol{\omega}$-subdivision on Z*. Furthermore when $\boldsymbol{\omega} = (1-\alpha)\boldsymbol{v} + \alpha \boldsymbol{v}'$ for $\alpha \in (0,1)$ and for $\boldsymbol{v}, \boldsymbol{v}' \in \{\boldsymbol{v}^1, \cdots, \boldsymbol{v}^n\}$ such that

$$\|\boldsymbol{v} - \boldsymbol{v}'\| = \max\{\, \|\boldsymbol{v}^i - \boldsymbol{v}^j\| \mid i \neq j,\ i = 1, \ldots, n,\ j = 1, \ldots, n \,\},$$

the subdivision is called the *bisection on Z of ratio α*. If $\alpha = 1/2$, we say that the bisection is *exact*. We next define a $\boldsymbol{\omega}$-subdivision on a polyhedral cone. Given a polyhedral cone $K \subsetneqq \mathbb{R}^n$ generated by n extreme rays, we construct a simplex $Z = K \cap L$ for a fixed hyperplane L intersecting all these rays. For some point $\boldsymbol{\omega} \in Z$, we perform the $\boldsymbol{\omega}$-subdivision on Z to obtain $|J|$ simplices Z_j. Let $\{\boldsymbol{v}^{(j,1)}, \cdots, \boldsymbol{v}^{(j,n)}\}$ be a vertex set of Z_j for each $j \in J$. The polyhedral cone K is then split into $|J|$ polyhedral cones K_j, each of which is generated by n extreme rays with directions $\boldsymbol{v}^{(j,1)}, \cdots, \boldsymbol{v}^{(j,n)}$. This splitting is called the *$\boldsymbol{\omega}$-subdivision on K*. An example of $\boldsymbol{\omega}$-subdivision on a polyhedral cone is shown in Figure 2.4, in which the cone K is splitted into three cones K_1, K_2 and K_3.

2.2.3 Algorithm and its convergence

The CS method for $(LRCP)$ is described as follows.

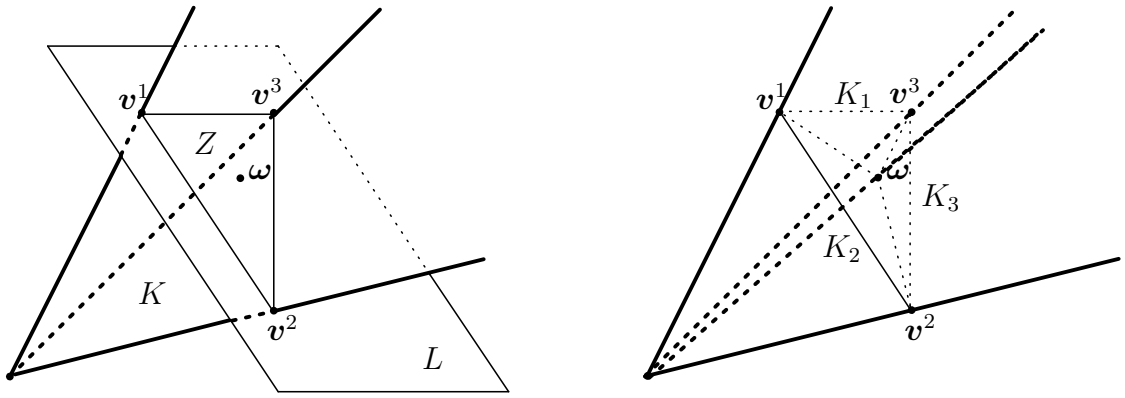


Figure 2.4: $\boldsymbol{\omega}$-subdivision on a polyhedral cone K

/** CS method for $(LRCP)$ **/

$\langle 0 \rangle$ (initialization) Find an initial feasible solution $\bar{\boldsymbol{x}} \in D \setminus \operatorname{int} H$ of $(LRCP)$. Set $K_0 := \mathbb{R}^n_+$, $\mathcal{S} := \{K_0\}$, $\mathcal{R} := \mathcal{S}$, and $k := 0$.

$\langle k \rangle$ (iteration k) For each $K \in \mathcal{S}$, solve $(LP(K))$ to obtain a solution $\boldsymbol{x}^K$ with the value $\beta_K := \boldsymbol{p}\boldsymbol{x}^K$.

$\langle k1 \rangle$ (update) If $\{\, K \in \mathcal{S} \mid h(\boldsymbol{x}^K) \geqq 0 \,\} \neq \emptyset$ then solve $\beta_{K^*} := \min\{\, \beta_K \mid K \in \mathcal{S},\ h(\boldsymbol{x}^K) \geqq 0 \,\}$ to obtain the cone K^*. If $\beta_{K^*} < \boldsymbol{p}\bar{\boldsymbol{x}}$ then set $\bar{\boldsymbol{x}} := \boldsymbol{x}^{K^*}$.

$\langle k2 \rangle$ (termination) Let $\mathcal{R}' := \{\, K \in \mathcal{R} \mid \beta_K < \boldsymbol{p}\bar{\boldsymbol{x}} \,\}$. If $\mathcal{R}' = \emptyset$ then stop ($\bar{\boldsymbol{x}}$ solves $(LRCP)$).

$\langle k3 \rangle$ (subdivision) Solve $\min\{\, \beta_K \mid K \in \mathcal{R}' \,\}$ to obtain the cone K^{**}. Perform the $\boldsymbol{\omega}^k$-subdivision on K^{**} for some $\boldsymbol{\omega}^k \in K^{**}$, and let $\mathcal{S}^{**}$ be the partition of K^{**}. Set $\mathcal{S} := \mathcal{S}^{**}$, $\mathcal{R} := \mathcal{S}^{**} \cup (\mathcal{R}' \setminus \{K^{**}\})$, $k := k + 1$ and go to $\langle k \rangle$.

The convergence of the CS method critically depends on the subdivision rule on K^{**}. A subdivision rule on a cone is said to be *exhaustive* if any nested sequence of cones generated by the algorithm will shrink to a single ray. It is known that if we set $\boldsymbol{\omega}^k := \boldsymbol{x}^{K^{**}}$ and the subdivision rule at Step $\langle k3 \rangle$ is exhaustive, there is at least one accumulation point $\boldsymbol{\omega}^*$ of $\{\boldsymbol{\omega}^k\}$ contained in ∂H, and hence $\boldsymbol{\omega}^*$ is feasible. Since $\boldsymbol{p}\boldsymbol{\omega}^k$ is a lower bound of $(LRCP)$ for all k, $\boldsymbol{\omega}^*$ is an optimal solution of $(LRCP)$. It is also known that if we perform only the bisection of fixed ratio α, the subdivision rule is exhaustive (See e.g., Tuy [62]).

2.3 Outer Approximation Method for (CDC)

In this section we explain an *outer approximation (OA for short) method for* (CDC):

$$(CDC) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{p}\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in D \setminus \operatorname{int} H, \end{array} \right.$$

where $D \subsetneqq \mathbb{R}^n$ is a convex set, H is given by (2.3).

2.3.1 Regularity and optimality condition

Assumption 2.4 (Assumption of the OA method) *We assume that D is bounded for simplicity and satisfies that*

$$\mathbf{0} \in D \cap int\, H \quad and \quad \min\{\, \boldsymbol{p}\boldsymbol{x} \mid \boldsymbol{x} \in D \,\} = 0, \tag{2.9}$$

and

$$D \setminus int\, H = cl\,(D \setminus H). \tag{2.10}$$

We say that (CDC) is *regular* if (2.10) holds. Figure 2.5 shows an example of (CDC) that is not regular, where $\boldsymbol{x}^* \in D \setminus \operatorname{int} H$, while $\boldsymbol{x}^* \notin \operatorname{cl}(D \setminus H)$.

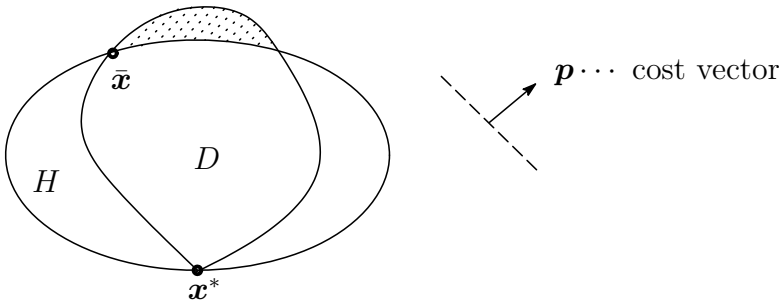


Figure 2.5: The case where (CDC) is not regular

The regularity assumption of (2.10) yields the optimality condition Theorem 2.5, which was given by Horst-Tuy [27]. To make this thesis self-contained, we will give an outline of the proof. In the followings we denote

$$D(\eta) = \{\, \boldsymbol{x} \in D \mid \boldsymbol{p}\boldsymbol{x} \leqq \eta \,\}, \tag{2.11}$$

for $\eta \in \mathbb{R}$.

Theorem 2.5 (Optimality condition of (CDC)) *Let $\bar{\boldsymbol{x}} \in D \setminus int\, H$ be a feasible solution of (CDC). If (CDC) is regular and*

$$D(\boldsymbol{p}\bar{\boldsymbol{x}}) \subseteqq H \tag{2.12}$$

then $\bar{\boldsymbol{x}}$ is an optimal solution.

Proof: Suppose that $\bar{\boldsymbol{x}} \in D \setminus \operatorname{int} H$ is not an optimal solution of (CDC), i.e., there exists $\boldsymbol{y} \in D \setminus \operatorname{int} H$ such that $\boldsymbol{py} < \boldsymbol{p}\bar{\boldsymbol{x}}$. Clearly, $\boldsymbol{y} \in D(\boldsymbol{p}\bar{\boldsymbol{x}})$ and $h(\boldsymbol{y}) \geqq 0$. If $h(\boldsymbol{y}) > 0$ then $\boldsymbol{y}$ is not contained in H, and hence $\boldsymbol{y} \in D(\boldsymbol{p}\bar{\boldsymbol{x}}) \setminus H$. By the regularity assumption, if $h(\boldsymbol{y}) = 0$, i.e., $\boldsymbol{y} \in \partial H$ then we can take $\boldsymbol{y}' \in N_\delta(\boldsymbol{y}) \cap D$ such that $\boldsymbol{py}' < \boldsymbol{p}\bar{\boldsymbol{x}}$ and $h(\boldsymbol{y}') > 0$ for a sufficiently small $\delta > 0$, and hence we see that $\boldsymbol{y}' \in D(\boldsymbol{p}\bar{\boldsymbol{x}}) \setminus H$. □

The above optimality condition is not valid unless (CDC) is regular. For instance, an optimal solution in Figure 2.5 is not $\bar{\boldsymbol{x}}$ but $\boldsymbol{x}^*$ while the inclusion $D(\boldsymbol{p}\bar{\boldsymbol{x}}) \subseteqq H$ is met.

2.3.2 Algorithm and its convergence

Let $\boldsymbol{x}^*$ be an optimal solution of (CDC) and $\bar{\boldsymbol{x}}^k \in D \setminus \operatorname{int} H$ be the incumbent at iteration k. In the OA method, we construct polytopes $P^0, P^1, \cdots, P^k, \cdots$ such that $P^0 \supseteqq P^1 \supseteqq \cdots \supseteqq P^k \supseteqq \cdots \supseteqq D(\boldsymbol{p}\boldsymbol{x}^*)$. If $\boldsymbol{p}\bar{\boldsymbol{x}}^k = 0$, we have done by (2.9). In the case where $\boldsymbol{p}\bar{\boldsymbol{x}}^k > 0$, we check the optimality condition $D(\boldsymbol{p}\bar{\boldsymbol{x}}^k) \subseteqq H$ by evaluating $h(\boldsymbol{v})$ at each vertex $\boldsymbol{v}$ of P^k. Namely, if $h(\boldsymbol{v}) \leqq 0$ for each vertex $\boldsymbol{v}$ of P^k, meaning $P^k \subseteqq H$, then $\bar{\boldsymbol{x}}^k$ solves (CDC). Otherwise we construct P^{k+1} by adding some linear inequality to P^k.

We describe the OA method for (CDC) as follows.

/** OA method for (CDC) **/

$\langle 0 \rangle$ (initialization) Find an initial feasible solution $\bar{\boldsymbol{x}}^0 \in D \setminus \operatorname{int} H$ of (CDC) and construct an initial polytope P^0 such that $P^0 \supseteqq D(\boldsymbol{p}\bar{\boldsymbol{x}}^0)$. Compute the vertex set P_V^0 of P^0 and set $k := 0$.

$\langle k \rangle$ (iteration k) Solve $\boldsymbol{v}^k \in \operatorname{argmax}\{\, h(\boldsymbol{v}) \mid \boldsymbol{v} \in P_V^k \,\}$.

$\langle k1\rangle$ (termination) If either $\boldsymbol{p}\bar{\boldsymbol{x}}^k = 0$ or $h(\boldsymbol{v}^k) \leqq 0$, meaning $P^k \subseteq H$, then stop ($\bar{\boldsymbol{x}}^k$ solves (CDC)). Otherwise, obtain the point $\boldsymbol{x}^k \in [\boldsymbol{0}, \boldsymbol{v}^k) \cap \partial H$.

$\langle k2\rangle$ (cutting the polytope) If $\boldsymbol{x}^k \notin D$ then set $\bar{\boldsymbol{x}}^{k+1} := \bar{\boldsymbol{x}}^k$ and $P^{k+1} := P^k \cap \{\, \boldsymbol{x} \in \mathbb{R}^n \mid l(\boldsymbol{x}) \leqq 0 \,\}$ for some affine function $l : \mathbb{R}^n \to \mathbb{R}$ such that $l(\boldsymbol{v}^k) > 0$ and $l(\boldsymbol{x}) \leqq 0$ for all $\boldsymbol{x} \in D(\boldsymbol{p}\bar{\boldsymbol{x}}^k)$. If $\boldsymbol{x}^k \in D$ then set $\bar{\boldsymbol{x}}^{k+1} := \boldsymbol{x}^k$ and $P^{k+1} := P^k \cap \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{p}\boldsymbol{x} \leqq \boldsymbol{p}\bar{\boldsymbol{x}}^{k+1} \,\}$.

$\langle k3\rangle$ Compute the vertex set P_V^{k+1} of P^{k+1}. Set $k := k + 1$ and go to $\langle k\rangle$.

Remark 2.6 *We will explain subroutines for computing the vertex set P_V^{k+1} from the knowledge of P_V^k in Section 2.4.*

Here we consider the case where D is given by $D = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leqq 0 \,\}$, where $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a convex function, and assume that

$$\min\{\, \boldsymbol{p}\boldsymbol{x} \mid \boldsymbol{x} \in D \setminus \operatorname{int} H \,\} > 0. \tag{2.13}$$

For the function f and a point $\bar{\boldsymbol{x}} \in \mathbb{R}^n$, the *subdifferential of f at $\bar{\boldsymbol{x}}$*, denoted by $\partial f(\bar{\boldsymbol{x}})$, is defined as

$$\partial f(\bar{\boldsymbol{x}}) = \{\, \boldsymbol{q} \in \mathbb{R}_n \mid f(\boldsymbol{x}) \geqq \boldsymbol{q}(\boldsymbol{x} - \bar{\boldsymbol{x}}) + f(\bar{\boldsymbol{x}}) \text{ for all } \boldsymbol{x} \in D \,\}. \tag{2.14}$$

The following theorem was shown in Tuy [62].

Theorem 2.7 (Basic outer approximation theorem) *Let $\{P^k\}_{k=0,1,\ldots}$ be a sequence of polytopes satisfying:*

(i) $D \subseteqq \cdots \subseteqq P^1 \subseteqq P^0$,

(ii) there is a point $\boldsymbol{v}^k \in P^k \setminus D$ for each k,

(iii) $P^{k+1} := P^k \cap \{\, \boldsymbol{x} \mid \boldsymbol{q}^k(\boldsymbol{x} - \boldsymbol{x}^k) + \alpha_k \leqq 0 \,\}$, *where* $\alpha_k \in [0, f(\boldsymbol{x}^k)]$ *and* $\boldsymbol{q}^k \in \partial f(\boldsymbol{x}^k)$ *for* $\boldsymbol{x}^k \in [\boldsymbol{0}, \boldsymbol{v}^k] \setminus \operatorname{int} D$ *such that* $\alpha_k - f(\boldsymbol{x}^k) \to 0$ *as* $k \to +\infty$.

Then any accumulation point $\boldsymbol{v}$ of the sequence $\{\boldsymbol{v}^k\}$ belongs to ∂D.

Let $\boldsymbol{v}^k$ be an optimal solution of $\max\{\, h(\boldsymbol{v}) \mid \boldsymbol{v} \in P_V^k \,\}$ and $\boldsymbol{x}^k \in [\boldsymbol{0}, \boldsymbol{v}^k] \cap \partial H$ at iteration k. If the algorithm iterates infinitely many times then any accumulation point $\boldsymbol{v}^*$ of the subsequence of $\{\boldsymbol{v}^k\}$, i.e, $\boldsymbol{v}^* = \lim_{\nu\to\infty} \boldsymbol{v}^{k_\nu}$ for some subsequence $\{\boldsymbol{v}^{k_\nu}\}$ of $\{\boldsymbol{v}^k\}$, belongs to $\partial D(\boldsymbol{p}\boldsymbol{x}^*)$ for $\boldsymbol{x}^* = \lim_{\nu\to\infty} \boldsymbol{x}^{k_\nu}$ by Theorem 2.7, and hence $\boldsymbol{v}^* = \boldsymbol{x}^*$ and $h(\boldsymbol{v}^*) = 0$. This implies $D(\boldsymbol{p}\boldsymbol{x}^*) \subseteqq H$.

2.4 On-line Vertex Enumeration Procedure, Double Description Method

In the OA method, adding a linear inequality to P^k yields P^{k+1} and the vertex set P_V^k of P^k is at hand. Enumerating all vertices of P^{k+1} is the key implementation issue. Subroutines for computing the vertex set P_V^{k+1} from the knowledge of P_V^k are provided in e.g., Horst-Vries-Thoai [25], Chen-Hansen-Jaumard [14], Subsection 7.4 of Padberg [39] and Chapter 18 of Chvátal [15]. Due to the possible degeneracy of P^k, a sophisticated implementation should be needed e.g., Fukuda-Prodon [21]. We explain in this section the *double description method* studied by Fukuda-Prodon [21]. The method is simple and of great use for enumerating all extreme rays of a polyhedral cone. Note that the method works for the polyhedral cone which is not full-dimensional.

Given matrices $A \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{n\times \ell}$, we define two sets

$$P(A) = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} \geqq \boldsymbol{0} \,\}, \text{ and} \tag{2.15}$$

$$P(R) = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = R\boldsymbol{\mu},\ \boldsymbol{\mu} \geqq \boldsymbol{0} \,\}. \tag{2.16}$$

Letting $\boldsymbol{a}^i$ be the ith row of A and $\boldsymbol{r}^j$ be the jth column of R, we write the above sets as

$$P(A) = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}^i\boldsymbol{x} \geqq 0 \text{ for } i = 1, \ldots, m \,\}, \text{ and}$$
$$P(R) = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = \textstyle\sum_{j=1}^{\ell} \mu_j \boldsymbol{r}^j,\ \mu_j \geqq 0 \text{ for } j = 1, \ldots, \ell \,\}.$$

As seen readily by the definition, $P(A)$ and $P(R)$ are the polyhedral cone of two different descriptions, namely, $P(A)$ is the polyhedral cone determined by m linear inequalities $\boldsymbol{a}^1\boldsymbol{x} \geqq 0, \cdots, \boldsymbol{a}^m\boldsymbol{x} \geqq 0$, and $P(R)$ is the polyhedral cone determined by the nonnegative linear combination of ℓ rays $\boldsymbol{r}^1, \cdots, \boldsymbol{r}^\ell$. The pair (A, R) is said to be a *double description pair* or a *D.D. pair* if $P(A) = P(R)$. Note that it is not easy to check if $P(A) = P(R)$ for given matrices $A \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{n\times \ell}$.

Suppose that we have a D.D. pair (A, R) at hand and we make the matrix

$$A' = \begin{bmatrix} & A & \\ & \boldsymbol{a} & \end{bmatrix}, \tag{2.17}$$

by adding some row vector $\boldsymbol{a} \in \mathbb{R}_n$ to A. The aim of this subsection is to explain how to obtain the matrix R' such that (A', R') is a D.D. pair.

Before pursuing the subject, we mention the homogenization. Let

$$P = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} \leqq \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0} \,\},$$

where $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, be a polytope and suppose that we have obtained the vertex set $P_V = \{\, \boldsymbol{v}^1, \cdots, \boldsymbol{v}^\ell \,\}$ at hand. Introducing a variable x_0 and letting

$$P' = \left\{ \begin{bmatrix} x_0 \\ \boldsymbol{x} \end{bmatrix} \in \mathbb{R}^{n+1} \;\middle|\; \begin{bmatrix} \boldsymbol{b} & -A \end{bmatrix} \begin{bmatrix} x_0 \\ \boldsymbol{x} \end{bmatrix} \geqq \boldsymbol{0},\ \begin{bmatrix} x_0 \\ \boldsymbol{x} \end{bmatrix} \geqq \boldsymbol{0} \right\},$$

we have

$$P = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix} \in P' \right\}.$$

Also we define

$$\boldsymbol{r}^j = \begin{bmatrix} 1 \\ \boldsymbol{v}^j \end{bmatrix},$$

for $j = 1, \ldots, \ell$. It is readily seen that $\boldsymbol{r}^1, \cdots, \boldsymbol{r}^\ell \in \mathbb{R}^{n+1}$ are the extreme rays of the polyhedral cone P'. Let R denote the $(n+1) \times \ell$ matrix whose jth column is $\boldsymbol{r}^j$. Meanwhile, if $\boldsymbol{r}^1, \cdots, \boldsymbol{r}^\ell$ are the extreme rays of P' then $\boldsymbol{v}^1, \cdots, \boldsymbol{v}^\ell$ are the vertex set of P, where $\boldsymbol{v}^j = (1/r_0^j)(r_1^j, \cdots, r_n^j)^\top$ for $\boldsymbol{r}^j = (r_0^j, \cdots, r_n^j)^\top$ and $j = 1, \ldots, \ell$. Constructing P' from P is called *homogenization* (See Figure 2.6). Therefore we can

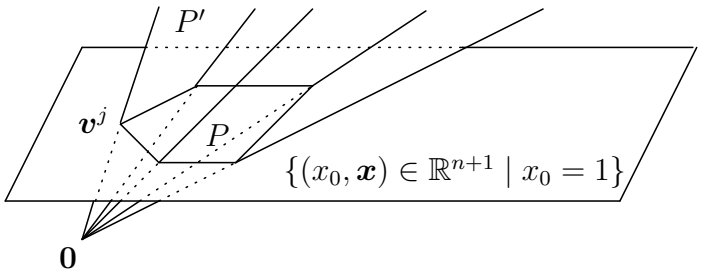


Figure 2.6: Homogenization

compute the vertex set P_V^{k+1} from the knowledge of P_V^k by using the homogenization and the double description method.

We go back to the aim of the subsection. Given a D.D. pair (A, R) and the matrix A' of (2.17), we need to obtain the matrix R' such that (A', R') is a D.D.

pair. In the sequel we assume that $P(A)$ is *pointed*, i.e., $\mathbf{0}$ is the vertex of $P(A)$, which is equivalent to that rank $A = n$. For the new row vector $\boldsymbol{a} \in \mathbb{R}_n$, let

$$\begin{aligned} J_+ &= \{\, j \in \{1, \ldots, \ell\} \mid \boldsymbol{a}\boldsymbol{r}^j > 0 \,\}, \\ J_0 &= \{\, j \in \{1, \ldots, \ell\} \mid \boldsymbol{a}\boldsymbol{r}^j = 0 \,\}, \text{ and} \\ J_- &= \{\, j \in \{1, \ldots, \ell\} \mid \boldsymbol{a}\boldsymbol{r}^j < 0 \,\}. \end{aligned} \tag{2.18}$$

We then have the following lemma.

Lemma 2.8 (Main lemma for double description method) *Let*

$$R' = \begin{bmatrix} R_1 & R_2 \end{bmatrix}, \quad R_1 = \left[\boldsymbol{r}^j\right]_{j \in J_+ \cup J_0} \text{ and } R_2 = \left[\boldsymbol{r}^{jj'}\right]_{(j,j') \in J_+ \times J_-},$$

where

$$\boldsymbol{r}^{jj'} = (\boldsymbol{a}\boldsymbol{r}^j)\boldsymbol{r}^{j'} + (-\boldsymbol{a}\boldsymbol{r}^{j'})\boldsymbol{r}^j, \tag{2.19}$$

for each $(j, j') \in J_+ \times J_-$. *Then* (A', R') *is a D.D. pair.*

Proof: We prove that $P(R') = P(A')$.

($\subseteqq$) Suppose $\boldsymbol{r}$ is a column of R'. If $\boldsymbol{r} = \boldsymbol{r}^j$ for some $j \in J_+ \cup J_0$, then $\boldsymbol{r}$ is a column of R and $\boldsymbol{a}\boldsymbol{r} \geqq 0$. Since (A, R) is a D.D. pair, we see that $\boldsymbol{r} \in P(R) = P(A)$, and hence $\boldsymbol{r} \in P(A')$. If $\boldsymbol{r} = \boldsymbol{r}^{jj'}$ for some $(j, j') \in J_+ \times J_-$ then by (2.19) we have $A\boldsymbol{r} = (\boldsymbol{a}\boldsymbol{r}^j)A\boldsymbol{r}^{j'} + (-\boldsymbol{a}\boldsymbol{r}^{j'})A\boldsymbol{r}^j \geqq \mathbf{0}$ and $\boldsymbol{a}\boldsymbol{r} = (\boldsymbol{a}\boldsymbol{r}^j)\boldsymbol{a}\boldsymbol{r}^{j'} + (-\boldsymbol{a}\boldsymbol{r}^{j'})\boldsymbol{a}\boldsymbol{r}^j = 0$, and hence $\boldsymbol{r} \in P(A')$. Since all columns of R' are contained in $P(A')$, we have $P(R') \subseteqq P(A')$.

($\supseteqq$) Suppose that $\boldsymbol{x}$ is contained in $P(A')$, i.e, $A\boldsymbol{x} \geqq \mathbf{0}$, $\boldsymbol{a}\boldsymbol{x} \geqq 0$ and $\boldsymbol{x} \geqq \mathbf{0}$. Since (A, R) is a D.D. pair we have $\boldsymbol{x} \in P(A) = P(R)$, and hence there are nonnegative coefficients $\mu_1, \cdots, \mu_\ell$ such that $\boldsymbol{x} = \sum_{j=1}^{\ell} \mu_j \boldsymbol{r}^j = \sum_{j \in J_+} \mu_j \boldsymbol{r}^j + \sum_{j \in J_0} \mu_j \boldsymbol{r}^j + \sum_{j \in J_-} \mu_j \boldsymbol{r}^j$. If $\mu_j = 0$ for all $j \in J_-$ then we have $\boldsymbol{x} \in P(R')$. If there is an index $k \in J_-$ such that $\mu_k > 0$ then we delete $\mu_k \boldsymbol{r}^k$ by the following operation. Letting

$$J_+(\boldsymbol{x}) = \{\, h \in J_+ \mid \mu_h > 0 \,\}, \tag{2.20}$$

we have

$$\boldsymbol{x} = \sum_{j \in J_+(\boldsymbol{x})} \mu_j \boldsymbol{r}^j + \sum_{j \in J_0} \mu_j \boldsymbol{r}^j + \sum_{j \in J_-} \mu_j \boldsymbol{r}^j. \tag{2.21}$$

Note that $J_+(\boldsymbol{x}) \neq \emptyset$ since $0 \leqq \boldsymbol{a}\boldsymbol{x} = \sum_{h \in J_+(\boldsymbol{x})} \mu_h(\boldsymbol{a}\boldsymbol{r}^h) + \sum_{j \in J_- \setminus \{k\}} \mu_j(\boldsymbol{a}\boldsymbol{r}^j) + \mu_k(\boldsymbol{a}\boldsymbol{r}^k)$, and $\sum_{j \in J_- \setminus \{k\}} \mu_j(\boldsymbol{a}\boldsymbol{r}^j) \leqq 0$ and $\mu_k(\boldsymbol{a}\boldsymbol{r}^k) < 0$. It is readily seen that

$$\frac{\mu_h(\boldsymbol{a}\boldsymbol{r}^h)}{\mu_k(-\boldsymbol{a}\boldsymbol{r}^k)} > 0, \tag{2.22}$$

for all $h \in J_+(\boldsymbol{x})$. Furthermore we see that

$$\sum_{h \in J_+(\boldsymbol{x})} \frac{\mu_h(\boldsymbol{a}\boldsymbol{r}^h)}{\mu_k(-\boldsymbol{a}\boldsymbol{r}^k)} \geqq 1, \tag{2.23}$$

since $0 \leqq \boldsymbol{a}\boldsymbol{x} \leqq \sum_{h \in J_+(\boldsymbol{x})} \mu_h(\boldsymbol{a}\boldsymbol{r}^h) + \mu_k(\boldsymbol{a}\boldsymbol{r}^k)$. Then there are nonnegative coefficients $\{\theta_h\}_{h \in J_+(\boldsymbol{x})}$ such that

$$0 \leqq \theta_h \leqq \frac{\mu_h(\boldsymbol{a}\boldsymbol{r}^h)}{\mu_k(-\boldsymbol{a}\boldsymbol{r}^k)} \text{ and } \sum_{h \in J_+(\boldsymbol{x})} \theta_h = 1. \tag{2.24}$$

Here for each $h \in J_+(\boldsymbol{x})$ defining

$$\boldsymbol{r}^{hk} = (\boldsymbol{a}\boldsymbol{r}^h)\boldsymbol{r}^k + (-\boldsymbol{a}\boldsymbol{r}^k)\boldsymbol{r}^h, \tag{2.25}$$

we see that $\theta_h \boldsymbol{r}^k = \theta_h \left(-\frac{(-\boldsymbol{a}\boldsymbol{r}^k)}{\boldsymbol{a}\boldsymbol{r}^h}\boldsymbol{r}^h + \frac{1}{\boldsymbol{a}\boldsymbol{r}^h}\boldsymbol{r}^{hk} \right)$, and hence

$$\boldsymbol{r}^k = \sum_{h \in J_+(\boldsymbol{x})} \theta_h \boldsymbol{r}^k = \sum_{h \in J_+(\boldsymbol{x})} \left(-\frac{(-\boldsymbol{a}\boldsymbol{r}^k)}{\boldsymbol{a}\boldsymbol{r}^h}\boldsymbol{r}^h + \frac{1}{\boldsymbol{a}\boldsymbol{r}^h}\boldsymbol{r}^{hk} \right). \tag{2.26}$$

Then we have

$$\begin{aligned} \boldsymbol{x} &= \sum_{j \in J_+(\boldsymbol{x})} \mu_j \boldsymbol{r}^j + \sum_{j \in J_0 \cup J_- \setminus \{k\}} \mu_j \boldsymbol{r}^j + \mu_k \boldsymbol{r}^k \\ &= \sum_{h \in J_+(\boldsymbol{x})} \left(\mu_h - \theta_h \frac{\mu_k(-\boldsymbol{a}\boldsymbol{r}^k)}{\boldsymbol{a}\boldsymbol{r}^h} \right) \boldsymbol{r}^h + \sum_{j \in J_0 \cup J_- \setminus \{k\}} \mu_j \boldsymbol{r}^j + \sum_{h \in J_+(\boldsymbol{x})} \frac{\theta_h \mu_k}{\boldsymbol{a}\boldsymbol{r}^h} \boldsymbol{r}^{hk}. \end{aligned} \tag{2.27}$$

Repeating the above operation until there is no index $k \in J_-$ such that $\mu_k > 0$, we see that $\boldsymbol{x} \in P(R')$. □

The matrix R' obtained by Lemma 2.8 contains many unnecessary columns. Namely there are columns of R' that are not extreme. As seen in Figure 2.7, we generate unnecessary rays $\boldsymbol{r}^{6,4}$, $\boldsymbol{r}^{1,4}$, $\boldsymbol{r}^{2,4}$ and so on, while extreme rays to be obtained are only $\boldsymbol{r}^{2,3}$ and $\boldsymbol{r}^{6,5}$. We should avoid generating such rays by checking the adjacency. Here we say that two distinct rays of P are *adjacent* if the minimal face of P containing both of two rays contains no other rays. Note that since P is polyhedral cone, new ray $\boldsymbol{r}^{jj'}$ becomes an extreme ray only when $\boldsymbol{r}^j$ and $\boldsymbol{r}^{j'}$ are adjacent.

Here we describe the procedure of obtaining R' such that (A', R') is a D.D. pair.

/** procedure of obtaining R' **/

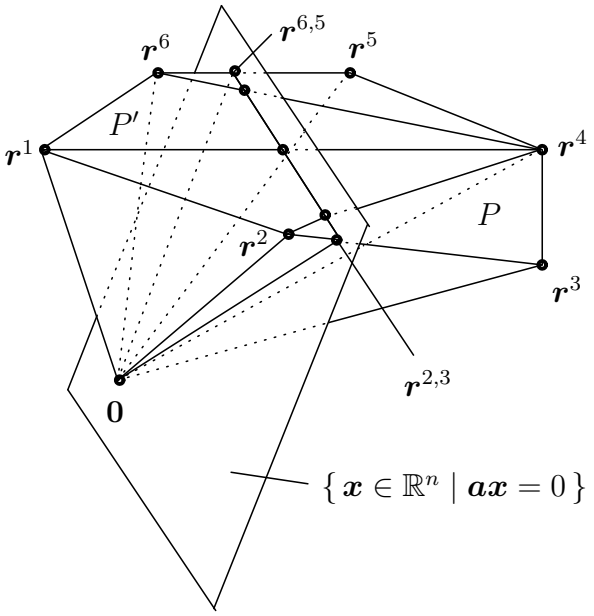


Figure 2.7: Unnecessary rays

⟨0⟩ (initialization) Let J_+, J_0, J_- be defined by (2.18) and set $R' := \left[\boldsymbol{r}^j\right]_{j \in J_+ \cup J_0}$.

⟨1⟩ (computing $\boldsymbol{r}^{jj'}$) For each $(j, j') \in J_+ \times J_-$, if $\boldsymbol{r}^j$ and $\boldsymbol{r}^{j'}$ are adjacent then compute $\boldsymbol{r}^{jj'}$ of (2.19) and set $R' := \left[R' \quad \boldsymbol{r}^{jj'}\right]$.

Further we denote the *active set of* $\boldsymbol{x}$ by $I(\boldsymbol{x})$, which is defined as

$$I(\boldsymbol{x}) = \{\, i \in \{1, \ldots, m\} \mid \boldsymbol{a}^i \boldsymbol{x} = 0 \,\}. \tag{2.28}$$

For an index set $S \subseteqq \{1, \ldots, m\}$, let A_S denote the matrix whose column is $\boldsymbol{a}^i$ $(i \in S)$. To check the adjacency, we have the following lemmas whose proofs can be found in Fukuda-Prodon [21].

Lemma 2.9 *Let $\boldsymbol{r}$ be a ray of P, $\bar{F} = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid A_{I(\boldsymbol{r})}\boldsymbol{x} = \boldsymbol{0} \,\}$, $F = \bar{F} \cap P$ and $k = n - rank(A_{I(\boldsymbol{r})})$. Then*

(a) $rank(A_{I(\boldsymbol{r})\cup\{i\}}) = n - k + 1$ for all $i \notin I(\boldsymbol{r})$.

(b) F contains k linearly independent rays.

(c) When $k \geqq 2$, $\boldsymbol{r}$ is a nonnegative linear combination of two distinct rays $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$ such that $rank(A_{I(\boldsymbol{r}^i)}) > n - k$ for $i = 1, 2$.

Lemma 2.10 *Let $\boldsymbol{r}$ be a ray of P. Then*

(a) $\boldsymbol{r}$ is an extreme ray of P if and only if rank$(A_{I(\boldsymbol{r})}) = n - 1$.

(b) $\boldsymbol{r}$ is a nonnegative combination of extreme rays of P.

When two rays $\boldsymbol{r}$ and $\boldsymbol{r}'$ satisfy $\boldsymbol{r} = \alpha\boldsymbol{r}'$ for some positive α, we denote by $\boldsymbol{r} \simeq \boldsymbol{r}'$. The above lemmas imply the following theorem which is useful of checking the adjacency.

Theorem 2.11 *Let $\boldsymbol{r}$ and $\boldsymbol{r}'$ be two distinct rays of P. Then the following statements are equivalent.*

(a) Rays $\boldsymbol{r}$ and $\boldsymbol{r}'$ are adjacent extreme rays.

(b) Rays $\boldsymbol{r}$ and $\boldsymbol{r}'$ are extreme rays and rank$(A_{I(\boldsymbol{r})\cap I(\boldsymbol{r}')}) = n - 2$.

(c) Either $\boldsymbol{r}'' \simeq \boldsymbol{r}$ or $\boldsymbol{r}'' \simeq \boldsymbol{r}'$ holds for any ray $\boldsymbol{r}''$ such that $I(\boldsymbol{r}) \cap I(\boldsymbol{r}') \subsetneqq I(\boldsymbol{r}'')$.

In our study, for each $(j, j') \in J_+ \times J_-$, we check the adjacency between $\boldsymbol{r}^j$ and $\boldsymbol{r}^{j'}$ by evaluating rank $(A_{I(\boldsymbol{r}^j)\cap I(\boldsymbol{r}^{j'})})$. When this rank equals $n-2$, meaning $\boldsymbol{r}^j$ and $\boldsymbol{r}^{j'}$ are adjacent, we then compute $\boldsymbol{r}^{jj'}$ of (2.19) in the procedure for obtaining R'. For further implementation techniques about double description method, the reader should refer to Fukuda-Prodon [21].

Chapter 3

Optimization over the Efficient Set

Over the past several decades, the optimization over the efficient set has seen a substantial development. The aim of this chapter is to provide a state-of-the-art survey of the development. Given p linear criteria $\boldsymbol{c}^1\boldsymbol{x}, \cdots, \boldsymbol{c}^p\boldsymbol{x}$ and the set of feasible solutions X of $\mathbb{R}^n$, the linear multicriteria problem is to find a point $\boldsymbol{x}$ of X such that no point $\boldsymbol{x}'$ of X satisfies $(\boldsymbol{c}^1\boldsymbol{x}', \cdots, \boldsymbol{c}^p\boldsymbol{x}') \geqq (\boldsymbol{c}^1\boldsymbol{x}, \cdots, \boldsymbol{c}^p\boldsymbol{x})$ and $(\boldsymbol{c}^1\boldsymbol{x}', \cdots, \boldsymbol{c}^p\boldsymbol{x}') \neq (\boldsymbol{c}^1\boldsymbol{x}, \cdots, \boldsymbol{c}^p\boldsymbol{x})$. Such a point is called an *efficient point*. The *optimization over the efficient set* is the maximization of a given function ϕ over the set of efficient points. Precise definitions of the efficient point and others will be given in the next section.

The difficulty of this problem is mainly due to the nonconvexity of this set. The existing algorithms for solving this problem could be classified into several groups such as adjacent vertex search algorithm, nonadjacent vertex search algorithm, branch-and-bound based algorithm, bisection algorithm, Lagrangian relaxation based approach, dual approach and D.C. approach. In this chapter we review a typical algorithm from each group and compare them from the computational point of view. We survey the existing algorithms for (P_E) as well as some variations. We will not discuss the merits and demerits of the algorithms because we have not yet had enough computational experience to evaluate them. Theoretically interesting algorithms do not always work efficiently, on the contrary, naive methods can surpass sophisticated algorithms in computation time. We should be careful not to nip the promising algorithms in the bud.

First we define the optimization problem over the efficient set. After reviewing

the well-known facts concerning the problem in Section 3.2, the adjacent vertex search algorithm and the nonadjacent vertex search algorithm will be explained in Section 3.3 and Section 3.4. Section 3.5 is devoted to the branch-and-bound method based on the conical partition, and Section 3.6 to the bisection algorithm. In Section 3.7 we introduce the face search algorithm, which is based on the enumeration of faces that constitute the efficient set. The Lagrangian relaxation based approach and the dual approach will be explained in Section 3.8 and Section 3.9. Applying the D.C. algorithm is explained in Section 3.10. Some other methods are discussed in Section 3.11. Finally, extending the problem to a nonlinear problem is discussed in Section 3.12.

3.1 Optimization Problem over the Efficient Set

Given a polyhedral set

$$X = \{ \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0} \}, \tag{3.1}$$

with $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, and a criterion matrix $C \in \mathbb{R}^{p \times n}$ with $p \geqq 2$, the *multicriteria problem*, abbreviated to (MC), is

$$(MC) \quad \left| \begin{array}{ll} \text{Vector Max} & C\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in X. \end{array} \right.$$

To avoid the technicality we assume throughout the chapter that X is nonempty and bounded.

A point $\boldsymbol{x} \in X$ is said to be an *efficient point for* (MC) if there is no point $\boldsymbol{x}' \in X$ such that $C\boldsymbol{x}' \geqq C\boldsymbol{x}$ and $C\boldsymbol{x}' \neq C\boldsymbol{x}$. We denote the set of efficient points for (MC) by X_E, i.e.,

$$X_E = \{ \boldsymbol{x} \in X \mid \nexists \boldsymbol{x}' \in X : C\boldsymbol{x}' \geqq C\boldsymbol{x},\ C\boldsymbol{x}' \neq C\boldsymbol{x} \}. \tag{3.2}$$

A point $\boldsymbol{x} \in X$ is said to be a *weakly efficient point for* (MC) if there is no point $\boldsymbol{x}' \in X$ such that $C\boldsymbol{x}' > C\boldsymbol{x}$. We denote the set of weakly efficient points for (MC) by X_W, i.e.,

$$X_W = \{ \boldsymbol{x} \in X \mid \nexists \boldsymbol{x}' \in X : C\boldsymbol{x}' > C\boldsymbol{x} \}. \tag{3.3}$$

The problems we consider in this chapter are the followings.

Definition 3.1 (Optimization problem over the efficient set) *The optimization problem over the efficient set, denoted by* (P_E)*, is*

$$(P_E) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ s.t. & \boldsymbol{x} \in X_E, \end{array} \right.$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ *is a continuous function to be maximized.*

Definition 3.2 (Optimization problem over the weakly efficient set) *The optimization problem over the weakly efficient set, denoted by* (P_W)*, is*

$$(P_W) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ s.t. & \boldsymbol{x} \in X_W. \end{array} \right.$$

For these problems we write $\phi(P_E)$ and $\phi(P_W)$ to denote their optimal values, respectively.

Figure 3.1 shows a two-dimensional example of the problem (P_E), where $\boldsymbol{c}^i$ is the ith row of C for $i = 1, 2$ and $\phi(\boldsymbol{x}) = \boldsymbol{d}\boldsymbol{x}$ for a given direction $\boldsymbol{d} \in \mathbb{R}_n$. The efficient set X_E is depicted by bold lines. As seen in Figure 3.1 X_E is nonconvex, and the locally optimal solution $\boldsymbol{x}^2$ fails to be a globally optimal solution.

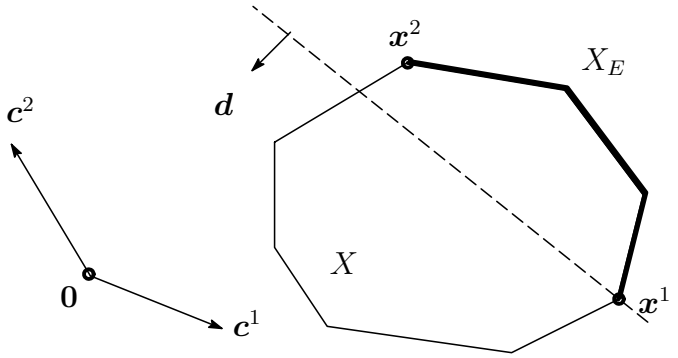


Figure 3.1: Problem (P_E)

The nonconvexity of the efficient set X_E often causes a multiextremality of (P_E). The main difficulty of the problem arises from the multiextremality. A three-dimensional example of (P_E) is shown in Figure 3.2. As seen in this figure, it is very difficult to solve (P_E) even if the dimension of X is relatively low. Also we see

in Figure 3.1 and Figure 3.2 that X_E is the connected union of several faces of X, which will be proved in the next section.

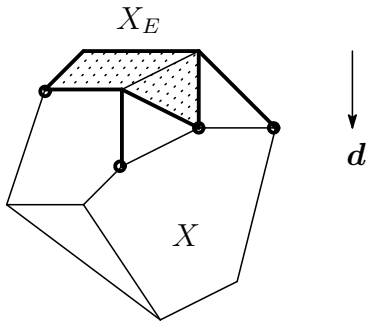


Figure 3.2: A three-dimensional example of (P_E)

3.2 Basic Results of Efficient Set X_E

3.2.1 Gap function

We define the *gap function*, denoted by $g : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty\}$, as

$$g(\boldsymbol{x}) = \max\{\boldsymbol{e}C\boldsymbol{x}' \mid \boldsymbol{x}' \in X,\ C\boldsymbol{x}' \geqq C\boldsymbol{x}\} - \boldsymbol{e}C\boldsymbol{x}. \tag{3.4}$$

Note that $g(\boldsymbol{x}) = -\infty$ if there is no point $\boldsymbol{x}' \in X$ such that $C\boldsymbol{x}' \geqq C\boldsymbol{x}$. An example of the gap function value at $\bar{\boldsymbol{x}}$ is shown in Figure 3.3, in which $\boldsymbol{x}^*$ is an optimal solution of $\max\{\boldsymbol{e}C\boldsymbol{x}' \mid \boldsymbol{x}' \in X,\ C\boldsymbol{x}' \geqq C\boldsymbol{x}\}$, and hence $g(\bar{\boldsymbol{x}}) = \boldsymbol{e}C\boldsymbol{x}^* - \boldsymbol{e}C\bar{\boldsymbol{x}}$. As seen in the figure, clearly $g(\boldsymbol{x}) \geqq 0$ for all $\boldsymbol{x} \in X$.

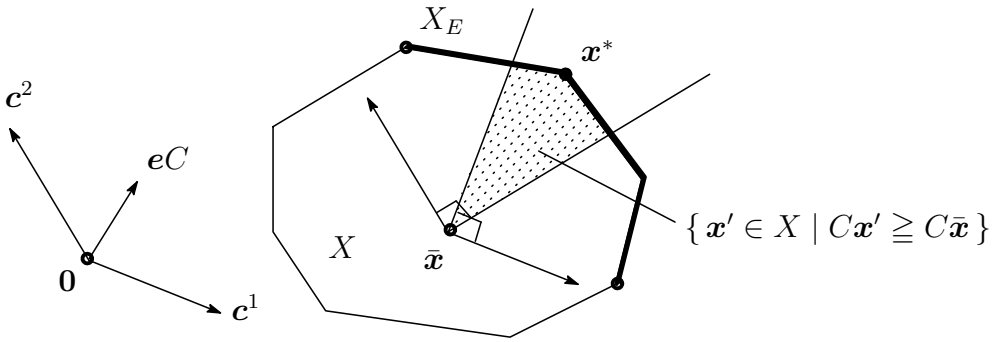


Figure 3.3: Gap function value at $\bar{\boldsymbol{x}}$

By the definition of the efficient point, if $\boldsymbol{x} \in X_E$ then $\boldsymbol{x}' \in X$ and $C\boldsymbol{x}' \geqq C\boldsymbol{x}$ imply $C\boldsymbol{x}' = C\boldsymbol{x}$. Hence it is readily seen that $X_E = \{\boldsymbol{x} \in X \mid g(\boldsymbol{x}) = 0\}$. Also it is well known that g is a concave and piece-wise linear function (See Figure 3.4).

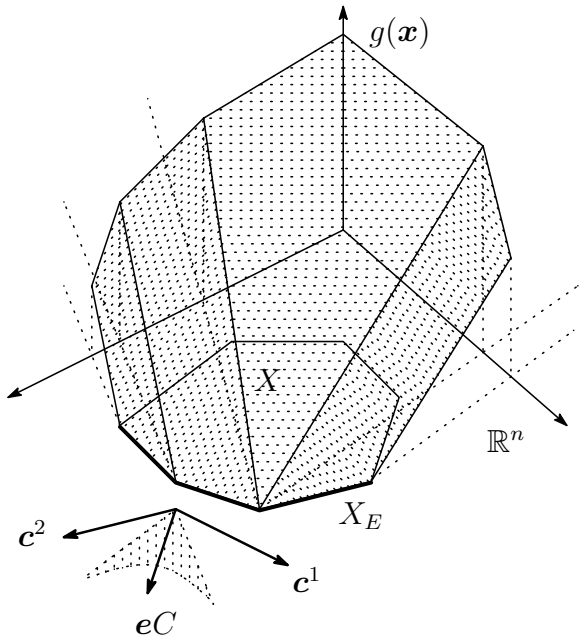


Figure 3.4: Gap function g and set X_E

Using the gap function g of (3.4), we can rewrite (P_E) as

$$(P_E) \quad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in X \\ & g(\boldsymbol{x}) = 0. \end{array} \right.$$

Since $g(\boldsymbol{x}) \geqq 0$ for all $\boldsymbol{x} \in X$, the last equality constraint $g(\boldsymbol{x}) = 0$ can be replaced by $g(\boldsymbol{x}) \leqq 0$, which yields

$$(P_E) \quad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in X \\ & g(\boldsymbol{x}) \leqq 0. \end{array} \right.$$

Since g is a concave function, the inequality constraint $g(\boldsymbol{x}) \leqq 0$ is the reverse convex constraint defined in Subsection 2.1.2, and hence (P_E) is $(LRCP)$ when $D = X$, $\boldsymbol{p} = -\boldsymbol{d}$ and $h = g$. Moreover, it is known that the above alternative form of (P_E) can be cast into the problem

$$(P_E(t)) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) - tg(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in X, \end{array} \right.$$

where $t > 0$ is an exact penalty parameter. Note that a tight exact penalty parameter for $\phi(\boldsymbol{x}) = -\boldsymbol{d}\boldsymbol{x}$ is given in e.g., An-Tao-Muu [3] and Dauer-Fosnaugh [16].

3.2.2 Gap function with direction $\boldsymbol{\lambda}$

We introduce several well-known results, whose proof can be found in e.g., Benson [10], Sawaragi-Nakayama-Tanino [44], Steuer [49], and White [63]. We will outline some of the proofs to make this thesis self-contained.

Lemma 3.3 *The point $\bar{\boldsymbol{x}} \in X$ is an efficient point for (MC) if and only if there exists $\boldsymbol{\lambda} \in \mathbb{R}_{p++}$ such that $\bar{\boldsymbol{x}}$ is an optimal solution of the single criterion problem*

$$(SC(\boldsymbol{\lambda})) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \boldsymbol{\lambda} C \boldsymbol{x} \\ s.t. & \boldsymbol{x} \in X. \end{array} \right.$$

Proof: ($\Leftarrow$) Assume that $\bar{\boldsymbol{x}} \in X$ is not an efficient point for (MC). There exists $\boldsymbol{y} \in X$ such that $C\boldsymbol{y} \geqq C\boldsymbol{x}$ and $C\boldsymbol{y} \neq C\boldsymbol{x}$. Then, $\bar{\boldsymbol{x}}$ is not an optimal solution of $(SC(\boldsymbol{\lambda}))$ for any $\boldsymbol{\lambda} \in \mathbb{R}_{p++}$.

($\Rightarrow$) Suppose $\bar{\boldsymbol{x}} \in X$ is an efficient point for (MC). Let $L_{\bar{\boldsymbol{x}}} = \mathrm{diag}\{l_1, \ldots, l_n\}$, where

$$l_i = \begin{cases} 1 & \text{if } \bar{\boldsymbol{x}}_i = 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 1, \ldots n.$$

If there exists a vector $\boldsymbol{u} \in \mathbb{R}^n$ satisfying the system

$$C\boldsymbol{u} \geqq \boldsymbol{0},\ C\boldsymbol{u} \neq \boldsymbol{0},\ L_{\bar{\boldsymbol{x}}}\boldsymbol{u} \geqq \boldsymbol{0},\ A\boldsymbol{u} = \boldsymbol{0}, \tag{3.5}$$

setting $\boldsymbol{x} = \bar{\boldsymbol{x}} + \theta\boldsymbol{u}$ for a sufficiently small $\theta > 0$, we see $\boldsymbol{x} \in X$ satisfies $C\boldsymbol{x} \geqq C\bar{\boldsymbol{x}}$ and $C\boldsymbol{x} \neq C\bar{\boldsymbol{x}}$. This contradicts that $\bar{\boldsymbol{x}}$ is an efficient point for (MC). Then, there is no vector $\boldsymbol{u} \in \mathbb{R}^n$ satisfies the system (3.5). Applying the Tucker's alternative theorem (See Mangasarian [34]), there are vectors $\boldsymbol{\lambda} \in \mathbb{R}_p$, $\boldsymbol{\mu} \in \mathbb{R}_n$ and $\boldsymbol{\nu} \in \mathbb{R}_m$ such that

$$\boldsymbol{\lambda} C + \boldsymbol{\mu} L_{\bar{\boldsymbol{x}}} + \boldsymbol{\nu} A = \boldsymbol{0},\ \boldsymbol{\lambda} > \boldsymbol{0},\ \boldsymbol{\mu} \geqq \boldsymbol{0}.$$

For any $\boldsymbol{x} \in \mathbb{R}^n$, we see that

$$\boldsymbol{\lambda} C(\bar{\boldsymbol{x}} - \boldsymbol{x}) + \boldsymbol{\mu} L_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}} - \boldsymbol{x}) + \boldsymbol{\nu} A(\bar{\boldsymbol{x}} - \boldsymbol{x}) = 0.$$

Therefore for any $\boldsymbol{x} \in X$, we have $\boldsymbol{\mu} L_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}} - \boldsymbol{x}) \leqq 0$ and $\boldsymbol{\nu} A(\bar{\boldsymbol{x}} - \boldsymbol{x}) = 0$, and hence $\boldsymbol{\lambda} C\bar{\boldsymbol{x}} \geqq \boldsymbol{\lambda} C\boldsymbol{x}$. □

For $\boldsymbol{\lambda} \in \mathbb{R}_{p++}$, we defined the *gap function with direction* $\boldsymbol{\lambda}$, denoted by $g_{\boldsymbol{\lambda}} : \mathbb{R}^n \to \mathbb{R}$, as

$$g_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \max\{\, \boldsymbol{\lambda} C\boldsymbol{x}' \mid \boldsymbol{x}' \in X \,\} - \boldsymbol{\lambda} C\boldsymbol{x}. \tag{3.6}$$

As can be seen readily, $\boldsymbol{x} \in X$ is in X_E if and only if there is $\boldsymbol{\lambda} \in \mathbb{R}_{p++}$ such that $g_{\boldsymbol{\lambda}}(\boldsymbol{x}) = 0$, and a point $\boldsymbol{x}'$ which solves $\max\{\, \boldsymbol{\lambda} C\boldsymbol{x}' \mid \boldsymbol{x}' \in X \,\}$ is in X_E. Hence we have $X_E = \{\, \boldsymbol{x} \in X \mid \exists \boldsymbol{\lambda} \in \mathbb{R}_{p++} : g_{\boldsymbol{\lambda}}(\boldsymbol{x}) = 0 \,\}$. The theory of parametric linear program shows that $g_{\boldsymbol{\lambda}}$ is a piece-wise linear concave function. Furthermore the following property is also readily seen.

Lemma 3.4 *If $C\boldsymbol{x} = C\boldsymbol{x}'$ then $g_{\boldsymbol{\lambda}}(\boldsymbol{x}) = g_{\boldsymbol{\lambda}}(\boldsymbol{x}')$.*

The following theorem implied by Lemma 3.3 is also important.

Theorem 3.5

$$X_E = \{\, \boldsymbol{x} \in X \mid \exists \boldsymbol{\lambda} \in \mathbb{R}_{p++} : \boldsymbol{\lambda} C\boldsymbol{x} \geqq \boldsymbol{\lambda} C\boldsymbol{x}' \text{ for all } \boldsymbol{x}' \in X \,\} \tag{3.7}$$

$$= \left\{ \boldsymbol{x} \in X \;\middle|\; \begin{array}{l} \exists (\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathbb{R}_{p++} \times \mathbb{R}_m \times \mathbb{R}_{n+} : \\ \boldsymbol{\lambda} C - \boldsymbol{\mu} A + \boldsymbol{\nu} = \boldsymbol{0},\ \boldsymbol{\nu}\boldsymbol{x} = 0 \end{array} \right\} \tag{3.8}$$

$$= \left\{ \boldsymbol{x} \in X \;\middle|\; \begin{array}{l} \exists (\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}_{p++} \times \mathbb{R}_m : \\ \boldsymbol{\lambda} C - \boldsymbol{\mu} A \leqq \boldsymbol{0},\ \boldsymbol{\lambda} C\boldsymbol{x} - \boldsymbol{\mu}\boldsymbol{b} = 0 \end{array} \right\}. \tag{3.9}$$

Furthermore, there is an $M > 0$ such that $\mathbb{R}_{p++}$ above can be replaced by the $(p-1)$-dimensional simplex defined by $\boldsymbol{\lambda}$ of (3.6)

$$\Lambda = \{\, \boldsymbol{\lambda} \in \mathbb{R}_p \mid \boldsymbol{\lambda} \geqq \boldsymbol{e},\ \boldsymbol{\lambda}\boldsymbol{1} = M \,\}. \tag{3.10}$$

Proof: The assertion (3.7) is straightforward by Lemma 3.3. The equivalence among (3.7), (3.8) and (3.9) follows from the duality theorem of linear program. We will prove only that Λ defined by (3.10) can replace $\mathbb{R}_{p++}$ in (3.7), (3.8) and

(3.9). By (3.7), X_E is the union of finitely many faces, say $F_1, \cdots, F_L$, of X such that F_ℓ is the optimal face of $(SC(\boldsymbol{\lambda}^\ell))$ for some $\boldsymbol{\lambda}^\ell \in \mathbb{R}_{p++}$. For $\ell = 1, \ldots, L$, let $\alpha_\ell = 1/\min\{\, \lambda_i^\ell \mid i = 1, \ldots, p \,\}$, and $M = \max\{\, \alpha_\ell(\boldsymbol{\lambda}^\ell \mathbf{1}) \mid \ell = 1, \ldots, L \,\}$. Then $\bar{\boldsymbol{\lambda}}^\ell = (M/\boldsymbol{\lambda}^\ell \mathbf{1})\boldsymbol{\lambda}^\ell \in \Lambda$ and F_ℓ remains the optimal face of $(SC(\bar{\boldsymbol{\lambda}}^\ell))$ for each $\ell = 1, \ldots, L$. □

Theorem 3.5 will provide several equivalent formulations of Problem (P_E). By (3.7) (P_E) is equivalent to

$$\left|\begin{array}{ll} \max\limits_{(\boldsymbol{x},\boldsymbol{\lambda})} & \phi(\boldsymbol{x}) \\ \text{s.t.} & (\boldsymbol{x},\boldsymbol{\lambda}) \in X \times \Lambda \\ & \boldsymbol{\lambda} C \boldsymbol{x} \geqq \boldsymbol{\lambda} C \boldsymbol{x}' \text{ for all } \boldsymbol{x}' \in X. \end{array}\right.$$

Using the gap function with direction $\boldsymbol{\lambda}$ of (3.6), we rewrite the above problem as

$$\left|\begin{array}{ll} \max\limits_{(\boldsymbol{x},\boldsymbol{\lambda})} & \phi(\boldsymbol{x}) \\ \text{s.t.} & (\boldsymbol{x},\boldsymbol{\lambda}) \in X \times \Lambda \\ & g_{\boldsymbol{\lambda}}(\boldsymbol{x}) \leqq 0. \end{array}\right.$$

By (3.8) and (3.9) we have

$$\left|\begin{array}{ll} \max\limits_{(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\nu})} & \phi(\boldsymbol{x}) \\ \text{s.t.} & (\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\nu}) \in X \times \Lambda \times \mathbb{R}_m \times \mathbb{R}_{n+} \\ & \boldsymbol{\lambda} C - \boldsymbol{\mu} A + \boldsymbol{\nu} = \mathbf{0} \\ & \boldsymbol{\nu}\boldsymbol{x} = 0, \end{array}\right.$$

and

$$\left|\begin{array}{ll} \max\limits_{(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})} & \phi(\boldsymbol{x}) \\ \text{s.t.} & (\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu}) \in X \times \Lambda \times \mathbb{R}_m \\ & \boldsymbol{\lambda} C - \boldsymbol{\mu} A \leqq \mathbf{0} \\ & \boldsymbol{\lambda} C \boldsymbol{x} - \boldsymbol{\mu}\mathbf{0} = 0. \end{array}\right.$$

Note that even if ϕ is linear, these problems contain a nonlinear equality constraint. Note also that to obtain a point in $X_E \cap X_V$ we have only to choose $\boldsymbol{\lambda} \in \Lambda$ and solve

$(SC(\boldsymbol{\lambda}))$.

3.2.3 Connectedness of X_E and a local search procedure

The condition (3.8) remains identical as long as the set of binding constraints at $\boldsymbol{x}$ does not change. Therefore, if points $\boldsymbol{x}$ and $\boldsymbol{x}'$ lie in the relative interior of the same face of X, we see that $\boldsymbol{x} \in X_E$ if and only if $\boldsymbol{x}' \in X_E$.

Theorem 3.6 *The set X_E is the connected union of several faces of X. Any two vertices in X_E are connected by a path of efficient edges, where an efficient edge is an edge of X contained in X_E.*

Proof: See Theorem 9.19 and Theorem 9.23 in Steuer [49], Theorem 3.31 in Sawaragi-Nakayama-Tanino [44], and Naccache [38]. □

Let $\boldsymbol{x} = (\boldsymbol{x}^B, \boldsymbol{x}^N) = (B^{-1}\boldsymbol{b}, \boldsymbol{0})$ be a basic feasible solution of X and let $A = \begin{bmatrix} B & N \end{bmatrix}$ and $C = \begin{bmatrix} C_B & C_N \end{bmatrix}$ be the partitions of A and C corresponding to the basic and nonbasic parts of $\boldsymbol{x}$, respectively. Theorem 3.5 implies the following lemma.

Lemma 3.7 *Let $\boldsymbol{x} = (\boldsymbol{x}^B, \boldsymbol{x}^N) = (B^{-1}\boldsymbol{b}, \boldsymbol{0})$ be a basic feasible solution of X. Then we have the followings.*

(i) The point $\boldsymbol{x}$ is in X_E if and only if $\boldsymbol{\lambda}(C_N - C_B B^{-1} N) - \boldsymbol{\nu}^B B^{-1} N \leqq 0$ for some $\boldsymbol{\lambda} \in \Lambda$ and $\boldsymbol{\nu}^B \in \mathbb{R}_{m+}$ such that $\boldsymbol{\nu}^B \boldsymbol{x}^B = 0$.

(ii) If $\boldsymbol{x}$ is a nondegenerate basic solution, the above condition is reduced to $\boldsymbol{\lambda}(C_N - C_B B^{-1} N) \leqq \boldsymbol{0}$ for some $\boldsymbol{\lambda} \in \Lambda$.

(iii) Let $\boldsymbol{c}^j$ and $\boldsymbol{a}^j$ be the columns of C_N and N, respectively, corresponding to a nonbasic variable x_j. If $\boldsymbol{\lambda}(C_N - C_B B^{-1} N) \leqq \boldsymbol{0}$ and $\boldsymbol{\lambda}(\boldsymbol{c}^j - C_B B^{-1} \boldsymbol{a}^j) = 0$ for some $\boldsymbol{\lambda} \in \Lambda$, then the edge obtained by increasing x_j is an efficient edge.

Note that for an efficient basic solution $\boldsymbol{x} = (B^{-1}\boldsymbol{b}, \boldsymbol{0})$ and a nonbasic variable x_j the condition of Lemma 3.7 (iii) holds if and only if the optimal value of

$$(Q^j) \quad \left| \begin{array}{ll} \max\limits_{\boldsymbol{\lambda}} & \boldsymbol{\lambda}(\boldsymbol{c}^j - C_B B^{-1}\boldsymbol{a}^j) \\ \text{s.t.} & \boldsymbol{\lambda}(C_N - C_B B^{-1} N) \leqq \mathbf{0} \\ & \boldsymbol{\lambda} \in \Lambda, \end{array} \right.$$

is equal to zero.

Lemma 3.7 provides the local search procedure described below. For two adjacent vertices $\boldsymbol{v}, \boldsymbol{v}'$ of X let $[\boldsymbol{v}, \boldsymbol{v}']$ denote the edge connecting $\boldsymbol{v}$ and $\boldsymbol{v}'$. For $\boldsymbol{v} \in X_E \cap X_V$ let

$$N_E(\boldsymbol{v}) = \{\, \boldsymbol{v}' \in X_E \cap X_V \mid [\boldsymbol{v}, \boldsymbol{v}'] \subseteqq X_E \,\}, \tag{3.11}$$

i.e., the set of efficient vertices linked to $\boldsymbol{v}$ by an efficient edge. When ϕ is quasi-convex we have the lemma.

Lemma 3.8 *Suppose $\boldsymbol{x} \in X_E \cap X_V$. If $\{\, \boldsymbol{x}' \in N_E(\boldsymbol{x}) \mid \phi(\boldsymbol{x}') > \phi(\boldsymbol{x}) \,\} = \emptyset$, then $\boldsymbol{x}$ is a locally optimal solution of (P_E).*

We describe the procedure starting from $\boldsymbol{v} \in X_E \cap X_V$ as follows.

/** local search procedure for (P_E) starting from $\boldsymbol{v} \in X_E \cap X_V$ **/

$\langle 0 \rangle$ (initialization) Set $\boldsymbol{v}^0 := \boldsymbol{v}$ and $k := 0$.

$\langle k \rangle$ (iteration k) Let B and N be a basic matrix and a nonbasic matrix associated with $\boldsymbol{v}^k = (\boldsymbol{v}^B, \boldsymbol{v}^N) = (B^{-1}\boldsymbol{b}, \mathbf{0})$. Let R be the index set of nonbasic variables.

- $\langle k1 \rangle$ (termination) If $R = \emptyset$ then stop ($\boldsymbol{v}^k$ is a locally optimal solution). Otherwise, choose $j \in R$ and set $R := R \setminus \{j\}$.
- $\langle k2 \rangle$ (obtaining $\boldsymbol{v}'$) If the optimal value of (Q^j) is equal to zero then obtain $\boldsymbol{v}' \in N_E(\boldsymbol{v}^k)$ by pivotting. Otherwise go to $\langle k1 \rangle$.
- $\langle k3 \rangle$ (update) If $\phi(\boldsymbol{v}') > \phi(\boldsymbol{v}^k)$ then set $\boldsymbol{v}^{k+1} := \boldsymbol{v}'$, $k := k + 1$, and go to $\langle k \rangle$. Otherwise go to $\langle k1 \rangle$.

3.2.4 Efficient outcome

The set

$$Y = CX = \{\, \boldsymbol{y} \in \mathbb{R}^p \mid \boldsymbol{y} = C\boldsymbol{x} \text{ for some } \boldsymbol{x} \in X \,\}$$

is called the *outcome set*. The sets

$$Y^{\leqq} = Y + \mathbb{R}^p_- = \{\, \boldsymbol{y} \in \mathbb{R}^p \mid \boldsymbol{y} \leqq C\boldsymbol{x} \text{ for some } \boldsymbol{x} \in X \,\} \text{ and}$$
$$Y^{<} = Y + \mathbb{R}^p_{--} = \{\, \boldsymbol{y} \in \mathbb{R}^p \mid \boldsymbol{y} < C\boldsymbol{x} \text{ for some } \boldsymbol{x} \in X \,\}$$

are called the *lower outcome set* and *strictly lower outcome set*, respectively.

A point $\boldsymbol{y} \in Y$ is said to be an *efficient outcome* if there is no point $\boldsymbol{y}' \in Y$ such that $\boldsymbol{y}' \geqq \boldsymbol{y}$ and $\boldsymbol{y}' \neq \boldsymbol{y}$, in other words, $Y \cap (\boldsymbol{y} + \mathbb{R}^p_+) = \{\boldsymbol{y}\}$. We denote the set of efficient outcomes by Y_E, i.e.,

$$Y_E = \{\, \boldsymbol{y} \in Y \mid \nexists \boldsymbol{y}' \in Y : \boldsymbol{y}' \geqq \boldsymbol{y},\ \boldsymbol{y}' \neq \boldsymbol{y} \,\}.$$

A point $\boldsymbol{y} \in Y$ is said to be a *weakly efficient outcome* if there is no point $\boldsymbol{y}' \in Y$ such that $\boldsymbol{y}' > \boldsymbol{y}$, in other words, $Y \cap (\boldsymbol{y} + \mathbb{R}^p_{++}) = \emptyset$. We denote the set of weakly efficient outcomes by Y_W, i.e.,

$$Y_W = \{\, \boldsymbol{y} \in Y \mid \nexists \boldsymbol{y}' \in Y : \boldsymbol{y}' > \boldsymbol{y} \,\}.$$

The following lemma is a restatement of these definitions.

Lemma 3.9
(i) $X_E = \{\, \boldsymbol{x} \in X \mid C\boldsymbol{x} \in Y_E \,\}$.
(ii) $X_W = \{\, \boldsymbol{x} \in X \mid C\boldsymbol{x} \in Y_W \,\}$.

3.3 Adjacent Vertex Search Algorithm

In this section we explain an adjacent vertex search algorithm. The algorithms proposed in Philip [40], Ecker-Song [19] and Fülöp [22] for a linear function ϕ, and in Bolintineanu [13] for a quasi-convex function ϕ are mainly based on the two techniques: applying the local search procedure, which is stated in the previous section, and cutting off the portion of X where ϕ takes a smaller value than objective function value of an incumbent. We assume for the time being the quasi-convex function ϕ and will follow the line of Bolintineanu [13].

3.3.1 Algorithm

The algorithm generates a sequence of efficient vertices $\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots$ and polytopes $X^0, X^1, \cdots$ such that $\phi(\boldsymbol{x}^0) < \phi(\boldsymbol{x}^1) < \cdots$ and $X = X^0 \supseteqq X^1 \supseteqq \cdots$. Here determined by a hyperplane $H = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}\boldsymbol{x} = \alpha \,\}$, we denote the two halfspaces by $H_+ = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}\boldsymbol{x} \geqq \alpha \,\}$ and $H_- = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}\boldsymbol{x} \leqq \alpha \,\}$, and their interiors by H_{++} and H_{--}, respectively. The algorithm is described as follows.

/** adjacent vertex search algorithm for (P_E) **/

$\langle 0 \rangle$ (initialization) Find an initial feasible solution $\boldsymbol{x}^0 \in X_E \cap X_V$. If $N_E(\boldsymbol{x}^0) = \emptyset$ then stop ($\boldsymbol{x}^0$ is the optimal solution of (P_E)). Set $p := 0$, $k := 0$ and $X^0 := X$.

$\langle\langle p \rangle\rangle$ (local search) Apply the local search procedure for (P_E) starting from $\boldsymbol{x}^p$ to obtain a locally optimal vertex $\boldsymbol{x}^{p+1}$ and the lower bound $\phi(\boldsymbol{x}^{p+1})$. Set $p := p + 1$.

$\langle k \rangle$ (global technique) Let $L^p := \{\, \boldsymbol{x} \mid \phi(\boldsymbol{x}) \leqq \phi(\boldsymbol{x}^p) \,\}$.

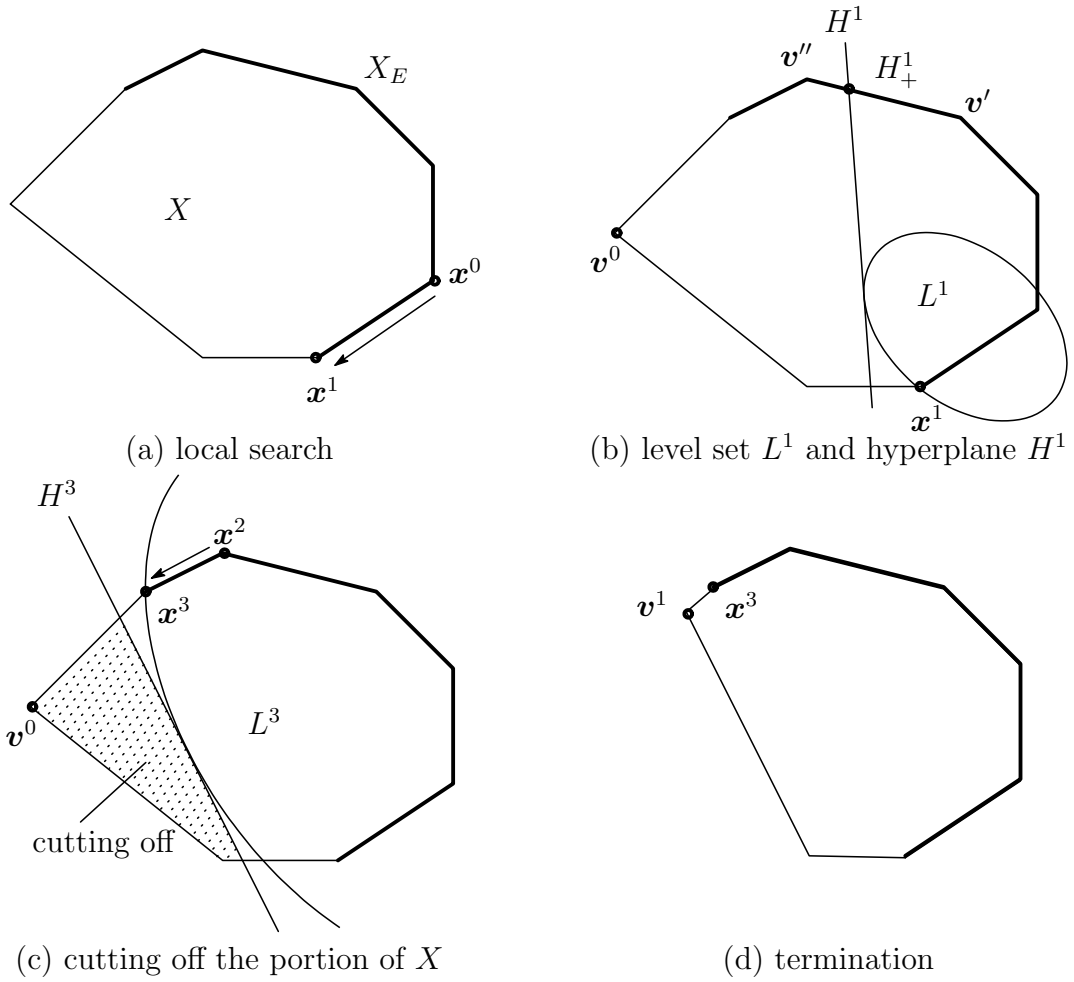
$\langle k1 \rangle$ (computing an upper bound) Solve $\max\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X^k \,\}$ to obtain a solution $\boldsymbol{v}^k$ and the upper bound $\phi(\boldsymbol{v}^k)$.

$\langle k2 \rangle$ (termination) If $\phi(\boldsymbol{x}^p) \geqq \phi(\boldsymbol{v}^k) - \varepsilon$ for some tolerance $\varepsilon > 0$, then stop ($\boldsymbol{x}^p$ is an ε-approximate optimal solution). Otherwise, find a supporting hyperplane H^k of L^p such that $L^p \subseteqq H^k_+$ and $\boldsymbol{v}^k \in H^k_{--}$.

$\langle k3 \rangle$ (update) If there is an efficient edge $[\boldsymbol{v}', \boldsymbol{v}''] \subseteqq X_E$ such that $[\boldsymbol{v}', \boldsymbol{v}''] \cap H^k \neq \emptyset$ and $\max\{\phi(\boldsymbol{v}'), \phi(\boldsymbol{v}'')\} > \phi(\boldsymbol{x}^p)$, then set $\boldsymbol{x}^{p+1}$ be one of $\boldsymbol{v}'$ and $\boldsymbol{v}''$ with a larger objective function value, $p := p + 1$ and go to $\langle\langle p \rangle\rangle$.

$\langle k4 \rangle$ (cutting the current polytope) Otherwise, set $X^{k+1} := X^k \cap H^k_+$, $k := k + 1$, and go to $\langle k \rangle$.

We illustrate the adjacent vertex search algorithm in Figure 3.5. By local search procedure we obtain $\boldsymbol{x}^1$ (See (a)). We construct the level set L^1 and the hyperplane H^1 (See (b)). Since the hyperplane H^1 intersects the efficient edge $[\boldsymbol{v}', \boldsymbol{v}'']$, we apply the local search procedure starting from $\boldsymbol{v}''$ to obtain $\boldsymbol{x}^3$. We construct the level set



(a) local search

(b) level set L^1 and hyperplane H^1

(c) cutting off the portion of X

(d) termination

Figure 3.5: Adjacent vertex search algorithm

L^3 and the hyperplane H^3, and then cut the portion of X off (See (c)). In the last place we terminate with an ε-approximate optimal solution $\boldsymbol{x}^3$ (See (d)).

The most costly and crucial step would be Step $\langle k3\rangle$ as well as Step $\langle k1\rangle$, in which a quasi-convex maximization problem $\max\{\,\phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X^k\,\}$ is to be solved. We will not go into detail of how to solve the quasi-convex maximization problem. See e.g., Horst-Tuy [27].

Step $\langle k3\rangle$ is based on the following observation.

Lemma 3.10 *Let $F^k = X^k \cap H^k$ and F^k_E be the set of efficient points for*

$$\left|\begin{array}{ll} \textit{Vector Max} & C\boldsymbol{x} \\ \textit{s.t.} & \boldsymbol{x} \in F^k. \end{array}\right.$$

Then $X_E \cap F^k \subseteqq F_E^k$.

Proof: If $\boldsymbol{x} \in X_E \cap F^k$, there is no point $\boldsymbol{x}' \in X$ such that $C\boldsymbol{x}' \geqq C\boldsymbol{x}$ and $C\boldsymbol{x}' \neq C\boldsymbol{x}$. Then clearly no points in F^k meet this condition, which means $\boldsymbol{x} \in F_E^k$. □

This lemma shows that if we enumerate all the efficient vertices of F_E^k, we can see if there is the edge desired in Step $\langle k3 \rangle$. Namely, Step $\langle k3 \rangle$ is carried out by generating the efficient vertices of F^k by a standard algorithm for linear multicriteria optimization such as ADBASE by Steuer [50] till one of them turns out to be in X_E, and then for such a point, checking if it lies on an efficient edge of X^k with endpoints $\boldsymbol{v}'$ and $\boldsymbol{v}''$ such that $\max\{\phi(\boldsymbol{v}'), \phi(\boldsymbol{v}'')\} > \phi(\boldsymbol{x}^p)$.

3.3.2 Convergence of the algorithm

Lemma 3.11 *Let* $\boldsymbol{u}^k$ *denote the point at which* H^k *supports* L^p. *If the angle between* $\boldsymbol{v}^k - \boldsymbol{u}^k$ *and the normal vector of* H^k *pointing toward* $\boldsymbol{v}^k$ *is less than some constant* δ, *then Step* $\langle k \rangle$ *terminates after a finite number of iterations for each* p.

Proof: The condition implies $\lim_{k\to\infty} \phi(\boldsymbol{v}^k) = \phi(\boldsymbol{x}^p)$, hence the stopping criterion $\phi(\boldsymbol{x}^p) \geqq \phi(\boldsymbol{v}^k) - \varepsilon$ will be satisfied within a finite number of iterations. □

Lemma 3.12 $X_E \subseteqq X^k$ *for* $k = 0, 1, \ldots$.

Proof: Since $X_E \subseteqq X^0 = X$, suppose $X_E \subseteqq X^k$ as the inductive hypothesis. If $X_E \not\subseteqq X^{k+1}$, there is $\boldsymbol{x}' \in X_E \cap X_V$ such that $\boldsymbol{x}' \notin H_+^k$. By the construction of H^k we see $\phi(\boldsymbol{x}') > \phi(\boldsymbol{x}^p)$. Then by Theorem 3.6 there is an efficient edge $[\boldsymbol{v}', \boldsymbol{v}'']$ with $[\boldsymbol{v}', \boldsymbol{v}''] \cap H^k \neq \emptyset$ and $\max\{\phi(\boldsymbol{v}'), \phi(\boldsymbol{v}'')\} > \phi(\boldsymbol{x}^p)$. This is contrary to the fact that X^{k+1} was generated. □

Lemma 3.13 *When the algorithm terminates with* $\boldsymbol{x}^p$ *and* $\boldsymbol{v}^k$ *satisfying* $\phi(\boldsymbol{x}^p) \geqq \phi(\boldsymbol{v}^k) - \varepsilon$, $\boldsymbol{x}^p$ *is an* ε*-approximate optimal solution of* (P_E).

Proof: By Lemma 3.12 we obtain $\phi(P_E) = \max\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X_E \,\} \leqq \max\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X^k \,\} = \phi(\boldsymbol{v}^k) \leqq \phi(\boldsymbol{x}^p) + \varepsilon \leqq \phi(P_E) + \varepsilon$. □

Theorem 3.14 *The algorithm provides an ε-approximate optimal solution of Problem (P_E) after a finite number of iterations.*

Proof: Step $\langle k \rangle$ terminates within finitely many iterations for each p as shown in Lemma 3.11, and points $\boldsymbol{x}^p$'s are efficient vertices of X satisfying $\phi(\boldsymbol{x}^0) < \phi(\boldsymbol{x}^1) < \cdots$, and hence distinct. Therefore the finiteness of $X_E \cap X_V$ and Lemma 3.13 imply the theorem. □

A preliminary computational experiment for small problems up to $n = 7, m = 7, p = 4$ with a convex quadratic or linear objective function is reported in Bolintineanu [13], where it is observed that the vertices, including those on the cutting planes, generated by the algorithm are fewer than the efficient vertices of X.

3.3.3 In the case where ϕ is linear

When $\phi(\boldsymbol{x}) = \boldsymbol{d}\boldsymbol{x}$ for $\boldsymbol{d} \in \mathbb{R}_n$, the algorithm is substantially simplified. Suppose we have obtained a locally optimal solution $\boldsymbol{x}^p \in X_E \cap X_V$. Then the lower level set is the half space $L^p = \{\, \boldsymbol{x} \mid \boldsymbol{d}\boldsymbol{x} \leqq \boldsymbol{d}\boldsymbol{x}^p \,\}$ and the supporting hyperplane of this set is uniquely determined by $H^p = \{\, \boldsymbol{x} \mid \boldsymbol{d}\boldsymbol{x} = \boldsymbol{d}\boldsymbol{x}^p \,\}$. Then the efficient vertices of $F^k = X \cap H^k$ are enumerated to check if H^k intersects an efficient edge $[\boldsymbol{v}', \boldsymbol{v}'']$ of X such that $\max\{\boldsymbol{d}\boldsymbol{v}', \boldsymbol{d}\boldsymbol{v}''\} > \boldsymbol{d}\boldsymbol{x}^p$. When no such edge exists, we conclude from the connectedness of X_E that

$$X_E \subseteqq \{\, \boldsymbol{x} \mid \boldsymbol{d}\boldsymbol{x} \leqq \boldsymbol{d}\boldsymbol{x}^p \,\}$$

and hence $\boldsymbol{x}^p$ is an optimal solution of (P_E). Thus, k is never incremented through the algorithm.

In the enumeration of efficient vertices of $F^k = X \cap H^k$ Fülöp [22] proposed a cutting plane algorithm based on convexity and disjunctive cuts. Assume we have a vertex $\bar{\boldsymbol{x}} \in F^k$ which is not efficient, i.e., $g_{\boldsymbol{\lambda}}(\bar{\boldsymbol{x}}) > 0$, where $g_{\boldsymbol{\lambda}}$ is the gap function of (3.6). The portion of F^k with $g_{\boldsymbol{\lambda}}(\boldsymbol{x}) > 0$, which is a convex set, should be cut off and eliminated for further enumeration. Fülöp proposed to introduce a convexity cut $\boldsymbol{t}\boldsymbol{x} \geqq 1$, where $\boldsymbol{t} \in \mathbb{R}_n$, and reduce the set F^k to $F^k \cap \{\, \boldsymbol{x} \mid \boldsymbol{t}\boldsymbol{x} \geqq 1 \,\}$. Suppose the nondegeneracy at $\bar{\boldsymbol{x}}$, and for each non-basic variable x_j let $\boldsymbol{z}^j$ be the direction of

edge of F^k adjacent to $\bar{\boldsymbol{x}}$ obtained by increasing x_j. Note that $\boldsymbol{z}^j$ is easily obtained from the dictionary corresponding to $\bar{\boldsymbol{x}}$. Let

$$s_j = \sup\{\, s \in \mathbb{R} \mid C(\bar{\boldsymbol{x}} + s\boldsymbol{z}^j) \leqq C\boldsymbol{x},\ \boldsymbol{x} \in F^k \,\}, \tag{3.12}$$

then we have the convexity cut as follows. Note that the constraint $C(\bar{\boldsymbol{x}}+s\boldsymbol{z}^j) \leqq C\boldsymbol{x}$ together with $\boldsymbol{x} \in F^k$ means that $C(\bar{\boldsymbol{x}} + s\boldsymbol{z}^j)$ be in the lower outcome set of CF^k.

Lemma 3.15 *Suppose $s_j > 0$ for every non-basic variable x_j of $\bar{\boldsymbol{x}}$. Let $\boldsymbol{t} \in \mathbb{R}_n$ be defined by*

$$t_j = \begin{cases} 1/s_j & \text{if } x_j \text{ is a non-basic variable and } s_j < \infty, \\ 0 & \text{otherwise.} \end{cases}$$

Then $\boldsymbol{t}\bar{\boldsymbol{x}} < 1$, and $\boldsymbol{t}\boldsymbol{x} \geqq 1$ for all efficient points $\boldsymbol{x}$ in F^k.

See Horst-Tuy [27] for further detail of convexity cut. Every time a nonefficient vertex is found, F^k is reduced by the convexity cut, which might lighten the computational burden. No computational experiment is reported in Fülöp [22]. Ecker-Song [19] proposed to solve $\max\{\, \boldsymbol{c}^i\boldsymbol{x} \mid \boldsymbol{x} \in X \cap H_+^k \,\}$ for $i = 1, \ldots, p$ to find the next iterate $\boldsymbol{x}^{p+1}$ before resorting to the vertex enumeration of F^k.

3.4 Non-adjacent Vertex Search Algorithm

The algorithms which trace the adjacent vertices need a step of enumerating all efficient vertices of a polyhedral set with a lower dimension. This section explains a nonadjacent vertex search algorithm proposed by Benson [8, 9], which dispenses with the vertex enumeration.

We assume the linear objective function $\phi(\boldsymbol{x}) = \boldsymbol{d}\boldsymbol{x}$. Suppose we have $k+1$ efficient points $\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^k \in X_E$ and let $\alpha_k = \max\{\, \boldsymbol{d}\boldsymbol{x}^j \mid j = 0, 1, \ldots, k \,\}$ and (P^k) be the problem, which plays a central role in the algorithm, of finding a point $(\boldsymbol{x}, \boldsymbol{\lambda}) \in \mathbb{R}^n \times \mathbb{R}_{p++}$ satisfying

$$(P^k) \qquad \left| \begin{array}{l} \boldsymbol{\lambda}C\boldsymbol{x} \geqq \boldsymbol{\lambda}C\boldsymbol{x}^j \text{ for } j = 0, 1, \ldots, k \\ (\boldsymbol{x}, \boldsymbol{\lambda}) \in X \times \Lambda \\ \boldsymbol{d}\boldsymbol{x} > \alpha_k. \end{array} \right.$$

Remark 3.16 *If* $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\lambda}}) \in X \times \Lambda$ *satisfies the constraints*

$$\boldsymbol{\lambda} C\boldsymbol{x} \geqq \boldsymbol{\lambda} C\boldsymbol{x}^j \text{ for } j = 0, 1, \ldots, k$$

of (P^k)*, we see that* $\bar{\boldsymbol{x}}$ *is an efficient point of the convex hull of* $\boldsymbol{x}^0, \cdots, \boldsymbol{x}^k$ *and* $\bar{\boldsymbol{x}}$ *itself. In this sense Problem* (P^k) *is an inner approximation of Problem* (P_E)*.*

We start with the case where Problem (P^k) has no solution.

Lemma 3.17 *Suppose* $\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^k \in X_E$ *and Problem* (P^k) *has no solution. Then* $\boldsymbol{x}^* \in \mathrm{argmax}\{\, \boldsymbol{d}\boldsymbol{x}^j \mid j = 0, 1, \ldots, k \,\}$ *is an optimal solution of* (P_E)*.*

Proof: Since (P^k) has no solution, if $\boldsymbol{x} \in X$ together with some $\boldsymbol{\lambda} \in \Lambda$ satisfies

$$\boldsymbol{\lambda} C\boldsymbol{x} \geqq \boldsymbol{\lambda} C\boldsymbol{x}' \text{ for all } \boldsymbol{x}' \in X \tag{3.13}$$

then $\boldsymbol{d}\boldsymbol{x} \leqq \alpha_k$, i.e., $\boldsymbol{x} \in X_E$ implies $\boldsymbol{d}\boldsymbol{x} \leqq \alpha_k$. This and $\boldsymbol{x}^* \in X_E$ yield the lemma. □

3.4.1 Algorithm and its convergence

Leaving the method of solving (P^k) till later on, we give the algorithm first.

/** nonadjacent vertex search algorithm for (P_E) **/

$\langle 0 \rangle$ (initialization) Find an efficient vertex $\boldsymbol{x}^0$, set $k := 0$ and go to $\langle k \rangle$.

$\langle k \rangle$ (Iteration k)

$\langle k1 \rangle$ Find a solution $(\boldsymbol{x}, \boldsymbol{\lambda}) \in \mathbb{R}^n \times \mathbb{R}_{p++}$ of (P^k). If no solution exists, $\boldsymbol{x}^* \in \mathrm{argmax}\{\, \boldsymbol{d}\boldsymbol{x}^j \mid j = 0, \ldots, k \,\}$ is an optimal solution of (P_E). Otherwise, set $(\bar{\boldsymbol{x}}^{k+1}, \bar{\boldsymbol{\lambda}}^{k+1})$ be the solution found.

$\langle k2 \rangle$ Solve the linear program

$$(Test^k) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \boldsymbol{e}C\boldsymbol{x} \\ \text{s.t.} & C\boldsymbol{x} \geqq C\bar{\boldsymbol{x}}^{k+1} \\ & \boldsymbol{x} \in X \end{array} \right.$$

for a solution $\hat{\boldsymbol{x}}$. If $\boldsymbol{e}C\hat{\boldsymbol{x}} = \boldsymbol{e}C\bar{\boldsymbol{x}}^{k+1}$, go to $\langle k3 \rangle$. Otherwise, go to $\langle k5 \rangle$.

$\langle k3 \rangle$ If $\bar{\boldsymbol{x}}^{k+1}$ is a vertex of X, then set $\boldsymbol{x}^{k+1} := \bar{\boldsymbol{x}}^{k+1}$, $k := k+1$ and go to $\langle k \rangle$. Otherwise, go to $\langle k4 \rangle$.

$\langle k4 \rangle$ Let F be a face of X whose relative interior contains $\bar{\boldsymbol{x}}^{k+1}$, and solve the linear program

$$(Face^k) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \boldsymbol{d}\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in F \end{array} \right.$$

for an extreme point $\boldsymbol{x}^{k+1}$. Set $k := k+1$ and go to $\langle k \rangle$.

$\langle k5 \rangle$ Solve $(SC(\bar{\boldsymbol{\lambda}}^{k+1}))$ for a solution $\boldsymbol{x}^{k+1}$, set $k := k+1$ and go to $\langle k \rangle$.

Note that whether $\bar{\boldsymbol{x}}^{k+1}$ is a vertex of X can be seen by checking the linear independence of columns of A corresponding to positive components of $\bar{\boldsymbol{x}}^{k+1}$.

There may be various ways of determining the face F of Step $\langle k4 \rangle$. One possible way is

$$F = \{\, \boldsymbol{x} \in X \mid x_j = 0 \text{ for } j \text{ with } \bar{x}_j^{k+1} = 0 \,\}. \tag{3.14}$$

Benson [8, 9] proposes to define it by

$$F = \{\, \boldsymbol{x} \in X \mid (\boldsymbol{e} + \boldsymbol{u})C\boldsymbol{x} = v \,\},$$

where $\boldsymbol{u}$ is an optimal dual variable vector corresponding to the constraint $C\boldsymbol{x} \geqq C\bar{\boldsymbol{x}}^{k+1}$ of $(Test^k)$ and $v = \max\{\, (\boldsymbol{e} + \boldsymbol{u})C\boldsymbol{x} \mid \boldsymbol{x} \in X \,\}$.

The following lemma shows that $\boldsymbol{x}^j$'s are efficient vertices of X.

Lemma 3.18 *$\boldsymbol{x}^j \in X_E \cap X_V$ for $j = 0, 1, \ldots$.*

Proof: Since it is clear that $\boldsymbol{x}^j \in X_V$, we only show that $\boldsymbol{x}^j \in X_E$. When $\boldsymbol{x}^{k+1}$ is computed in either Step $\langle k3 \rangle$ or Step $\langle k5 \rangle$, it is an optimal solution of either $(Test^k)$ or $(SC(\bar{\boldsymbol{\lambda}}^{k+1}))$. Then clearly $\boldsymbol{x}^{k+1} \in X_E$. When $\boldsymbol{x}^{k+1}$ is generated in Step $\langle k4 \rangle$, it lies in the face whose relative interior contains the efficient point $\bar{\boldsymbol{x}}^{k+1}$. Then by Theorem 3.5 we see $\boldsymbol{x}^{k+1} \in X_E$. □

Now we show that the algorithm always generates a sequence of distinct vertices of X_E.

Lemma 3.19 *$\boldsymbol{x}^{k+1} \notin \{\, \boldsymbol{x}^j \mid j = 0, 1, \ldots, k \,\}$.*

Proof: Three cases should be considered. In Step $\langle k3\rangle$ $\boldsymbol{x}^{k+1}$ is given by $\boldsymbol{x}^{k+1} = \bar{\boldsymbol{x}}^{k+1}$, which satisfies $\boldsymbol{d}\bar{\boldsymbol{x}}^{k+1} > \max\{\, \boldsymbol{d}\boldsymbol{x}^j \mid j = 0, \ldots, k \,\}$, and hence $\boldsymbol{x}^{k+1}$ differs from any point of $\boldsymbol{x}^0, \cdots, \boldsymbol{x}^k$. By construction $\boldsymbol{d}\boldsymbol{x}^{k+1} \geqq \boldsymbol{d}\bar{\boldsymbol{x}}^{k+1}$ in Step $\langle k4\rangle$ and the same argument applies. Now suppose $\boldsymbol{x}^{k+1}$ is generated in Step $\langle k5\rangle$. Then $\bar{\boldsymbol{x}}^{k+1} \notin X_E$, i.e., there is a point, say $\tilde{\boldsymbol{x}} \in X$ with $C\tilde{\boldsymbol{x}} \geqq C\bar{\boldsymbol{x}}^{k+1}$ and $C\tilde{\boldsymbol{x}} \neq C\bar{\boldsymbol{x}}^{k+1}$. Since $\bar{\boldsymbol{\lambda}}^{k+1} > 0$ we see $\bar{\boldsymbol{\lambda}}^{k+1}C\tilde{\boldsymbol{x}} > \bar{\boldsymbol{\lambda}}^{k+1}C\bar{\boldsymbol{x}}^{k+1}$. Since $\boldsymbol{x}^{k+1}$ solves $(SC(\bar{\boldsymbol{\lambda}}^{k+1}))$, we also see $\bar{\boldsymbol{\lambda}}^{k+1}C\boldsymbol{x}^{k+1} \geqq \bar{\boldsymbol{\lambda}}^{k+1}C\tilde{\boldsymbol{x}}$. Then for $j = 0, \ldots, k$

$$\bar{\boldsymbol{\lambda}}^{k+1}C\boldsymbol{x}^{k+1} \geqq \bar{\boldsymbol{\lambda}}^{k+1}C\tilde{\boldsymbol{x}} > \bar{\boldsymbol{\lambda}}^{k+1}C\bar{\boldsymbol{x}}^{k+1} \geqq \bar{\boldsymbol{\lambda}}^{k+1}C\boldsymbol{x}^j$$

holds.

This means that $\boldsymbol{x}^{k+1} \notin \{\, \boldsymbol{x}^j \mid j = 0, \ldots, k \,\}$. □

Note that in either case of Step $\langle k3\rangle$ and Step $\langle k4\rangle$ $\boldsymbol{d}\boldsymbol{x}^{k+1} > \max\{\, \boldsymbol{d}\boldsymbol{x}^j \mid j = 0, \ldots, k \,\}$, i.e., monotone increasing of the objective function value, but in case Step $\langle k5\rangle$ it may decrease. Combining the above lemmas we have the following theorem.

Theorem 3.20 *Suppose Problem (P^k) is solved within a finite number of iterations. Then the algorithm provides an optimal solution $\boldsymbol{x}^*$ of Problem (P_E) after a finite number of iterations.*

3.4.2 Solving Problem (P^k)

Now we go back to Problem (P^k) and explain the algorithm proposed by Benson [8]. For a solution of (P^k) it suffices to solve

$$(\overline{P^k}) \qquad \left| \begin{array}{ll} \max\limits_{(\boldsymbol{x},\boldsymbol{\lambda})} & \boldsymbol{d}\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{\lambda}C\boldsymbol{x} \geqq \boldsymbol{\lambda}C\boldsymbol{x}^j \text{ for } j = 0, 1, \ldots, k \\ & (\boldsymbol{x}, \boldsymbol{\lambda}) \in X \times \Lambda. \end{array} \right.$$

Let

$$\overline{Y} = \left\{ \boldsymbol{y} \;\middle|\; \min\{-\boldsymbol{c}^i\boldsymbol{x} \mid \boldsymbol{x} \in X\} \leqq y_i \leqq \max\{-\boldsymbol{c}^i\boldsymbol{x} \mid \boldsymbol{x} \in X\} \text{ for } i = 1, \ldots, p \right\},$$

and $\overline{\Lambda}$ be a p-dimensional hypercube containing Λ, e.g.,

$$\overline{\Lambda} = \{\, \boldsymbol{\lambda} \in \mathbb{R}_p \mid \boldsymbol{e} \leqq \boldsymbol{\lambda} \leqq (M + p - 1)\boldsymbol{e} \,\}.$$

Then $(\overline{P^k})$ is equivalent to

$$\left|\begin{array}{ll} \max\limits_{(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{y})} & \boldsymbol{dx} \\ \text{s.t.} & \boldsymbol{\lambda y} + \boldsymbol{\lambda} C\boldsymbol{x}^j \leqq 0 \text{ for } j = 0, 1, \ldots, k \\ & \boldsymbol{y} + C\boldsymbol{x} = 0 \\ & (\boldsymbol{x}, \boldsymbol{\lambda}) \in X \times \overline{\Lambda} \\ & \boldsymbol{y} \in \overline{Y}. \end{array}\right.$$

The constraint $\boldsymbol{\lambda 1} = M$ could be added, but is not necessary. The bilinear term $\boldsymbol{\lambda y}$ makes the problem difficult to solve and hence should be relaxed. The algorithm in Benson [8] is based on the successive partition of the hypercube $\overline{Y} \times \overline{\Lambda}$ into smaller hypercubes and the relaxation of the problem restricted to the smaller hypercubes to a linear program. Let

$$\overline{Y}' \times \overline{\Lambda}' = \prod_{i=1}^{p} [\underline{\alpha}_i, \overline{\alpha}_i] \times \prod_{i=1}^{p} [\underline{\beta}_i, \overline{\beta}_i]$$

be a smaller hypercube contained in $\overline{Y} \times \overline{\Lambda}$. Note that $\overline{Y}' \times \overline{\Lambda}' = \prod_{i=1}^{p} ([\underline{\alpha}_i, \overline{\alpha}_i] \times [\underline{\beta}_i, \overline{\beta}_i])$ by rearranging the coordinates and $\boldsymbol{\lambda y}$ is the sum of bilinear terms $\lambda_i y_i$ defined on $[\underline{\alpha}_i, \overline{\alpha}_i] \times [\underline{\beta}_i, \overline{\beta}_i]$. Al-Khayyal-Falk [1] show that the *convex envelope* of $\lambda_i y_i$ on the two-dimensional cube $[\underline{\alpha}_i, \overline{\alpha}_i] \times [\underline{\beta}_i, \overline{\beta}_i]$, the point-wise supremum of all convex functions underestimating $\lambda_i y_i$ on $[\underline{\alpha}_i, \overline{\alpha}_i] \times [\underline{\beta}_i, \overline{\beta}_i]$, is given by the piece-wise linear convex function $\max\{\, \underline{\beta}_i y_i + \underline{\alpha}_i \lambda_i - \underline{\beta}_i \underline{\alpha}_i, \overline{\beta}_i y_i + \overline{\alpha}_i \lambda_i - \overline{\beta}_i \overline{\alpha}_i \,\}$. Then the convex envelope of $\boldsymbol{\lambda y}$ is given by $\sum_{i=1}^{p} \max\{\, \underline{\beta}_i y_i + \underline{\alpha}_i \lambda_i - \underline{\beta}_i \underline{\alpha}_i, \overline{\beta}_i y_i + \overline{\alpha}_i \lambda_i - \overline{\beta}_i \overline{\alpha}_i \,\}$ and the constraint $\boldsymbol{\lambda y} + \boldsymbol{\lambda} C\boldsymbol{x}^j \leqq 0$ is relaxed to

$$\sum_{i=1}^{p} \max\{\, \underline{\beta}_i y_i + \underline{\alpha}_i \lambda_i - \underline{\beta}_i \underline{\alpha}_i, \overline{\beta}_i y_i + \overline{\alpha}_i \lambda_i - \overline{\beta}_i \overline{\alpha}_i \,\} + \boldsymbol{\lambda} C\boldsymbol{x}^j \leqq 0. \tag{3.15}$$

This constraint is, by introducing variables w_i's, rewritten as

$$\begin{aligned} \underline{\beta}_i y_i + \underline{\alpha}_i \lambda_i - \underline{\beta}_i \underline{\alpha}_i &\leqq w_i \text{ for } i = 1, \ldots, p \\ \overline{\beta}_i y_i + \overline{\alpha}_i \lambda_i - \overline{\beta}_i \overline{\alpha}_i &\leqq w_i \text{ for } i = 1, \ldots, p \\ \sum_{i=1}^{p} w_i + \boldsymbol{\lambda} C\boldsymbol{x}^j &\leqq 0. \end{aligned}$$

Thus we yield a linear programming relaxation of $(\overline{P^k})$ restricted to a smaller hypercube $\overline{Y}' \times \overline{\Lambda}'$ contained in $\overline{Y} \times \overline{\Lambda}$. In Benson [8] (3.15) is further relaxed to a single linear inequality.

It would be a routine to construct a branch-and-bound algorithm based on this relaxation. If we employ the bisection procedure to divide a hypercube, i.e., to divide it into two hypercubes with equal volumes such that the midpoint of one of the longest edges is a vertex of both new hypercubes, we will see the following theorem.

Theorem 3.21 *If the branch-and-bound procedure does not terminate after a finite number of iterations, any accumulation point of the sequence* $(\boldsymbol{x}^\nu, \boldsymbol{y}^\nu, \boldsymbol{\lambda}^\nu, \boldsymbol{w}^\nu)$ *generated by the procedure is an optimal solution of* $(\overline{P^k})$.

Proof: See e.g., Section 4 of Chapter VII in Horst-Tuy [27] for the convergence proof. □

3.5 Branch-and-Bound Based Algorithm

This section is devoted to introducing the branch-and-bound algorithm for Problem (P_W) with a concave function ϕ proposed by Horst-Thoai [26] and Thoai [59].

3.5.1 Master problem (MP) and subproblem $(MP(K))$

First we observe the following characterization of the weakly efficient outcome set Y_W.

Lemma 3.22 *Let* $\partial Y^{\leqq}$ *denote the boundary of* $Y^{\leqq}$. *Then* $Y_W = Y \cap \partial Y^{\leqq}$.

Proof: This lemma follows the equivalence $Y \cap \operatorname{int} Y^{\leqq} = Y \setminus Y_W$. If $\boldsymbol{y} \in Y \cap \operatorname{int} Y^{\leqq}$, $\boldsymbol{y} < \boldsymbol{y}'$ for some $\boldsymbol{y}' \in Y^{\leqq}$, for which there is $\boldsymbol{y}'' \in Y$ such that $\boldsymbol{y}' \leqq \boldsymbol{y}''$. Therefore $\boldsymbol{y} \notin Y_W$. If $\boldsymbol{y} \in Y \setminus Y_W$, there is $\boldsymbol{y}' \in Y$ with $\boldsymbol{y} < \boldsymbol{y}'$, and hence its neighbor $\{\, \boldsymbol{z} \in \mathbb{R}^p \mid \boldsymbol{y} - (\boldsymbol{y}' - \boldsymbol{y}) \leqq \boldsymbol{z} \leqq \boldsymbol{y}' \,\}$ is contained in $Y^{\leqq}$. This implies $\boldsymbol{y} \in \operatorname{int} Y^{\leqq}$. □

Then Problem (P_W) is rewritten as

$$\max\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X,\ C\boldsymbol{x} \in \partial Y^{\leqq} \,\}.$$

Introducing additional variables $\boldsymbol{y} \in \mathbb{R}^p$ and $t \in \mathbb{R}$, it is cast into the following problem called *Master Problem*

$$(MP) \quad \left| \begin{array}{ll} \max\limits_{(\boldsymbol{x},\boldsymbol{y},t)} & t \\ \text{s.t.} & t \leqq \phi(\boldsymbol{x}) \\ & \boldsymbol{x} \in X \\ & \boldsymbol{y} = C\boldsymbol{x} \\ & \boldsymbol{y} \in \partial Y^{\leqq}, \end{array} \right.$$

for which the following theorem holds.

Theorem 3.23 *If $\boldsymbol{x}^*$ is an optimal solution of (P_W), then $(\boldsymbol{x}^*, \boldsymbol{y}^*, t^*)$ with $\boldsymbol{y}^* = C\boldsymbol{x}^*$, $t^* = \phi(\boldsymbol{x}^*)$ is an optimal solution of (MP). If $(\boldsymbol{x}^*, \boldsymbol{y}^*, t^*)$ is an optimal solution of (MP), then $\boldsymbol{x}^*$ is an optimal solution of (P_W) with $\phi(\boldsymbol{x}^*) = t^*$.*

Since we assume that the feasible region X is bounded, there is a point $\boldsymbol{y}^0 \in \mathbb{R}^p$ whose ith component y_i^0 satisfies

$$y_i^0 \leqq \min\{\, y_i \mid \boldsymbol{y} \in Y \,\} = \min\{\, \boldsymbol{c}^i\boldsymbol{x} \mid \boldsymbol{x} \in X \,\}.$$

Then

$$Y_W \subseteqq (\boldsymbol{y}^0 + \mathbb{R}^p_+) \cap \partial Y^{\leqq} \subseteqq (\boldsymbol{y}^0 + \mathbb{R}^p_+) \cap Y^{\leqq}.$$

The key idea of the algorithm is to decompose the truncated lower outcome set $(\boldsymbol{y}^0+\mathbb{R}^p_+)\cap Y^{\leqq}$ into cones K with vertex at $\boldsymbol{y}^0$ and consider the subproblem $(MP(K))$ with variable $\boldsymbol{y}$ restricted to $\partial Y^{\leqq} \cap K$

$$(MP(K)) \quad \left| \begin{array}{ll} \max\limits_{(\boldsymbol{x},\boldsymbol{y},t)} & t \\ \text{s.t.} & t \leqq \phi(\boldsymbol{x}) \\ & \boldsymbol{x} \in X \\ & \boldsymbol{y} = C\boldsymbol{x} \\ & \boldsymbol{y} \in \partial Y^{\leqq} \cap K. \end{array} \right.$$

3.5.2 Relaxation problem $(\overline{MP}(K))$

For a convex function $f : S \to [-\infty, +\infty]$ on a set $S \subseteqq \mathbb{R}^n$,

$$\operatorname{dom} f = \{\, \boldsymbol{x} \in S \mid f(\boldsymbol{x}) < +\infty \,\}. \tag{3.16}$$

is called a *domain of f*. When f is a concave function, the domain of f is $\operatorname{dom} f = \{\, \boldsymbol{x} \in S \mid f(\boldsymbol{x}) > -\infty \,\}$.

There are two things to have done: to replace the concave function ϕ by a function easier to handle, and to construct a polyhedral set containing $Y \cap \partial Y^{\leqq} \cap K$. They propose a piece-wise linear concave function Φ to replace ϕ. Suppose we have a finite number of points $\boldsymbol{x}^1, \cdots, \boldsymbol{x}^k$ in the domain of ϕ and a subgradient $\boldsymbol{s}^i \in \mathbb{R}_n$ of ϕ at $\boldsymbol{x}^i$. Then

$$\Phi(\boldsymbol{x}) = \min\{\, \phi(\boldsymbol{x}^i) + \boldsymbol{s}^i(\boldsymbol{x} - \boldsymbol{x}^i) \mid i = 1, \ldots, k \,\}$$

is a piece-wise linear concave function which overestimates ϕ, i,e., $\Phi(\boldsymbol{x}) \geqq \phi(\boldsymbol{x})$ at any point $\boldsymbol{x}$. Furthermore note that the constraint $t \leqq \phi(\boldsymbol{x})$ with ϕ replaced by Φ is equivalent to the k linear inequality constraints

$$t \leqq \phi(\boldsymbol{x}^i) + \boldsymbol{s}^i(\boldsymbol{x} - \boldsymbol{x}^i) \text{ for } i = 1, \ldots, k.$$

Let $\boldsymbol{r}^1, \cdots, \boldsymbol{r}^p \in \mathbb{R}^p$ be p extreme rays generating the cone $K - \boldsymbol{y}^0$ and for each $i = 1, \ldots, p$ let $\boldsymbol{y}^i$ be the intersection point of the ray $\{\, \boldsymbol{y} \mid \boldsymbol{y} = \boldsymbol{y}^0 + \alpha \boldsymbol{r}^i,\ \alpha \geqq 0 \,\}$ and $\partial Y^{\leqq}$. The intersection point $\boldsymbol{y}^i$ is found by solving the linear program

$$\max\{\, \alpha \mid \boldsymbol{y}^0 + \alpha \boldsymbol{r}^i \leqq C\boldsymbol{x},\ \boldsymbol{x} \in X,\ \alpha \geqq 0 \,\}.$$

Once we have these points $\boldsymbol{y}^1, \cdots, \boldsymbol{y}^p$ and the hyperplane, say H, passing through them, we see the following lemma. See Figure 3.6.

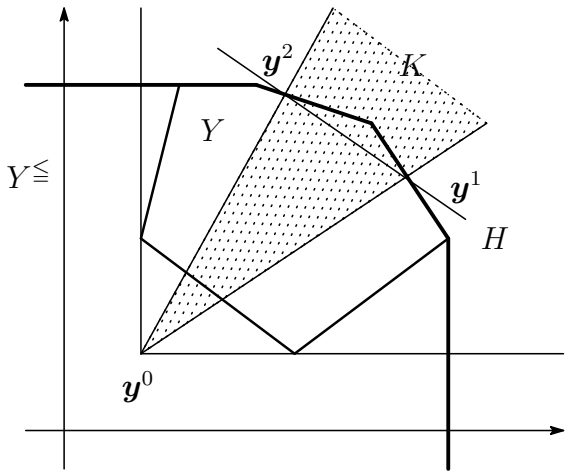


Figure 3.6: Problem $(\overline{MP}(K))$

Lemma 3.24 *Let H^+ be the half space defined by H that does not contain $\boldsymbol{y}^0$. Then*

$$Y \cap \partial Y^{\leqq} \cap K \subseteqq Y \cap Y^{\leqq} \cap K \cap H_+ = Y \cap K \cap H_+.$$

Therefore as a relaxation problem of $(MP(K))$ we obtain

$$(\overline{MP}(K)) \quad \left| \begin{array}{ll} \max\limits_{(\boldsymbol{x},\boldsymbol{y},t)} & t \\ \text{s.t.} & t \leqq \phi(\boldsymbol{x}^i) + \boldsymbol{s}^i(\boldsymbol{x} - \boldsymbol{x}^i) \text{ for } i = 1, \ldots, k \\ & \boldsymbol{x} \in X \\ & \boldsymbol{y} = C\boldsymbol{x} \\ & \boldsymbol{y} \in K \cap H_+. \end{array} \right.$$

Let V be the $p \times p$ matrix consisting of columns $\boldsymbol{y}^1 - \boldsymbol{y}^0, \cdots, \boldsymbol{y}^p - \boldsymbol{y}^0$, then the last two constraints are equivalent to

$$C\boldsymbol{x} = V\boldsymbol{\mu} + \boldsymbol{y}^0, \quad \boldsymbol{e}\boldsymbol{\mu} \geqq 1, \quad \boldsymbol{\mu} \in \mathbb{R}^p_+.$$

Clearly the optimal value of $(\overline{MP}(K))$ provides an upper bound of the optimal value of $(MP(K))$.

3.5.3 Convergence of the algorithm

Once the relaxation problem is so constructed, it will be a routine to make a branch-and-bound algorithm and we omit the description. To guarantee the convergence

(a) the piece-wise linear approximation Φ of ϕ should become better, and

(b) the conical partition should become finer

as the process proceeds. Every time an optimal solution $(\boldsymbol{x}(K), \boldsymbol{y}(K), t(K))$ of Problem $(\overline{MP}(K))$ is obtained, the set of points $\boldsymbol{x}^1, \cdots, \boldsymbol{x}^k$ is incremented by $\boldsymbol{x}(K)$, which improves the approximation accuracy of Φ. Concerning the conical partition, the desired property is referred to as *exhaustiveness* and defined as

Definition 3.25 *The partition procedure is said to be exhaustive when $\bigcap_k K^k$ is a ray for any nested sequence $\{K^k\}_{k=1,\ldots}$ of cones generated by the procedure.*

See Horst-Tuy [27] for a full detail of exhaustiveness.

Theorem 3.26 *Assume that the conical partition procedure is exhaustive. Then every cluster point* $(\boldsymbol{x}^*, \boldsymbol{y}^*, t^*)$ *of the sequence of points* $(\boldsymbol{x}^\nu, \boldsymbol{y}^\nu, t^\nu)$ *generated by the branch-and-bound algorithm is a solution of Master Problem* (MP)*. Hence* $\boldsymbol{x}^*$ *is a solution of* (P_W)*.*

Preliminary computational results are reported in Thoai [59] for linear case. He ran the algorithm on randomly generated test problems with $p = 2$ to 4, $m = 10$ to 50 and $n = 35$ to 250, and reported the average number of iterations, the maximal number of cones stored at an iteration and the average CPU time.

3.6 Bisection Algorithm

This section is devoted to the explanation of the algorithm proposed by Phong-Tuyen [41] for Problem (P_E) with linear objective function $\phi(\boldsymbol{x}) = \boldsymbol{dx}$. The main idea is the bisection method for locating $\phi(P_E)$. Namely, they start with an interval $[\ell_0, u_0]$ which is known to contain $\phi(P_E)$, solve for $\alpha = (\ell_k + u_k)/2$

$$(P_\alpha) \qquad \left| \text{ Find } \boldsymbol{x} \in X_E \quad \text{such that } \boldsymbol{dx} \geqq \alpha \right.$$

and then reduce the interval $[\ell_k, u_k]$ to either $[\alpha, u_k]$ when (P_α) has a solution or $[\ell_k, \alpha]$ when (P_α) has no solution. Thus after a finitely many iterations they obtain an ϵ-approximate solution.

For $\boldsymbol{\lambda} \in \Lambda$ let $\sigma(\boldsymbol{\lambda})$ denote the optimal value of Problem $(SC(\boldsymbol{\lambda}))$, i.e.,

$$\sigma(\boldsymbol{\lambda}) = \max\{\, \boldsymbol{\lambda} C\boldsymbol{x} \mid \boldsymbol{x} \in X \,\}, \tag{3.17}$$

and

$$\tau_\alpha(\boldsymbol{\lambda}) = \max\{\, \boldsymbol{\lambda} C\boldsymbol{x} \mid \boldsymbol{x} \in X,\ \boldsymbol{dx} \geqq \alpha \,\}. \tag{3.18}$$

Since X is the convex hull of its vertex set X_V and an efficient vertex solves Problem $(SC(\boldsymbol{\lambda}))$ for $\boldsymbol{\lambda} \in \Lambda$, we see

Lemma 3.27

(i) $\sigma(\boldsymbol{\lambda}) = \max\{\, \boldsymbol{\lambda} C\boldsymbol{v} \mid \boldsymbol{v} \in X_E \cap X_V \,\}$ *for* $\boldsymbol{\lambda} \in \Lambda$*.*

(ii) $\sigma(\cdot)$ *is a piece-wise linear convex function on* Λ*.*

Proof: From (i) σ is the maximum of finitely many linear functions $\boldsymbol{\lambda} C \boldsymbol{v}$ each of which corresponds to a vertex $\boldsymbol{v}$ of $X_E \cap X_V$. Thus it is piecewise linear convex. □

In the same way we obtain

Lemma 3.28

(i) $\tau_\alpha(\boldsymbol{\lambda}) = \max\{\, \boldsymbol{\lambda} C \boldsymbol{v} \mid \boldsymbol{v}$ *is an efficient vertex of* $X \cap \{\, \boldsymbol{x} \mid \boldsymbol{dx} \geqq \alpha \,\}\,\}$.

(ii) $\tau_\alpha(\boldsymbol{\lambda}) \leqq \sigma(\boldsymbol{\lambda})$ *for any* $\boldsymbol{\lambda} \in \mathbb{R}_p$.

(iii) $\tau_\alpha(\cdot)$ *is a piece-wise linear convex function on* Λ.

(iv) $\tau_\alpha(\boldsymbol{\lambda})$ *is a nonincreasing function in* $\alpha \in \mathbb{R}$.

Let us denote the *epigraph of* σ by $\operatorname{epi}\sigma$, i.e.,

$$\operatorname{epi}\sigma = \{\, (\boldsymbol{\lambda}, \mu) \mid (\boldsymbol{\lambda}, \mu) \in \Lambda \times \mathbb{R},\ \sigma(\boldsymbol{\lambda}) \leqq \mu \,\}. \tag{3.19}$$

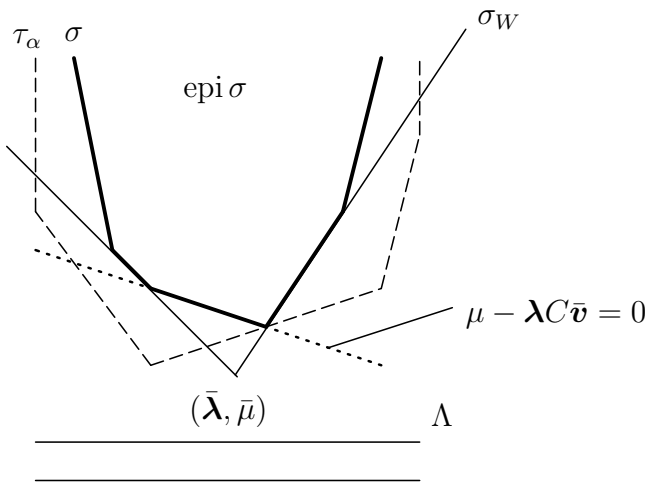


Figure 3.7: Functions σ and τ_α

For the existence of a solution of (P_α) we have the following theorem.

Theorem 3.29

(i) $X_E \cap \{\, \boldsymbol{x} \mid \boldsymbol{dx} \geqq \alpha \,\} \neq \emptyset$ *if and only if* $\sigma(\boldsymbol{\lambda}) = \tau_\alpha(\boldsymbol{\lambda})$ *for some* $\boldsymbol{\lambda} \in \Lambda$.

(ii) $\sigma(\boldsymbol{\lambda}) = \tau_\alpha(\boldsymbol{\lambda})$ *for some* $\boldsymbol{\lambda} \in \Lambda$ *if and only if there is a vertex* $(\bar{\boldsymbol{\lambda}}, \bar{\mu})$ *of* $epi\,\sigma$ *such that* $\bar{\mu} = \tau_\alpha(\bar{\boldsymbol{\lambda}})$.

Proof: We show only the first assertion because the second assertion is clear from the piece-wise linearity of σ and the fact that $\tau_\alpha \leqq \sigma$. Suppose $\boldsymbol{x} \in X_E \cap \{\, \boldsymbol{x} \mid \boldsymbol{dx} \geqq \alpha \,\}$, then $\sigma(\boldsymbol{\lambda}) = \boldsymbol{\lambda} C \boldsymbol{x}$ for some $\boldsymbol{\lambda} \in \Lambda$. Since $\boldsymbol{dx} \geqq \alpha$, $\boldsymbol{\lambda} C \boldsymbol{x} \leqq \tau_\alpha(\boldsymbol{\lambda}) \leqq \sigma(\boldsymbol{\lambda})$. Therefore $\sigma(\boldsymbol{\lambda}) = \tau_\alpha(\boldsymbol{\lambda})$. Suppose $\sigma(\boldsymbol{\lambda}) = \tau_\alpha(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda} \in \Lambda$ and let $\boldsymbol{x}$ be a point that attains $\max\{\, \boldsymbol{\lambda} C \boldsymbol{x} \mid \boldsymbol{x} \in X,\ \boldsymbol{dx} \geqq \alpha \,\} = \tau_\alpha(\boldsymbol{\lambda})$. Then, since $\sigma(\boldsymbol{\lambda}) = \tau_\alpha(\boldsymbol{\lambda})$, $\boldsymbol{x}$ maximizes $\boldsymbol{\lambda} C \boldsymbol{x}$ over X, meaning $\boldsymbol{x} \in X_E$. □

Now let W be a nonempty subset of $X_E \cap X_V$ and let

$$\sigma_W(\boldsymbol{\lambda}) = \max\{\, \boldsymbol{\lambda} C \boldsymbol{v} \mid \boldsymbol{v} \in W \,\}. \tag{3.20}$$

Then for any $\boldsymbol{\lambda} \in \Lambda$

$$\sigma_W(\boldsymbol{\lambda}) \leqq \sigma(\boldsymbol{\lambda}) \tag{3.21}$$

and we have the following corollary from Theorem 3.29 and the piece-wise linearity of $\sigma_W(\boldsymbol{\lambda})$.

Corollary 3.30

(i) $\tau_\alpha(\boldsymbol{\lambda}) < \sigma_W(\boldsymbol{\lambda})$ *for any* $\boldsymbol{\lambda} \in \Lambda$, *then* $X_E \cap \{\, \boldsymbol{x} \mid \boldsymbol{dx} \geqq \alpha \,\} = \emptyset$.

(ii) $\tau_\alpha(\boldsymbol{\lambda}) \geqq \sigma_W(\boldsymbol{\lambda})$ *for some* $\boldsymbol{\lambda} \in \Lambda$ *if and only if there is a vertex* $(\bar{\boldsymbol{\lambda}}, \bar{\mu})$ *of* $epi\,\sigma_W$ *such that* $\bar{\mu} \leqq \tau_\alpha(\bar{\boldsymbol{\lambda}})$.

This corollary means that we can check whether $\tau_\alpha(\boldsymbol{\lambda}) = \sigma_W(\boldsymbol{\lambda})$ at some $\boldsymbol{\lambda} \in \Lambda$ by evaluating $\tau_\alpha(\bar{\boldsymbol{\lambda}})$ at vertices $(\bar{\boldsymbol{\lambda}}, \bar{\mu})$ of $\operatorname{epi} \sigma_W$. If $\tau_\alpha(\bar{\boldsymbol{\lambda}}) < \bar{\mu}$ for every vertex $(\bar{\boldsymbol{\lambda}}, \bar{\mu})$, we conclude that $\tau_\alpha < \sigma$, and hence $X_E \cap \{\, \boldsymbol{x} \mid \boldsymbol{dx} \geqq \alpha \,\} = \emptyset$ by (i) of Theorem 3.29. Otherwise, i.e., we have found a vertex $(\bar{\boldsymbol{\lambda}}, \bar{\mu})$ with $\tau_\alpha(\bar{\boldsymbol{\lambda}}) \geqq \bar{\mu}$. Two possible cases occur. If $\sigma(\bar{\boldsymbol{\lambda}}) \leqq \bar{\mu}$, implying $\sigma(\bar{\boldsymbol{\lambda}}) = \bar{\mu} = \tau_\alpha(\bar{\boldsymbol{\lambda}})$, we see that $X_E \cap \{\, \boldsymbol{x} \mid \boldsymbol{dx} \geqq \alpha \,\} \neq \emptyset$ by Theorem 3.29. If $\sigma(\bar{\boldsymbol{\lambda}}) > \bar{\mu}$, a vertex $\bar{\boldsymbol{v}}$ of X that attains $\max\{\, \bar{\boldsymbol{\lambda}} C \boldsymbol{x} \mid \boldsymbol{x} \in X \,\}$ is not in W. Then W is incremented by this vertex $\bar{\boldsymbol{v}}$ to make a better underestimation $\sigma_{W \cup \{\bar{\boldsymbol{v}}\}}$ of σ.

Lemma 3.31 *The above procedure terminates after a finite number of incrementation of* W *and shows whether* $X_E \cap \{\, \boldsymbol{x} \mid \boldsymbol{dx} \geqq \alpha \,\}$ *is empty or not.*

Proof: Clear from the finiteness of the vertices of X. □

The main technique used in the procedure is generating the vertex set of epi $\sigma_{W\cup\{\bar{\boldsymbol{v}}\}}$ from that of epi σ_W. Note first that epi σ_W is represented by finitely many linear inequalities each of which corresponds to a vertex of W:

$$\text{epi}\,\sigma_W = \{\,(\boldsymbol{\lambda},\mu) \mid (\boldsymbol{\lambda},\mu) \in \Lambda \times \mathbb{R},\ \mu - \boldsymbol{\lambda}C\boldsymbol{v} \geqq 0 \text{ for } \boldsymbol{v} \in W\,\}.$$

Suppose that we have known the vertex set of epi σ_W, the second case above occurs and we find a vertex $\bar{\boldsymbol{v}}$ of X by maximizing $\bar{\boldsymbol{\lambda}}C\boldsymbol{x}$ over X. This vertex will add an inequality $\mu - \boldsymbol{\lambda}C\bar{\boldsymbol{v}} \geqq 0$, which cuts off the vertex $(\bar{\boldsymbol{\lambda}},\bar{\mu})$ of epi σ_W. To generate the vertex set of epi $\sigma_{W\cup\{\bar{\boldsymbol{v}}\}}$ we have only to generate the vertex set of $(\text{epi}\,\sigma_W)\cap\{\,(\boldsymbol{\lambda},\mu) \mid \mu - \boldsymbol{\lambda}C\bar{\boldsymbol{v}} = 0\,\}$. There have been proposed a lot of algorithms for this purpose, e.g., Horst-Vries-Thoai [25], Chen-Hansen-Jaumard [14], and Thieu-Tam-Ban [57]. See also Section4.2, Chapter II of Horst-Tuy [27].

For a given tolerance $\epsilon > 0$ after finitely many bisections we obtain an interval $[\ell_k, u_k]$ such that (P_{u_k}) has no solution while (P_{ℓ_k}) has a solution together with $\bar{\boldsymbol{\lambda}} \in \Lambda$ at which σ coincides with τ_{ℓ_k}. Then solve $\max\{\,\bar{\boldsymbol{\lambda}}C\boldsymbol{x} \mid \boldsymbol{x} \in X,\ \boldsymbol{dx} \geqq \ell_k\,\}$ to obtain $\boldsymbol{x}^*$. This is an ϵ-approximate solution of Problem (P_E), i.e., $\boldsymbol{x}^* \in X_E$ and $\boldsymbol{dx}^* \geqq \boldsymbol{dx} - \epsilon$ for any $\boldsymbol{x} \in X_E$.

It is reported in Phong-Tuyen [41] that an illustrative example of $p = 2, n = 3, m = 4$ required 11 iterations for $\epsilon = 0.1$.

3.7 Face Search Algorithm

In this section we introduce the algorithm for Problem (P_E) proposed by Sayin [46], which is based on the enumeration method of efficient faces in Sayin [45].

For a point $\boldsymbol{x} \in X$ let $I(\boldsymbol{x})$ be the index set of zero components of $\boldsymbol{x}$, i.e., $I(\boldsymbol{x}) = \{\, i \in \{1,\ldots,n\} \mid x_i = 0\,\}$. For $I \subseteqq \{1,\ldots,n\}$ let

$$F(I) = \{\,\boldsymbol{x} \in X \mid x_i = 0 \text{ for } i \in I\,\},$$

which is a, possibly vacant, face of X. Then the efficient set X_E is decomposed as

$$X_E = \bigcup_{I\subseteqq\{1,\ldots,n\}} (X_E \cap F(I)).$$

Therefore Problem (P_E) reduces to the family of following problems

$$(P_E(I)) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in X_E \cap F(I), \end{array} \right.$$

each of which is corresponding to $I \subseteqq \{1, \ldots, n\}$. For a mutually disjoint decomposition of X_E see Corollary 3.3 in Benson [10]. Since $X_E \cap F(I) \subseteqq X \cap F(I) = F(I)$,

$$(\overline{P}_E(I)) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in F(I), \end{array} \right.$$

is a relaxation problem of $(P_E(I))$. Note that this is a linear program when ϕ is a linear function.

Suppose we have at hand an incumbent, i.e., a point $\boldsymbol{x}^* \in X_E$, and the list of problems $(P_E(I))$ to solve. At the beginning the list consists of the single problem $(P_E(\emptyset))$, which is identical to (P_E) since $F(\emptyset) = X$. Choosing a problem $(P_E(I))$ on the list and solving its relaxation $(\overline{P}_E(I))$, we have the following cases.

1. $(\overline{P}_E(I))$ is infeasible: Problem $(P_E(I))$ is fathomed and deleted from the list.
2. $(\overline{P}_E(I))$ has an optimal solution $\boldsymbol{x}$.
   (a) $\phi(\boldsymbol{x}) < \phi(\boldsymbol{x}^*)$: Problem $(P_E(I))$ is fathomed and deleted from the list.
   (b) $\phi(\boldsymbol{x}) > \phi(\boldsymbol{x}^*)$:
      i. $\boldsymbol{x} \in X_E$: The incumbent is updated as $\boldsymbol{x}^* = \boldsymbol{x}$, and Problem $(P_E(I))$ is fathomed and deleted from the list.
      ii. $x \notin X_E$: Problem $(P_E(I))$ is fathomed and deleted from the list, and for each index $k \in \{1, \ldots, n\} \setminus I$ Problem $(P_E(I \cup \{k\}))$ is added to the list.

The last case where $\boldsymbol{x} \notin X_E$ may need an explanation. We see from Theorem 3.5 that no point in the relative interior of $F(I)$ is efficient in this case. Since any point in the relative boundary of $F(I)$ belongs to $F(I \cup \{k\})$ for some $k \in \{1, \ldots, n\} \setminus I$, Problem $(P_E(I))$ is fathomed and can be deleted from the list.

In the case of $\boldsymbol{x} \notin X_E$, if $x_k = 0$, it remains optimal to Problem $(\overline{P}_E(I \cup \{k\}))$, which therefore needs not be solved. Even if this is not the case, Problem $(\overline{P}_E(I \cup$

$\{k\}))$ differs slightly from $(\overline{P}_E(I))$.

The key issue of implementation would be the list-management as it is always the case in the branch-and-bound method. Especially, a subset $I = \{i_1, \ldots, i_\ell\}$ of $\{1, \ldots, n\}$ would be generated from each of $\{i_2, \ldots, i_\ell\}$, $\{i_1, i_3, \ldots, i_\ell\}$, $\cdots$, and $\{i_1, \ldots, i_{\ell-1}\}$. The redundancy can be avoided by a simple technique. Even incorporating the technique, the list grows very rapidly and becomes too large to keep in the memory. Due to the rapid growth of problem list, the computational experiment reported in Sayin [46] is restricted in problem size.

3.8 Lagrangian Relaxation Based Approach

White [64] considered Problem (P_E) with linear function $\phi(\boldsymbol{x}) = \boldsymbol{d}\boldsymbol{x}$ and presented several equivalent formulations. Dauer-Fosnaugh [16] considered the problem with quasi-convex function ϕ and showed a way of converting it to a bicriteria problem, which could be viewed as a Lagrangian relaxation of Problem (P_E). An-Tao-Muu [3] showed that there is no duality gap for a sufficiently large Lagrangian multiplier. We will explain the common idea in terms of the Lagrangian relaxation method. The central role will be played by the gap function of (3.4), i.e.,

$$g(\boldsymbol{x}) = \max\{\, \boldsymbol{e}C\boldsymbol{x}' \mid \boldsymbol{x}' \in X,\ C\boldsymbol{x}' \geqq C\boldsymbol{x} \,\} - \boldsymbol{e}C\boldsymbol{x}.$$

We call a point $\boldsymbol{x}'$ that attains the maximum above a *projected point* of $\boldsymbol{x}$. It is easily seen from the theory of parametric linear program that g is a piece-wise linear concave function on X. As stated in Theorem 3.5 $g(\boldsymbol{x}) \geqq 0$ for $\boldsymbol{x} \in X$, and $\boldsymbol{x} \in X_E$ if and only if $g(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in X$. See Theorem 4.1 of Benson [10]. Thus Problem (P_E) is reformulated as follows:

$$(P_E) \quad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in X \\ & g(\boldsymbol{x}) \leqq 0. \end{array} \right.$$

Note that the last constraint $g(\boldsymbol{x}) \leqq 0$ is a reverse convex constraint, which has been attracting attention. See e.g., Horst-Tuy [27] and Tuy [62]. To solve Problem (P_E) we combine the objective function $\phi(\boldsymbol{x})$ with the constraint $g(\boldsymbol{x}) \leqq 0$ multiplied by a Lagrangian multiplier $\pi \geqq 0$ to have the *Lagrangian relaxation problem*

$$(Q(\pi)) \qquad \left| \begin{array}{ll} z(\pi) = & \max\limits_{\boldsymbol{x}} \phi(\boldsymbol{x}) - \pi g(\boldsymbol{x}) \\ & \text{s.t.} \quad \boldsymbol{x} \in X. \end{array} \right.$$

In the sequel $\boldsymbol{x}(\pi)$ denotes an optimal solution of $(Q(\pi))$ and $\boldsymbol{x}'(\pi)$ denotes its projected point. Note that $(Q(\pi))$ is a quasi-convex maximization and that the optimality is always attained at a vertex of X. For we assume that X is a polytope, we reformulate Problem $(Q(\pi))$ in terms of the vertices of X and obtain

$$z(\pi) = \max\{\, \phi(\boldsymbol{v}) - \pi g(\boldsymbol{v}) \mid \boldsymbol{v} \in X_V \,\}. \tag{3.22}$$

Note that for each vertex $\boldsymbol{v} \in X_V$ the function $\phi(\boldsymbol{v}) - \pi g(\boldsymbol{v})$ is a linear function with nonpositive slope in variable π. These linear functions as well as $z(\pi)$ are shown in Figure 3.8, in which $z(\pi)$ is depicted by a bold piece-wise linear line. Notice that horizontal lines, meaning $g(\boldsymbol{v}) = 0$, correspond to vertices in X_E.

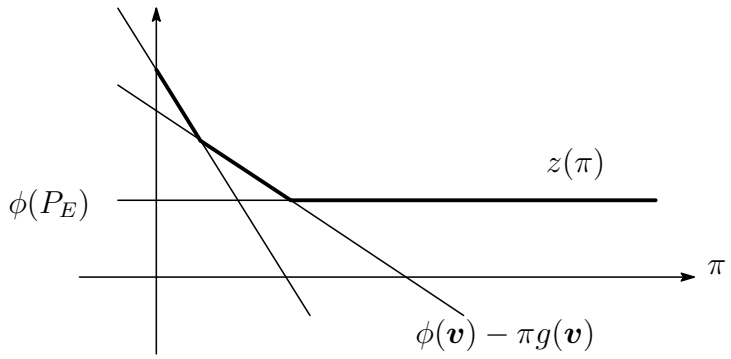


Figure 3.8: Functions $z(\pi)$ and $\phi(\boldsymbol{v}) - \pi g(\boldsymbol{v})$

Though the following lemmas are straightforward from this observation, brief proofs will be given.

Lemma 3.32 *If $g(\boldsymbol{x}(\pi)) = 0$ for some $\pi \geqq 0$, then $\boldsymbol{x}(\pi)$ is an optimal solution of (P_E).*

Proof: For any $\boldsymbol{x}$ in X_E, we readily see $\phi(\boldsymbol{x}(\pi)) = \phi(\boldsymbol{x}(\pi)) - \pi g(\boldsymbol{x}(\pi)) \geqq \phi(\boldsymbol{x}) - \pi g(\boldsymbol{x}) = \phi(\boldsymbol{x})$. □

Concerning $z(\pi)$ we have the following property.

Lemma 3.33 *Let $0 \leqq \pi \leqq \pi'$ and let $\boldsymbol{x}'(\pi)$ be a projected point of $\boldsymbol{x}(\pi)$. Then*

$$\phi(\boldsymbol{x}'(\pi)) \leqq \phi(P_E) \leqq z(\pi') \leqq z(\pi).$$

Proof: Since the projected point lies in X_E, the first inequality is trivial. By the definition of $z(\pi')$, it holds that $\phi(\boldsymbol{x}) - \pi' g(\boldsymbol{x}) \leqq z(\pi')$ for any $\boldsymbol{x} \in X$ and also for any $\boldsymbol{x} \in X_E$. Then we see $\phi(\boldsymbol{x}) \leqq z(\pi')$ for any $\boldsymbol{x} \in X_E$, which implies the second inequality. The last inequality is derived from

$$\begin{aligned} z(\pi') &= \phi(\boldsymbol{x}(\pi')) - \pi' g(\boldsymbol{x}(\pi')) \\ &\leqq \phi(\boldsymbol{x}(\pi')) - \pi g(\boldsymbol{x}(\pi')) \\ &\leqq \phi(\boldsymbol{x}(\pi)) - \pi g(\boldsymbol{x}(\pi)) = z(\pi). \end{aligned}$$

□

This lemma means that $z(\pi)$ gives an upper bound of $\phi(P_E)$ and also $\boldsymbol{x}'(\pi)$, the projected point of $\boldsymbol{x}(\pi)$, gives a lower bound. Above two lemmas suggest that solution $\boldsymbol{x}(\pi)$ of $(Q(\pi))$ for a sufficiently large $\pi > 0$ solves Problem (P_E). In fact, because of the finiteness of X_V we readily see the following theorem. See Figure 3.8.

Theorem 3.34 *There is a $\pi^* > 0$ such that for any $\pi > \pi^*$ $\boldsymbol{x}(\pi)$ is an optimal solution of (P_E).*

An-Tao-Muu showed the same result for a convex function ϕ in Lemma 4 of [3]. Dauer-Fosnaugh [16] showed that $z(\pi)$ converges to $\phi(P_E)$ as π goes to infinity for a more general setting.

Muu [35] reduces the variables of the gap function by using Lemma 3.4. Let r be the rank of C and without loss of generality we assume that the first r rows $\boldsymbol{c}^1, \cdots, \boldsymbol{c}^r$ are linearly independent. Let L be the range space of matrix $C^\top$ and $L^\perp$ be its orthogonal complement in $\mathbb{R}^n$ and suppose we have a basis $\boldsymbol{b}^{r+1}, \cdots, \boldsymbol{b}^n$ of $L^\perp$. Then any $\boldsymbol{x} \in \mathbb{R}^n$ is uniquely represented as $\boldsymbol{x} = \overline{C}^\top \boldsymbol{\alpha} + B\boldsymbol{\beta}$ for $\boldsymbol{\alpha} \in \mathbb{R}^r$ and $\boldsymbol{\beta} \in \mathbb{R}^{n-r}$, where $\overline{C}$ is the matrix of rows $\boldsymbol{c}^1, \cdots, \boldsymbol{c}^r$ and B is the matrix of columns $\boldsymbol{b}^{r+1}, \cdots, \boldsymbol{b}^n$. Then Problem $(Q(\pi))$ is rewritten as

$$\left| \begin{array}{lll} z(\pi) = & \max\limits_{(\boldsymbol{\alpha},\boldsymbol{\beta})} & \phi(\overline{C}^\top \boldsymbol{\alpha} + B\boldsymbol{\beta}) - \pi g(\overline{C}^\top \boldsymbol{\alpha} + B\boldsymbol{\beta}) \\ & \text{s.t.} & \overline{C}^\top \boldsymbol{\alpha} + B\boldsymbol{\beta} \in X. \end{array} \right.$$

We see, however, by Lemma 3.4 that

$$g(\overline{C}^\top \boldsymbol{\alpha} + B\boldsymbol{\beta}) = g(\overline{C}^\top \boldsymbol{\alpha}), \tag{3.23}$$

then Problem $(Q(\pi))$ is

$$\left| \begin{array}{lll} z(\pi) = & \max\limits_{(\boldsymbol{\alpha},\boldsymbol{\beta})} & \phi(\overline{C}^\top \boldsymbol{\alpha} + B\boldsymbol{\beta}) - \pi g(\overline{C}^\top \boldsymbol{\alpha}) \\ & \text{s.t.} & A\overline{C}^\top \boldsymbol{\alpha} + AB\boldsymbol{\beta} = b \\ & & \overline{C}^\top \boldsymbol{\alpha} + B\boldsymbol{\beta} \geqq 0. \end{array} \right.$$

The rank r of C is no more than p, which is usually much smaller than n. When ϕ is a linear function, the above problem contains a small number of nonconvex variables.

Dauer-Fosnaugh [16] also showed that when ϕ is a linear function $\boldsymbol{dx}$ and $\boldsymbol{d}$ is a linear combination of rows $\boldsymbol{c}^i$'s of C, i.e., $\boldsymbol{d} = \boldsymbol{\gamma} C$ for some $\boldsymbol{\gamma} \in \mathbb{R}_p$, the π^* in Theorem 3.34 is given by $\|\boldsymbol{\gamma}\|_\infty$. Notice that this value is 1 if $\boldsymbol{d} = \pm \boldsymbol{c}^i$ for some $i = 1, \ldots, p$. Muu [35] generalized this result to the nonlinear case where $\phi(\boldsymbol{x})$ is given by $\varphi(C\boldsymbol{x})$ for some function φ.

The transformation of Problem (P_E) by White [64] is based on Theorem 3.5. Note that Problem (P_E) is equivalent to

$$\max \left\{ \phi(\boldsymbol{x}) \;\middle|\; \begin{array}{l} \boldsymbol{x} \in X,\ \boldsymbol{\lambda} \in \Lambda,\ \boldsymbol{\mu} \in \mathbb{R}_m, \\ \boldsymbol{\mu} A - \boldsymbol{\lambda} C \geqq \mathbf{0},\ \boldsymbol{\lambda} C\boldsymbol{x} - \boldsymbol{\mu}\boldsymbol{b} = 0 \end{array} \right\}.$$

By multiplying the bilinear constraint $\boldsymbol{\lambda} C\boldsymbol{x} - \boldsymbol{\mu}\boldsymbol{b} = 0$ by π we have its Lagrangian relaxation

$$\max \left\{ \phi(\boldsymbol{x}) + \pi(\boldsymbol{\lambda} C\boldsymbol{x} - \boldsymbol{\mu}\boldsymbol{b}) \;\middle|\; \begin{array}{l} \boldsymbol{x} \in X,\ \boldsymbol{\lambda} \in \Lambda,\ \boldsymbol{\mu} \in \mathbb{R}_m, \\ \boldsymbol{\mu} A - \boldsymbol{\lambda} C \geqq \mathbf{0} \end{array} \right\},$$

which is to maximize a bilinear objective function under linear inequality constraints. Several properties of this relaxation are discussed in White [64].

3.9 Dual Approach

Nonconvex duality is one of the most promising subject in the global optimization. We will not go into details of the duality theory in this paper. The readers who are

interested in it should refer Atteia-El-Qortobi [6] and Thach [53–55]. In this section we will briefly explain the dual approach of Thach-Konno-Yokota [56].

3.9.1 Dual problem of (P_E)

Let

$$C^{\leqq} = \{\, \boldsymbol{y} \in \mathbb{R}^n \mid C\boldsymbol{y} \leqq \boldsymbol{0},\ \boldsymbol{c}^i\boldsymbol{y} < 0 \text{ for some } i = 1, \ldots, p \,\}.$$

Then the efficient set X_E is written as the difference of two convex sets. See Figure 3.9.

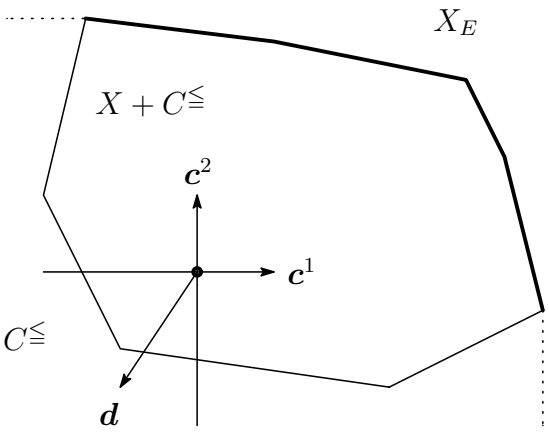


Figure 3.9: Set X_E equals $X \setminus (X + C^{\leqq})$

Lemma 3.35 $X_E = X \setminus (X + C^{\leqq})$.

Proof: We see that

$$\begin{aligned} X_E &= \{\, \boldsymbol{x} \in X \mid \nexists \boldsymbol{x}' : C\boldsymbol{x}' \geqq C\boldsymbol{x}, \boldsymbol{c}^i\boldsymbol{x}' > \boldsymbol{c}^i\boldsymbol{x} \text{ for some } i \,\} \\ &= X \setminus \{\, \boldsymbol{x} \mid \exists \boldsymbol{x}' : C(\boldsymbol{x} - \boldsymbol{x}') \leqq \boldsymbol{0}, \boldsymbol{c}^i(\boldsymbol{x} - \boldsymbol{x}') < 0 \text{ for some } i \,\} \\ &= X \setminus \{\, \boldsymbol{x} + \boldsymbol{y} \mid \boldsymbol{x} \in X,\ C\boldsymbol{y} \leqq 0,\ \boldsymbol{c}^i\boldsymbol{y} < 0 \text{ for some } i \,\} \\ &= X \setminus (X + C^{\leqq}). \end{aligned}$$

□

Then Problem (P_E) is written as

$$(P_E) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in X \setminus (X + C^{\leqq}). \end{array} \right.$$

Since X is now assumed to be a polytope, we show that the set $X + C^{\leqq}$ can be replaced by the interior of a closed convex set. Let E be the $p \times p$ matrix all of whose elements are unity, and for a positive parameter s define a $p \times p$ matrix C_s, sets $C_s^{\leqq}$ and X_s by

$$\begin{aligned} C_s &= (I + sE)C \\ C_s^{\leqq} &= \{\, \boldsymbol{y} \mid C_s \boldsymbol{y} \leqq \boldsymbol{0} \,\} \\ X_s &= X \setminus \operatorname{int}(X + C_s^{\leqq}), \end{aligned}$$

where I is the $p \times p$ identity matrix. Note that X_s is also the difference of two convex sets.

Lemma 3.36

$$X_s = \left\{ \boldsymbol{x} \in X \;\middle|\; \begin{array}{l} \exists \boldsymbol{\lambda} \in \mathbb{R}_{p+} \setminus \{\boldsymbol{0}\} : \\ \boldsymbol{\lambda} C_s \boldsymbol{x} \geqq \boldsymbol{\lambda} C_s \boldsymbol{x}' \textit{ for all } \boldsymbol{x}' \in X \end{array} \right\}.$$

Proof: Let $\boldsymbol{x}$ be a point in X_s. By the separation theorem, there is a $\boldsymbol{v} \neq \boldsymbol{0}$ satisfying $\boldsymbol{v}\boldsymbol{x} \geqq \boldsymbol{v}\boldsymbol{z}$ for all $\boldsymbol{z} \in X + C_s^{\leqq}$. Hence $\boldsymbol{v}\boldsymbol{x} \geqq \boldsymbol{v}(\boldsymbol{x} + \boldsymbol{y})$ holds for all $\boldsymbol{y}$ such that $C_s \boldsymbol{y} \leqq 0$. Applying Farkas' alternative theorem, we have $\boldsymbol{v} = \boldsymbol{\lambda} C_s$ for some $\boldsymbol{\lambda} \in \mathbb{R}_{p+} \setminus \{\boldsymbol{0}\}$, and hence $\boldsymbol{\lambda} C_s \boldsymbol{x} \geqq \boldsymbol{\lambda} C_s \boldsymbol{z}$ holds for all $\boldsymbol{z} \in X + C_s^{\leqq}$. Noting that $\boldsymbol{0} \in C_s^{\leqq}$ we see that $\boldsymbol{\lambda} C_s \boldsymbol{x} \geqq \boldsymbol{\lambda} C_s \boldsymbol{x}'$ for all $\boldsymbol{x}' \in X$, and hence $\boldsymbol{x}$ is contained in the set on the right side.

Suppose $\boldsymbol{x}$ maximizes $\boldsymbol{\lambda} C_s \boldsymbol{x}$ over X for some $\boldsymbol{\lambda} \in \mathbb{R}_{p+} \setminus \{\boldsymbol{0}\}$. Then clearly it also maximizes $\boldsymbol{\lambda} C_s \boldsymbol{x}$ over $X + C_s^{\leqq}$ and does not lie in the interior of $X + C_s^{\leqq}$. □

By this lemma we see that X_s coincides with X_E when s is sufficiently small.

Lemma 3.37 *There is an $\hat{s} > 0$ such that $X_s = X_E$ if $0 < s < \hat{s}$.*

Proof: To show that $X_s \subseteqq X_E$, choose arbitrarily $\boldsymbol{x} \in X_s$. Then by the above lemma, there is a $\boldsymbol{\lambda} \in \mathbb{R}_{p+} \setminus \{\boldsymbol{0}\}$ such that $\boldsymbol{x}$ maximizes $\boldsymbol{\lambda} C_s \boldsymbol{x}$ over X. Here we

assume that $\boldsymbol{\lambda}\mathbf{1} = 1$ without loss of generality. Substituting the definition for C_s, we see $\boldsymbol{\lambda}C_s = (\boldsymbol{\lambda} + s\boldsymbol{e})C$. This and the equality

$$X_E = \{\, \boldsymbol{x} \in X \mid \exists \boldsymbol{\lambda} \in \mathbb{R}_{p++} : \boldsymbol{\lambda}C\boldsymbol{x} \geqq \boldsymbol{\lambda}C\boldsymbol{x}' \text{ for all } \boldsymbol{x}' \in X \,\}$$

of Theorem 3.5 imply that $\boldsymbol{x} \in X_E$.

By Theorem 3.5 X_E is the union of finitely many faces $F^1, \cdots, F^L$ of X such that F^ℓ is the optimal set of maximizing $\boldsymbol{\lambda}^\ell C\boldsymbol{x}$ over X for some $\boldsymbol{\lambda}^\ell \in \mathbb{R}_{p++}$ such that $\boldsymbol{\lambda}^\ell\mathbf{1} = 1$. Let $\hat{s} = \min\{\, \lambda_i^\ell/(1 - p\lambda_i^\ell) \mid \lambda_i^\ell < 1/p \,\}$ and choose s such that $0 < s < \hat{s}$. Then $s/(1 + sp) < \lambda_i^\ell$ for all $\ell = 1, \ldots, L$ and $i = 1, \ldots, p$. Let $\theta_i^\ell = \lambda_i^\ell - \frac{s}{1+sp}$ for $\ell = 1, \ldots, L, i = 1, \ldots, p$. Then we readily see that $\theta_i^\ell > 0$ and

$$\boldsymbol{\lambda}^\ell C = \theta^\ell C_s.$$

This means that $F^\ell \subseteqq X_s$ by Lemma 3.36, and hence $X_E \subseteqq X_s$. □

We assume hereafter that $0 < s < \hat{s}$. Then Problem (P_E) is equivalently rewritten as

$$(P_E) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in X_s = X \setminus \operatorname{int}(X + C_s^{\leqq}). \end{array} \right.$$

For $\boldsymbol{v} \in \mathbb{R}_n$ let

$$\xi(\boldsymbol{v}) = \sup\{\phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X,\ \boldsymbol{v}\boldsymbol{x} \geqq 1\,\},$$

where $\xi(\boldsymbol{v}) = -\infty$ when $\{\, \boldsymbol{x} \in X \mid \boldsymbol{v}\boldsymbol{x} \geqq 1 \,\} = \emptyset$.

Definition 3.38 *For $Z \subseteqq \mathbb{R}^n$ the set $\{\, \boldsymbol{v} \in \mathbb{R}_n \mid \boldsymbol{v}\boldsymbol{x} \leqq 1$ for all $\boldsymbol{x} \in Z \,\}$ is called the polar set of Z and denoted by Z°.*

See e.g., Section 2.14 of Stoer-Witzgall [51], and Section E of Chapter 11 in Rockafellar-Wets [43] for the properties of polar set. We here assume that $\mathbf{0} \in \operatorname{int} X$, $\operatorname{int} C_s^{\leqq} \neq \emptyset$ and ϕ is a concave function. Then by the nonconvex duality theory of Thach [53] we obtain the following duality theorem between Problem (P_E) and its dual problem

$$(D_s) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{v}} & \xi(\boldsymbol{v}) \\ \text{s.t.} & \boldsymbol{v} \in (X + C_s^{\leqq})^\circ. \end{array} \right.$$

Theorem 3.39 *Let $\xi(D_s)$ denote the optimal value of (D_s), then*

$$\phi(P_E) = \xi(D_s).$$

Proof: See Thach [53] and Chapter 4 of Konno-Thach-Tuy [33]. □

Since $\mathbf{0} \in \operatorname{int} X$, $(X + C_s^{\leqq})^\circ \subseteqq (C_s^{\leqq})^\circ$, which is identical to $\{\, \boldsymbol{\gamma} C_s \mid \boldsymbol{\gamma} \in \mathbb{R}_{p+} \,\}$. Therefore $\boldsymbol{v} \in (X+C_s^{\leqq})^\circ$ if and only if $\boldsymbol{v} = \boldsymbol{\gamma} C_s$ for some $\boldsymbol{\gamma} \in \mathbb{R}_{p+}$ and $\sup\{\, \boldsymbol{v}(\boldsymbol{x}+\boldsymbol{y}) \mid \boldsymbol{x} \in X,\ \boldsymbol{y} \in C_s^{\leqq} \,\} \leqq 1$. The latter condition can be replaced by $\sup\{\, \boldsymbol{v}\boldsymbol{x} \mid \boldsymbol{x} \in X \,\} \leqq 1$ from the definition of $C_s^{\leqq}$ and $\boldsymbol{v} = \boldsymbol{\gamma} C_s$. Letting

$$\Gamma = \{\, \boldsymbol{\gamma} \in \mathbb{R}_{p+} \mid \sup_{\boldsymbol{x} \in X} \boldsymbol{\gamma} C_s \boldsymbol{x} \leqq 1 \,\},$$

we have

$$(X + C_s^{\leqq})^\circ = \{\, \boldsymbol{\gamma} C_s \mid \boldsymbol{\gamma} \in \Gamma \,\}.$$

Now let

$$\Xi(\boldsymbol{\gamma}) = \sup\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X,\ \boldsymbol{\gamma} C_s \boldsymbol{x} \geqq 1 \,\}.$$

The above argument yields an equivalent form of (D_s) in variable $\boldsymbol{\gamma} \in \mathbb{R}_p$.

Theorem 3.40 *Problem (D_s) is equivalent to*

$$\left|\begin{array}{ll} \max\limits_{\boldsymbol{\gamma}} & \Xi(\boldsymbol{\gamma}) \\ s.t. & \boldsymbol{\gamma} \in \Gamma. \end{array}\right.$$

We will see that this problem is a quasi-convex maximization over a convex polyhedral set.

Lemma 3.41

(i) Γ *is a convex polyhedral subset of* $\mathbb{R}_p$.

(ii) Ξ *is a quasi-convex function.*

Proof: The first assertion can be seen from the finitely constrained representation

$$\Gamma = \{\, \boldsymbol{\gamma} \in \mathbb{R}_{p+} \mid \boldsymbol{\gamma} C_s \boldsymbol{x} \leqq 1 \text{ for } \boldsymbol{x} \in X_V \,\}.$$

To show the second assertion let $\boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2$ be two point of the level set $\{\, \boldsymbol{\gamma} \mid \Xi(\boldsymbol{\gamma}) \leqq \alpha \,\}$, meaning $\sup\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X,\ \boldsymbol{\gamma}^k C_s \boldsymbol{x} \geqq 1 \,\} \leqq \alpha$ for $k = 1, 2$, and suppose $\sup\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X,\ (\beta\boldsymbol{\gamma}^1 + (1-\beta)\boldsymbol{\gamma}^2) C_s \boldsymbol{x} \geqq 1 \,\} > \alpha$ for some $\beta \in (0,1)$. Then there is $\tilde{\boldsymbol{x}} \in X$ such that $(\beta\boldsymbol{\gamma}^1 + (1-\beta)\boldsymbol{\gamma}^2) C_s \tilde{\boldsymbol{x}} \geqq 1$ and $\phi(\tilde{\boldsymbol{x}}) > \alpha$. For $\tilde{\boldsymbol{x}}$ either $\boldsymbol{\gamma}^1 C_s \tilde{\boldsymbol{x}} \geqq 1$ or $\boldsymbol{\gamma}^2 C_s \tilde{\boldsymbol{x}} \geqq 1$ holds. Hence we obtain either $\sup\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X,\ \boldsymbol{\gamma}^1 C_s \boldsymbol{x} \geqq 1 \,\} \geqq \phi(\tilde{\boldsymbol{x}}) > \alpha$ or $\sup\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X,\ \boldsymbol{\gamma}^2 C_s \boldsymbol{x} \geqq 1 \,\} \geqq \phi(\tilde{\boldsymbol{x}}) > \alpha$, which is a contradiction. □

3.9.2 Algorithm and its convergence

They exploited the outer approximation method to solve the dual problem in Theorem 3.40 and proposed the following algorithm.

/** outer approximation method for solving the dual problem **/

⟨0⟩ (Initialization) Construct a polyhedral set Γ^0 such that $\Gamma \subseteqq \Gamma^0$ and the vertex set of Γ^0 is easily enumerated. Set $k := 0$ and go to ⟨k⟩.

⟨k⟩ (Iteration k)

⟨k1⟩ Solve the relaxation problem

$$\left| \begin{array}{ll} \max\limits_{\boldsymbol{\gamma}} & \Xi(\boldsymbol{\gamma}) \\ \text{s.t.} & \boldsymbol{\gamma} \in \Gamma^k, \end{array} \right.$$

to obtain a solution $\boldsymbol{\gamma}^k$.

⟨k2⟩ Solve the linear program

$$\left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \boldsymbol{\gamma}^k C_s \boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in X, \end{array} \right.$$

to obtain a vertex solution $\boldsymbol{x}^k$ and the optimal value $\sigma_k = \boldsymbol{\gamma}^k C_s \boldsymbol{x}^k$.

⟨k3⟩ If $\sigma_k \leqq 1$, meaning that $\boldsymbol{\gamma}^k$ is in Γ and hence solves $\max\{\, \Xi(\boldsymbol{\gamma}) \mid \boldsymbol{\gamma} \in \Gamma \,\}$, then solve $\max\{\, \phi(\boldsymbol{x}) \mid \boldsymbol{x} \in X,\ \boldsymbol{\gamma}^k C_s \boldsymbol{x} \geqq 1 \,\}$ and obtain a solution $\boldsymbol{x}^*$. Stop with $\boldsymbol{x}^*$ as an optimal solution of (P_E).

$\langle k4 \rangle$ If $\sigma_k > 1$, meaning $\boldsymbol{\gamma}^k \notin \Gamma$, reduce Γ^k to $\Gamma^{k+1} = \Gamma^k \cap \{\, \boldsymbol{\gamma} \mid \boldsymbol{\gamma} C_s \boldsymbol{x}^k \leqq 1 \,\}$. Set $k := k + 1$ and go to $\langle k \rangle$.

See Figure 3.10.

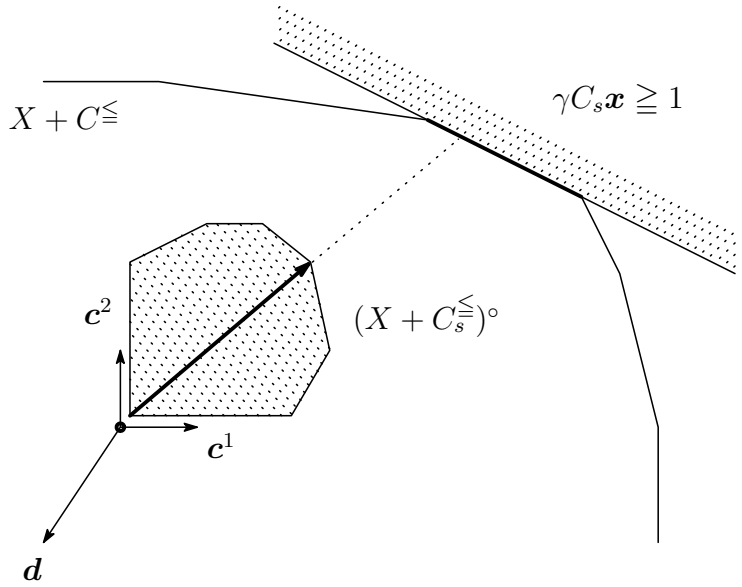


Figure 3.10: Set $X + C_s^{\leqq}$ and its polar

Theorem 3.42 *The algorithm terminates after a finite number of iterations and provides an optimal solution of* (P_E).

Proof: The theorem is readily seen from the fact that Γ is a polyhedral set defined by a finite number of constraints each of which corresponds to a vertex of X and that $\{\boldsymbol{x}^k\}_{k=0,1,\ldots}$ generated by the algorithm is a sequence of distinct vertices of X. □

The most costly step of the algorithm is Step $\langle k1 \rangle$ of maximizing $\Xi(\gamma)$ over Γ^k. Thach-Konno-Yokota [56] proposed to enumerate the vertex set of Γ^{k+1} from that of Γ^k in this step. Numerical results are reported in [56] with two different objective functions: absolute deviation $\phi(\boldsymbol{x}) = -\sum_{i=1}^n w_i |x_i - \bar{x}_i|$ and linear function $\phi(\boldsymbol{x}) = -\sum_{i=1}^n w_i x_i$. They used the enumeration method by Thieu-Tam-Ban [57] in Step $\langle k1 \rangle$. They fixed $m = 20$ and varied $p = 2$ to $5, n = 20$ to 100, and concluded that the number of vertices generated through the computation does not grow very

rapidly as long as p is kept small, and also most of the computation time was spent in solving linear programs.

Based on the same duality concept Yamada-Tanino-Inuiguchi [65] proposed an algorithm to the problem (P_W) with the concave objective function ϕ and the closed convex feasible region X satisfying Slater's constraint qualification.

3.10 D.C. Algorithm for (P_E)

D.C. algorithm (DCA for short) is a primal-dual approach for finding a locally optimal solution of D.C. problem. Recently a general scheme of DCA, first presented by Tao-Souad [52], has been considerably improved by An [2] for solving a broad class of D.C. problem. In this section we briefly review DCA for finding a locally optimal solution of (P_E). For the details the reader should refer to An-Tao-Muu [3]. We describe the simplified form of DCA in the next subsection, and then explain the application to (P_E).

3.10.1 Simplified form of D.C. algorithm

A convex function $f : S \to [-\infty, +\infty]$ on a set $S \subseteqq \mathbb{R}^n$ is said to be *proper* if $\operatorname{dom} f \neq \emptyset$ and $f(\boldsymbol{x}) > -\infty$ for all $\boldsymbol{x} \in S$. Also f is said to be lower semi-continuous (l.s.c for short) if $\liminf_{\boldsymbol{y}\to\boldsymbol{x}} f(\boldsymbol{y}) \geqq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in S$, which equivalent that $\operatorname{epi} f$ is closed. For a proper function $f : \mathbb{R}^n \to [-\infty, +\infty]$, the *conjugate function of* f, denoted by f^*, is given by

$$f^*(\boldsymbol{y}) = \sup\{\, \langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathbb{R}^n \,\}. \tag{3.24}$$

Since f^* is a point-wise supremum of infinitely many linear functions, f^* is clearly l.s.c and convex. It is also known that $f^{**} \leqq f$, and

$$f^{**} \geqq f \text{ if and only if } f \text{ is l.s.c and convex.}$$

See e.g., Tuy [61] for fundamental results on conjugate function. A proper l.s.c convex function f is said to be *locally polyhedral convex* if it satisfies that for every $(\boldsymbol{x}, \mu) \in \operatorname{epi} f$ there is a polyhedral convex neighbourhood of $(\boldsymbol{x}, \mu)$ relative to $\operatorname{epi} f$.

The ϵ-*subdifferential of* f *at* $\bar{\boldsymbol{x}}$, denoted by $\partial_\epsilon f(\bar{\boldsymbol{x}})$, is defined as

$$\partial_\epsilon f(\bar{\boldsymbol{x}}) = \{\, \boldsymbol{t} \in \mathbb{R}_n \mid f(\boldsymbol{x}) \geqq \langle \boldsymbol{t}, \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle + f(\bar{\boldsymbol{x}}) - \epsilon \text{ for all } \boldsymbol{x} \in X \,\}. \tag{3.25}$$

When $\epsilon = 0$, we simply write $\partial f(\bar{\boldsymbol{x}})$, which is the same definition of the subdifferential of (2.14).

DCA provides a locally optimal solution of the following D.C. problem

$$(P_{dc}) \qquad \left| \begin{array}{ll} \inf & p(\boldsymbol{x}) - q(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in \mathbb{R}^n, \end{array} \right.$$

where p and q are l.s.c proper convex functions on $\mathbb{R}^n$. The dual problem of (P_{dc}), first studied by Toland [60], is formulated as

$$(D_{dc}) \qquad \left| \begin{array}{ll} \inf & q^*(\boldsymbol{y}) - p^*(\boldsymbol{y}) \\ \text{s.t.} & \boldsymbol{y} \in \mathbb{R}^n. \end{array} \right.$$

A point $\boldsymbol{x}^*$ is said to be the *critical point* of $p - q$ if it satisfies $\partial p(\boldsymbol{x}^*) \cap \partial q(\boldsymbol{x}^*) \neq \emptyset$.

The following theorem, whose proof can be seen in An [2], implies the global optimality condition for (P_{dc}).

Theorem 3.43 *The point* $\boldsymbol{x}^*$ *is an optimal solution of* (P_{dc}) *if and only if* $\partial_\epsilon p(\boldsymbol{x}^*) \subseteqq \partial_\epsilon q(\boldsymbol{x}^*)$ *for all* $\epsilon > 0$.

Since this condition is difficult to use for deriving solution methods, DCA scheme is based on the local optimality condition implied by the following theorem.

Theorem 3.44 *If* q *is locally polyhedral convex and* $\boldsymbol{x}^* \in \mathrm{dom}\, q$ *then* $\partial q(\boldsymbol{x}^*) \subseteqq \partial p(\boldsymbol{x}^*)$ *is a necessary and sufficient condition for* $\boldsymbol{x}^*$ *to be a locally optimal solution.*

Proof: See An-Tao-Muu [3]. □

DCA generates infinitely sequences $\{\boldsymbol{x}^k\}_{k=0,1,\ldots}$ and $\{\boldsymbol{y}^k\}_{k=0,1,\ldots}$ satisfying the following conditions:

(i) $\boldsymbol{y}^{k+1} \in \partial q(\boldsymbol{x}^k)$ and $\boldsymbol{x}^{k+1} \in \partial p^*(\boldsymbol{y}^k)$ for each $k = 0, 2, \ldots$.

(ii) $(p - q)(\boldsymbol{x}^k)$ and $(q^* - p^*)(\boldsymbol{y}^k)$ are decreasing.

(iii) Every limit point $\boldsymbol{x}^*$ of $\{\boldsymbol{x}^k\}_{k=0,1,\ldots}$ and $\boldsymbol{y}^*$ of $\{\boldsymbol{y}^k\}_{k=0,1,\ldots}$ is a critical point of $p-q$ and q^*-p^*, respectively.

The above points $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ are usually locally optimal solutions of (P_{dc}) and (D_{dc}), respectively, and satisfy $(p-q)(\boldsymbol{x}^*) = (q^*-p^*)(\boldsymbol{y}^*)$. If we have

$$\begin{aligned} \partial q(\boldsymbol{x}^k) &= \operatorname{argsup}\{\, \langle \boldsymbol{x}^k, \boldsymbol{y}\rangle - q^*(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{R}^n \,\} \\ &= \operatorname{arginf}\{\, q^*(\boldsymbol{y}) - (p^*(\boldsymbol{y}^{k-1}) + \langle \boldsymbol{x}^k, \boldsymbol{y}-\boldsymbol{y}^{k-1}\rangle) \mid \boldsymbol{y} \in \mathbb{R}^n \,\}, \end{aligned} \tag{3.26}$$

and

$$\begin{aligned} \partial p^*(\boldsymbol{y}^k) &= \operatorname{argsup}\{\, \langle \boldsymbol{x}, \boldsymbol{y}^k\rangle - p(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathbb{R}^n \,\} \\ &= \operatorname{arginf}\{\, p(\boldsymbol{x}) - (q(\boldsymbol{x}^{k-1}) + \langle \boldsymbol{x}-\boldsymbol{x}^{k-1}, \boldsymbol{y}^k\rangle) \mid \boldsymbol{x} \in \mathbb{R}^n \,\}, \end{aligned} \tag{3.27}$$

then we can obtain $\boldsymbol{y}^{k+1} \in \partial q(\boldsymbol{x}^k)$ and $\boldsymbol{x}^{k+1} \in p^*(\boldsymbol{y}^k)$ by solving the above convex problems. Here we describe the simplified form of DCA.

/*** Simplified form of DCA ***/

$\langle 0\rangle$ (initialization) Choose $\boldsymbol{x}^0 \in \operatorname{dom} q$ and calculate $\boldsymbol{y}^0 \in \partial q(\boldsymbol{x}^0)$. Set k := 0.

$\langle k\rangle$ (iteration k)

$\langle k1\rangle$ If $\partial q(\boldsymbol{x}^k) \cap \partial p(\boldsymbol{x}^k) \neq \emptyset$ then stop.

$\langle k2\rangle$ Obtain
$\boldsymbol{y}^{k+1} \in \partial q(\boldsymbol{x}^k) = \operatorname{argsup}\{\, \langle \boldsymbol{x}^k, \boldsymbol{y}\rangle - q^*(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{R}^n \,\}$ and
$\boldsymbol{x}^{k+1} \in \partial p^*(\boldsymbol{y}^k) = \operatorname{argsup}\{\, \langle \boldsymbol{x}, \boldsymbol{y}^k\rangle - p(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathbb{R}^n \,\}$.

$\langle k3\rangle$ Set $k := k+1$ and go to $\langle k\rangle$.

Assumptions for the above conditions (3.26) and (3.27), and for the convergence of DCA are studied in e.g., An-Tao-Muu [3].

3.10.2 Application of DCA to (P_E)

As stated in Section 3.8, (P_E) is equivalent to

$$(Q(\pi)) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) - \pi g(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in X, \end{array} \right.$$

for a sufficiently large $\pi > 0$. By the duality theorem of linear programming the function

$$g(\boldsymbol{x}) = \max\{\, \boldsymbol{e}C\boldsymbol{x}' \mid C\boldsymbol{x}' \geqq C\boldsymbol{x},\ A\boldsymbol{x}' = \boldsymbol{b},\ \boldsymbol{x}' \geqq \boldsymbol{0}\,\} - \boldsymbol{e}C\boldsymbol{x},$$

is equal to

$$g(\boldsymbol{x}) = \min\{\, -\boldsymbol{\lambda}C\boldsymbol{x} + \boldsymbol{\mu}\boldsymbol{b} \mid -\boldsymbol{\lambda}C + \boldsymbol{\mu}A \geqq \boldsymbol{e}C,\ \boldsymbol{\lambda} \geqq \boldsymbol{0}\,\} - \boldsymbol{e}C\boldsymbol{x}.$$

Letting $(\boldsymbol{\lambda}(\boldsymbol{x}), \boldsymbol{\mu}(\boldsymbol{x}))$ be an optimal solution of

$$\min\{\, -\boldsymbol{\lambda}C\boldsymbol{x} + \boldsymbol{\mu}\boldsymbol{b} \mid -\boldsymbol{\lambda}C + \boldsymbol{\mu}A \geqq \boldsymbol{e}C,\ \boldsymbol{\lambda} \geqq \boldsymbol{0}\,\}, \tag{3.28}$$

we have $-\boldsymbol{e}C - \boldsymbol{\lambda}(\boldsymbol{x})C \in \partial\, g(\boldsymbol{x})$.

For a set $S \subseteqq \mathbb{R}^n$, the *indicator function of* S, denoted by $\delta_S : \mathbb{R}^n \to \{0, +\infty\}$, is given by

$$\delta_S(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in S \\ +\infty & \text{otherwise.} \end{cases} \tag{3.29}$$

Using the indicator function of X, we can rewrite $(Q(\pi))$ as

$$(Q(\pi)) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) - \pi g(\boldsymbol{x}) - \delta_X(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in \mathbb{R}^n. \end{array} \right.$$

For the following three cases, we explain how to apply DCA.

Case where ϕ is concave

Letting $p(\boldsymbol{x}) = -\phi(\boldsymbol{x}) + \delta_X(\boldsymbol{x})$ and $q_\pi(\boldsymbol{x}) = -\pi g(\boldsymbol{x})$ implies that p and q_π are l.s.c convex functions, and hence we can convert $(Q(\pi))$ to the problem

$$\left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & p(\boldsymbol{x}) - q_\pi(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in \mathbb{R}^n, \end{array} \right.$$

which is of the form of (P_{dc}). DCA in this case are described as follows.

/*** DCA in the case where ϕ is concave ***/

$\langle 0 \rangle$ (initialization) Choose $\boldsymbol{x}^0 \in X \subseteqq \operatorname{dom} q_\pi$, and set $\boldsymbol{y}^0 := \pi(\boldsymbol{e}C + \boldsymbol{\lambda}(\boldsymbol{x}^0)C) \in \partial\, q_\pi(\boldsymbol{x}^0)$ and k := 0.

$\langle k \rangle$ (iteration k)

$\langle k1 \rangle$ If $\partial\, q_\pi(\boldsymbol{x}^k) \cap \partial\, p(\boldsymbol{x}^k) \neq \emptyset$ then stop.

$\langle k2 \rangle$ Obtain
$\boldsymbol{y}^{k+1} := \pi(\boldsymbol{e}C + \boldsymbol{\lambda}(\boldsymbol{x}^k)C) \in \partial\, q_\pi(\boldsymbol{x}^k)$ and
$\boldsymbol{x}^{k+1} \in \operatorname{argmax}\{\, \phi(\boldsymbol{x}) + \langle \boldsymbol{y}^k, \boldsymbol{x} \rangle \mid \boldsymbol{x} \in X \,\}$.

$\langle k3 \rangle$ Set $k := k + 1$ and go to $\langle k \rangle$.

Case where ϕ is convex

Letting $p(\boldsymbol{x}) = \delta_X(\boldsymbol{x})$ and $q_\pi(\boldsymbol{x}) = \phi(\boldsymbol{x}) - \pi g(\boldsymbol{x})$ implies that p and q_π are l.s.c convex functions. Suppose that we can obtain $\boldsymbol{\nu}(\boldsymbol{x}) \in \partial\, \phi(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathbb{R}^n$. DCA in this case are described as follows.

/*** DCA in the case where ϕ is convex ***/

$\langle 0 \rangle$ (initialization) Choose $\boldsymbol{x}^0 \in X \subseteqq \operatorname{dom} q_\pi$, and set $\boldsymbol{y}^0 := \pi(\boldsymbol{e}C + \boldsymbol{\lambda}(\boldsymbol{x}^0)C + \boldsymbol{\nu}(\boldsymbol{x}^0)) \in \partial\, q_\pi(\boldsymbol{x}^0)$ and k := 0.

$\langle k \rangle$ (iteration k)

$\langle k1 \rangle$ If $\partial\, q_\pi(\boldsymbol{x}^k) \cap \partial\, p(\boldsymbol{x}^k) \neq \emptyset$ then stop.

$\langle k2 \rangle$ Obtain
$\boldsymbol{y}^{k+1} := \pi(\boldsymbol{e}C + \boldsymbol{\lambda}(\boldsymbol{x}^k)C + \boldsymbol{\nu}(\boldsymbol{x}^k)) \in \partial\, q_\pi(\boldsymbol{x}^k)$ and
$\boldsymbol{x}^{k+1} \in \operatorname{argmax}\{\, \langle \boldsymbol{y}^k, \boldsymbol{x} \rangle \mid \boldsymbol{x} \in X \,\}$.

$\langle k3 \rangle$ Set $k := k + 1$ and go to $\langle k \rangle$.

Case where ϕ is quadratic

In the case where $\phi(\boldsymbol{x}) = \frac{1}{2}\langle \boldsymbol{x}M, \boldsymbol{x} \rangle + \langle \boldsymbol{r}, \boldsymbol{x} \rangle$ with a symmetric matrix $M \in \mathbb{R}^{n \times n}$ and a row vector $\boldsymbol{r} \in \mathbb{R}^n$, we calculate $\rho > 0$ such that $(\rho I - M)$ is positive definite and define $p(\boldsymbol{x}) = \frac{1}{2}\langle \boldsymbol{x}(\rho I - M), \boldsymbol{x} \rangle + \delta_X(\boldsymbol{x})$ and $q_\pi(\boldsymbol{x}) = \frac{1}{2}\rho \|\boldsymbol{x}\|^2 + \langle \boldsymbol{r}, \boldsymbol{x} \rangle - \pi g(\boldsymbol{x})$. DCA in this case are described as follows.

/*** DCA in the case where ϕ is quadratic ***/

⟨0⟩ (initialization) Choose $\boldsymbol{x}^0 \in X \subseteqq \operatorname{dom} q_\pi$, and set $\boldsymbol{y}^0 := (\rho\boldsymbol{x}^0 + \boldsymbol{r})^\top + \pi(\boldsymbol{e}C + \boldsymbol{\lambda}(\boldsymbol{x}^0)C) \in \partial\, q_\pi(\boldsymbol{x}^0)$ and k := 0.

⟨k⟩ (iteration k)

⟨k1⟩ If $\partial\, q_\pi(\boldsymbol{x}^k) \cap \partial\, p(\boldsymbol{x}^k) \neq \emptyset$ then stop.

⟨k2⟩ Obtain
$\boldsymbol{y}^{k+1} := (\rho\boldsymbol{x}^k + \boldsymbol{r})^\top + \pi(\boldsymbol{e}C + \boldsymbol{\lambda}(\boldsymbol{x}^k)C) \in \partial\, q_\pi(\boldsymbol{x}^k)$ and
$\boldsymbol{x}^{k+1} \in \operatorname{argmin}\{\, \frac{1}{2}\langle \boldsymbol{x}(\rho I - M), \boldsymbol{x}\rangle - \langle \boldsymbol{y}^k, \boldsymbol{x}\rangle \mid \boldsymbol{x} \in X \,\}$.

⟨k3⟩ Set $k := k + 1$ and go to ⟨k⟩.

3.11 Other Methods

Benson-Sayin [12] consider four special cases of linear (P_E), and propose simple linear programming procedures. Benson-Lee [11] consider (MC) with two criteria and propose an algorithm for maximizing an upper semicontinuous function ϕ. In this case the outcome set Y is of dimension at most two, and Y_E is of dimension at most one, i.e., Y_E consists of edges and vertices.

Thoai [58] considers the case where $\phi(\boldsymbol{x}) = \varphi(C\boldsymbol{x})$ and propose an outer approximation algorithm. He assumes that φ is a quasi-convex function and *nondecreasing* in the sense that $\boldsymbol{y}' \geqq \boldsymbol{y}$ implies $\varphi(\boldsymbol{y}') \geqq \varphi(\boldsymbol{y})$. It is seen that

$$\max\{\, \varphi(C\boldsymbol{x}) \mid \boldsymbol{x} \in X_E \,\} = \max\{\, \varphi(C\boldsymbol{x}) \mid \boldsymbol{x} \in X \,\}.$$

His algorithm makes a sequence of polyhedral sets Y^k shrinking to the lower outcome set $Y^{\leqq}$, solves the relaxation problem $\max\{\, \varphi(\boldsymbol{y}) \mid \boldsymbol{y} \in Y_E^k \,\}$ to find a solution $\boldsymbol{y}^k$, where Y_E^k is the set of efficient points of Y^k. If $\boldsymbol{y}^k \in Y^{\leqq}$, any point $\boldsymbol{x} \in X$ with $C\boldsymbol{x} = \boldsymbol{y}^k$ is an optimal solution of (P_E). Otherwise, it generates a cutting plane defined by the linear equation $\ell^k(\boldsymbol{y}) = 0$ to cut $\boldsymbol{y}^k$ off the set Y^k and reduces Y^k to $Y^{k+1} \cap \{\, \boldsymbol{y} \mid \boldsymbol{y} \in \mathbb{R}_p,\ \ell^k(\boldsymbol{y}) \leqq 0 \,\}$. Since φ is quasi-convex, a vertex of Y^k attains $\max\{\, \varphi(\boldsymbol{y}) \mid \boldsymbol{y} \in Y_E^k \,\}$. Thus for solving the relaxation problem he proposes to compute all the vertices of Y^{k+1} from the vertex set of Y^k. The key of the algorithm is the step of checking whether $\boldsymbol{y}^k$ lies in $Y^{\leqq}$ and generating the cutting

plane. Note that $X = \{\, \boldsymbol{x} \in \mathbb{R}^n_+ \mid A\boldsymbol{x} = \boldsymbol{b} \,\}$, then $\boldsymbol{y}^k \in Y^{\leqq}$ if and only if the system

$$\boldsymbol{y}^k \leqq C\boldsymbol{x}, \quad A\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{x} \geqq \boldsymbol{0}$$

has a solution $\boldsymbol{x}$. By the linear programming duality theorem this is equivalent to

$$\max\{\, -\boldsymbol{\lambda}\boldsymbol{y}^k + \boldsymbol{\mu}\boldsymbol{b} \mid -\boldsymbol{\lambda}C + \boldsymbol{\mu}A \leqq 0,\ \boldsymbol{\lambda} \geqq \boldsymbol{0} \,\} = 0.$$

When this problem has a positive optimal value, $\boldsymbol{y}^k \notin Y^{\leqq}$ and further $\ell^k(\boldsymbol{y}) = -\boldsymbol{\lambda}^k\boldsymbol{y} + \boldsymbol{\mu}^k\boldsymbol{b} = 0$ is the desired cutting plane, where $(\boldsymbol{\lambda}^k, \mu^k)$ is an optimal solution of this problem. In Theorem 4.1 of Thoai [58] the procedure is shown to be finite. Thoai also considers the nonlinear case, namely $\phi(\boldsymbol{x}) = \varphi(c^1(\boldsymbol{x}), \cdots, c^p(\boldsymbol{x}))$, $c^i(\boldsymbol{x})$'s are concave functions, and also X is a closed convex set defined by nonlinear inequalities. A preliminary experiment for the quadratically constrained problems with quadratic c^i's shows that the most expensive step of the algorithm is the enumeration of vertices, whose number grows rapidly as the number p of criteria increases.

One of the often occurred objective functions ϕ is $\phi(\boldsymbol{x}) = -\boldsymbol{c}^i\boldsymbol{x}$, i.e., (P_E) is to minimize the ith objective function $\boldsymbol{c}^i\boldsymbol{x}$ of $C\boldsymbol{x}$. To estimate the optimal value of this problem, the process of using the *payoff table* was proposed by several authors. See e.g., Section 9.13 of Steuer [49]. Consider the linear program

$$\max\{\, \boldsymbol{c}^j\boldsymbol{x} \mid \boldsymbol{x} \in X \,\}$$

and let $\boldsymbol{x}^j$ be its optimal solution for $j = 1, \ldots, p$. Then the payoff table is the matrix whose (i, j)-element is $\boldsymbol{c}^i\boldsymbol{x}^j$. The popular way of estimating $\min\{\, \boldsymbol{c}^i\boldsymbol{x} \mid \boldsymbol{x} \in X_E \,\}$ is to scan the table and determine the minimum of each column. Notice that this column-wise minimum value gives neither an upper bound nor an lower bound of $\min\{\, \boldsymbol{c}^i\boldsymbol{x} \mid \boldsymbol{x} \in X_E \,\}$ because $\boldsymbol{x}^j$ might not be efficient. In order to ensure

Table 3.1: Payoff table

	1	2	$\cdots$	p
1	$\boldsymbol{c}^1\boldsymbol{x}^1$	$\boldsymbol{c}^1\boldsymbol{x}^2$	$\cdots$	$\boldsymbol{c}^1\boldsymbol{x}^p$
		$\cdots$		
p	$\boldsymbol{c}^p\boldsymbol{x}^1$	$\boldsymbol{c}^p\boldsymbol{x}^2$	$\cdots$	$\boldsymbol{c}^p\boldsymbol{x}^p$

that $\boldsymbol{x}^j$ is efficient, lexicographical maximization could be employed, i.e., to find $\boldsymbol{x}^1$ first maximize $\boldsymbol{c}^1\boldsymbol{x}$ over X and obtain the optimal value z^1, maximize $\boldsymbol{c}^2\boldsymbol{x}$ over $X \cap \{\, \boldsymbol{x} \mid \boldsymbol{c}^1\boldsymbol{x} = z^1 \,\}$, and maximize $\boldsymbol{c}^3\boldsymbol{x}$ over $X \cap \{\, \boldsymbol{x} \mid \boldsymbol{c}^1\boldsymbol{x} = z^1,\ \boldsymbol{c}^2\boldsymbol{x} = z^2 \,\}$ and so on. Then each column-wise minimum of the payoff table thus obtained gives an upper bound of $\min\{\, \boldsymbol{c}^i\boldsymbol{x} \mid \boldsymbol{x} \in X_E \,\}$. In Isermann-Steuer [32], and Reeves-Reid [42] is reported how a good approximation is obtained from the payoff table based on the computational experience of randomly generated problems.

3.12 Nonlinear Optimization Problem over the Efficient Set

In this section, we explore the possibility of extending functions in the optimization problem over the efficient set to nonlinear functions.

Given a closed set X of $\mathbb{R}^n$ and a criterion function $\boldsymbol{f} : X \to \mathbb{R}^p$ with $p \geqq 2$, the *nonlinear multicriteria problem*, denoted by (MC^N), is

$$(MC^N) \qquad \left| \begin{array}{ll} \text{Vector Max} & \boldsymbol{f}(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{x} \in X. \end{array} \right.$$

Let X_E^N denote the set of efficient points for (MC^N), i.e.,

$$X_E^N = \{\, \boldsymbol{x} \in X \mid \nexists \boldsymbol{y} \in X : \boldsymbol{f}(\boldsymbol{y}) \geqq \boldsymbol{f}(\boldsymbol{x}),\ \boldsymbol{f}(\boldsymbol{y}) \neq \boldsymbol{f}(\boldsymbol{x}) \,\}. \tag{3.30}$$

Definition 3.45 (nonlinear optimization problem over the efficient set) *The nonlinear optimization problem over the efficient set, denoted by (P_E^N), is*

$$(P_E^N) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & \phi(\boldsymbol{x}) \\ s.t. & \boldsymbol{x} \in X_E^N, \end{array} \right.$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ is a continuous function to be maximized.

The assumptions on X and $\boldsymbol{f}$ diversify fundamental properties of X_E^N. We observe the variation of X_E^N by showing some examples.

Example 3.46 Let X be a nonempty polytope and

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f_3(\boldsymbol{x}) \end{bmatrix},$$

where f_1, f_2 and f_3 are all concave functions. We show an example in Figure 3.11, in which the contour lines of these functions are depicted by dashed curved lines. As seen in the figure, there may be no vertex of X that attains the optimal value of (P_E^N).

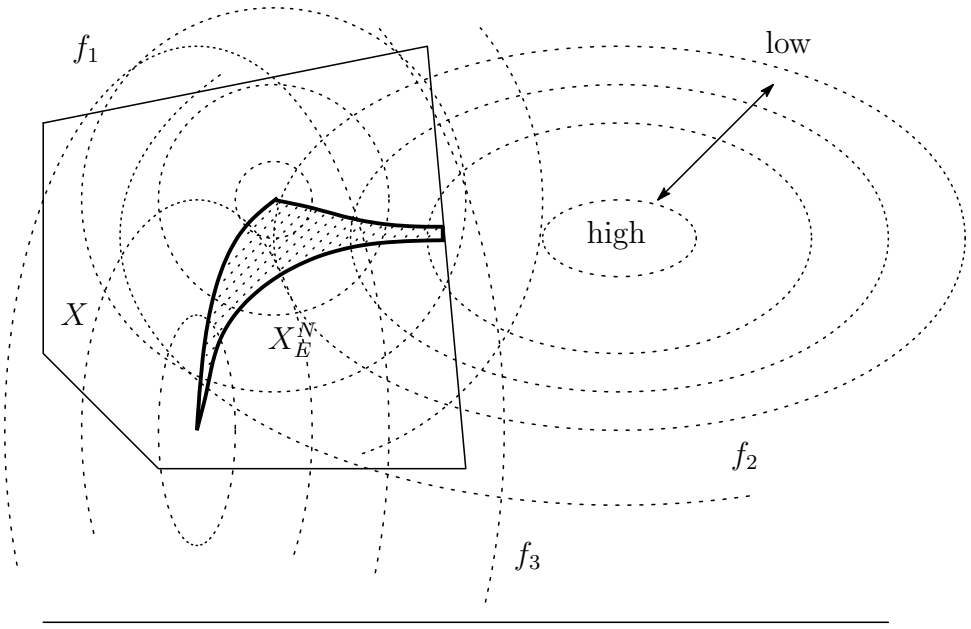


Figure 3.11: Set X_E^N when f_1, f_2 and f_3 are all concave

Example 3.47 We redefine f_2 of the above example as a quasi-concave function. As seen in Figure 3.12, X_E^N is no longer connected in this case.

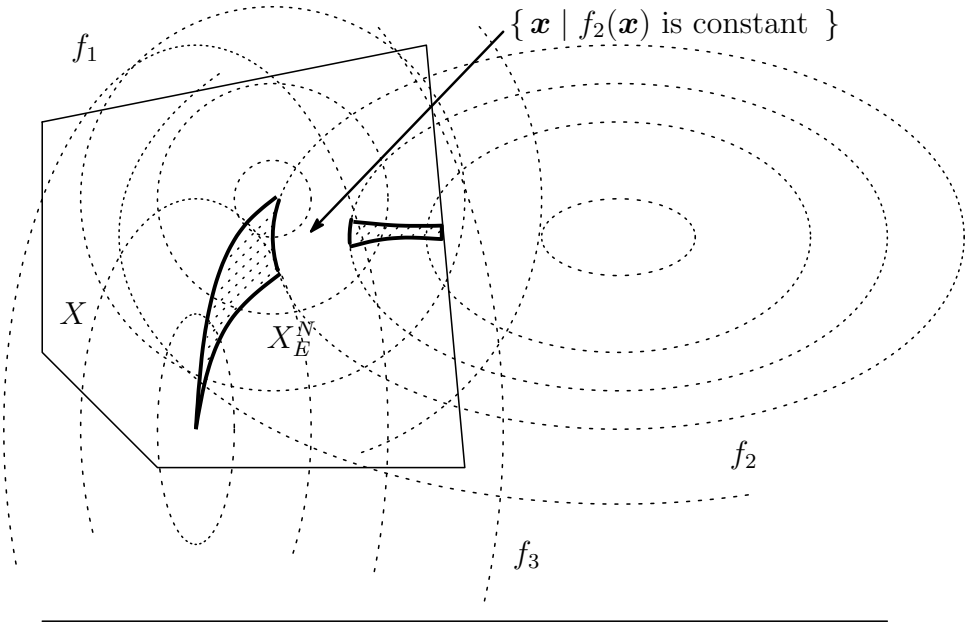


Figure 3.12: Set X_E^N when f_2 is quasi-concave

Example 3.48 We show an example with convex functions f_1, f_2 and f_3 in Figure 3.13. As seen in the figure, X_E^N is not connected.

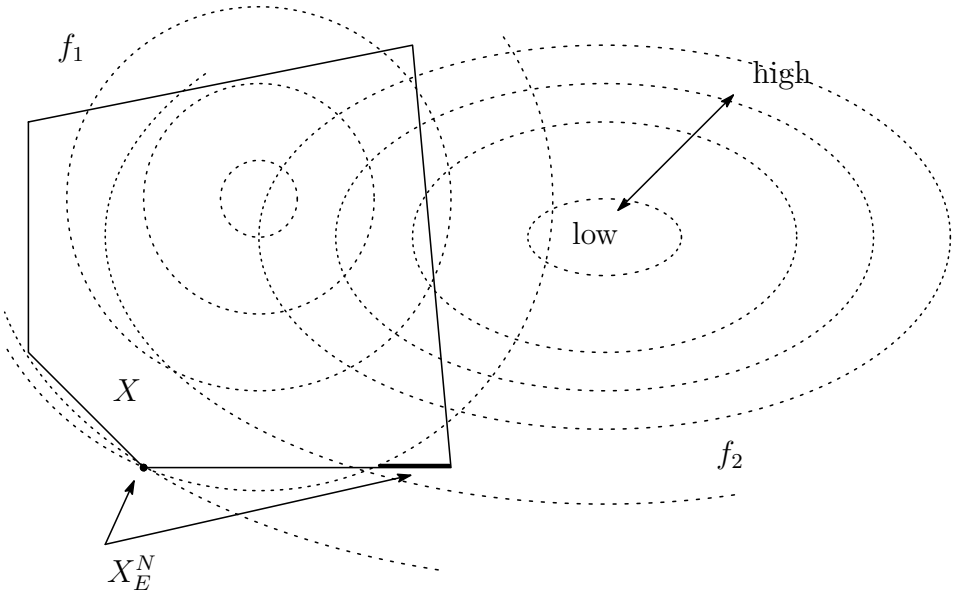


Figure 3.13: Set of X_E^N when f_1, f_2 and f_3 are all convex

Example 3.49 We next observe an example of X_E^N when X is not convex. Let $X = \{\, \boldsymbol{x} \in \mathbb{R}^2 \mid \mathbf{0} \leqq \boldsymbol{x} \leqq \mathbf{1},\ x_1 + x_2 \leqq 1 \text{ or } x_2 = x_1 \,\}$ and

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{c}^1 \\ \boldsymbol{c}^2 \\ \boldsymbol{c}^3 \end{bmatrix} \boldsymbol{x},$$

where $\boldsymbol{c}^1 = \begin{bmatrix} 1 & -1 \end{bmatrix}$, $\boldsymbol{c}^2 = \begin{bmatrix} -1 & 1 \end{bmatrix}$ and $\boldsymbol{c}^3 = \begin{bmatrix} 1 & 1 \end{bmatrix}$. As seen in Figure 3.14, X_E^N is not closed set at $(x_1, x_2) = (1/2, 1/2)$.

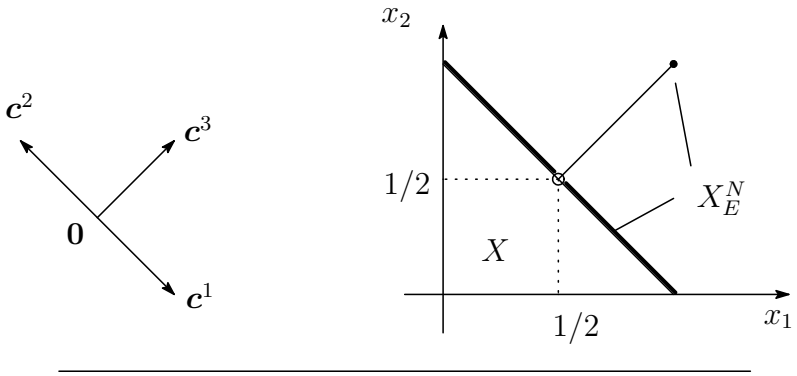


Figure 3.14: Set X_E^N when X is not a convex set

Example 3.50 We show an example in the case where X_E^N is not closed set while X is a polytope. Let $X = \{\, \boldsymbol{x} \in \mathbb{R}^2 \mid \mathbf{0} \leqq \boldsymbol{x} \leqq \mathbf{1} \,\}$, and

$$\boldsymbol{f}(\boldsymbol{x}) = \left[\begin{bmatrix} \cos \frac{1}{2}x_2 & \sin \frac{1}{2}x_2 \\ -\sin \frac{1}{2}x_2 & \cos \frac{1}{2}x_2 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \right]^\top \boldsymbol{x}.$$

We see in Figure 3.15 that X_E^N is not closed set since $\{\, (x_1, 1/2) \mid x_1 \in [0,1) \,\}$ is not contained in X_E^N.

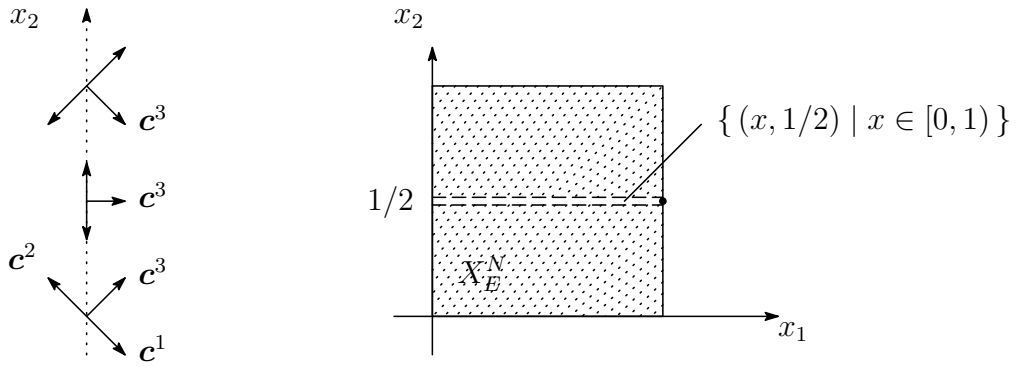


Figure 3.15: Set X_E^N is not closed while X is a polytope

When a criterion function $\boldsymbol{f}$ is nonlinear, the feasible set X_E^N may be unconnected, neither a closed nor an open set even when X is a polytope. Thus it is hopelessly difficult to solve nonlinear optimization problems over the efficient set. To put it the other way around, connectedness and closedness of X_E are very important properties should be exploited. However, solving (P_E) is still hard task even if we make the most of these properties.

Chapter 4

Minimum Maximal Flow Problem and Preceding Algorithms

Most of the algorithms for (P_E) reviewed in Chapter 3 anticipate a small number p of criteria of Problem (MC) and convert Problem (P_E) to a global optimization problem in p or so variables. However, there are interesting and important problems that do not enjoy the low dimensionality of p. An example is a minimum maximal flow problem, abbreviated to (mmF). The number of criteria with respect to (mmF) equals the number of variables, meaning $p = n$. In this sense, we can say that (mmF) is relatively difficult case of (P_E). After Shi-Yamamoto [47] first studied (mmF) and proposed an algorithm, several algorithms for (mmF) combining local search and global optimization technique have been proposed in e.g., Gotoh-Thoai-Yamamoto [24] and Shigeno-Takahashi-Yamamoto [48]. An approach based on D.C. algorithm is found in Muu-Shi [37].

In this chapter we define (mmF) and review the following two algorithms preceding our study: outer approximation algorithm of parameter set, which is presented by Shigeno-Takahashi-Yamamoto [48], and D.C. algorithm by Muu-Shi [37].

4.1 Minimum Maximal Flow Problem

We are given a directed network $\mathcal{N} = (V, E, \boldsymbol{c})$, where V is the set of $m + 2$ nodes containing the single source node s and the single sink node t, E is the set of n arcs, and $\boldsymbol{c}$ is the n-dimensional column vector whose hth element c_h is the capacity

of arc $h \in E$. Without loss of generality, we assume throughout this paper that a given network is *simple*, i.e., there are no parallel arcs. An example is shown in Figure 4.1, in which the number beside each arc is a capacity.

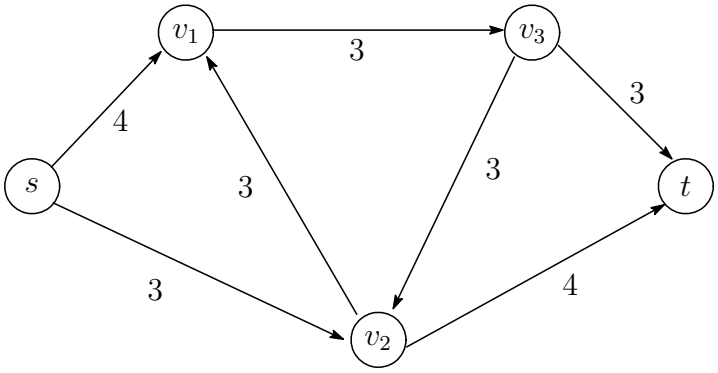


Figure 4.1: A given network $\mathcal{N} = (V, E, \boldsymbol{c})$

Let $\partial^+ : E \to V$ and $\partial^- : E \to V$ be incidence functions. When $h = (u, v)$, i.e., arc h leaves node u and enters node v, we write $\partial^+ h = u$ and $\partial^- h = v$. A vector $\boldsymbol{x} \in \mathbb{R}^n$ is said to be a *feasible flow* if it satisfies the *capacity constraints*:

$$0 \leqq x_h \leqq c_h \text{ for each arc } h \in E,$$

and *conservation equations*:

$$\sum_{\partial^+ h = v} x_h = \sum_{\partial^- h = v} x_h \text{ for each node } v \in V \setminus \{s, t\}.$$

Note that conservation equations can be simply written as

$$A\boldsymbol{x} = \boldsymbol{0},$$

where the $m \times n$ matrix $A = \left[a_{vh}\right]$ is the well-known node-arc incidence matrix, whose $(v, h) \in V \setminus \{s, t\} \times E$ element a_{vh} is

$$a_{vh} = \begin{cases} +1 & \text{if arc } h \text{ leaves node } v, \text{ i.e., } \partial^+ h = v \\ -1 & \text{if arc } h \text{ enters node } v, \text{ i.e., } \partial^- h = v \\ 0 & \text{otherwise.} \end{cases}$$

Further, let X denote the set of feasible flows, i.e.,

$$X = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c} \,\}. \tag{4.1}$$

Throughout the chapter we assume the following.

Assumption 4.1 *Every capacity takes a positive integer value, i.e.,* $c_h \in \mathbb{Z}$ *and* $c_h > 0$ *for each* $h \in E$.

For a feasible flow $\boldsymbol{x} \in X$, the *flow value of* $\boldsymbol{x}$, denoted by $f_v(\boldsymbol{x})$, is given by

$$f_v(\boldsymbol{x}) = \sum_{\partial^+ h = s} x_h - \sum_{\partial^- h = s} x_h.$$

Using the n-dimensional row vector $\boldsymbol{d}$ whose hth element d_h is

$$d_h = \begin{cases} +1 & \text{if arc } h \text{ leaves the source node } s, \text{ i.e., } \partial^+ h = s \\ -1 & \text{if arc } h \text{ enters the source node } s, \text{ i.e., } \partial^- h = s \\ \ 0 & \text{otherwise,} \end{cases}$$

we can simply write $f_v(\boldsymbol{x}) = \boldsymbol{d}\boldsymbol{x}$. Assumption 4.1 implies that if $\boldsymbol{v}$ is a vertex of X then each element of $\boldsymbol{v}$ is integer, so is the flow value $f_v(\boldsymbol{v})$ of $\boldsymbol{v}$.

The well-known conventional *maximum flow problem* is formulated as follows.

Definition 4.2 (Maximum flow problem) *The maximum flow problem is an optimization problem of maximizing the flow value among feasible flows:*

$$\left| \begin{array}{ll} \max\limits_{\boldsymbol{x}} & f_v(\boldsymbol{x}) = \boldsymbol{d}\boldsymbol{x} \\ s.t. & \boldsymbol{x} \in X. \end{array} \right.$$

A feasible flow $\boldsymbol{x} \in X$ is said to be a *maximal flow* if there is no feasible flow $\boldsymbol{y} \in X$ such that $\boldsymbol{y} \geqq \boldsymbol{x}$ and $\boldsymbol{y} \neq \boldsymbol{x}$. We use X_M to denote the set of maximal flows, i.e.,

$$X_M = \{\, \boldsymbol{x} \in X \mid \nexists \boldsymbol{y} \in X : \boldsymbol{y} \geqq \boldsymbol{x},\ \boldsymbol{y} \neq \boldsymbol{x} \,\}. \tag{4.2}$$

Definition 4.3 (Minimum maximal flow problem) *A minimum maximal flow problem, abbreviated to* (mmF)*, is an optimization problem of minimizing the flow value among maximal flows:*

$$(mmF) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & f_v(\boldsymbol{x}) = \boldsymbol{d}\boldsymbol{x} \\ s.t. & \boldsymbol{x} \in X_M. \end{array} \right.$$

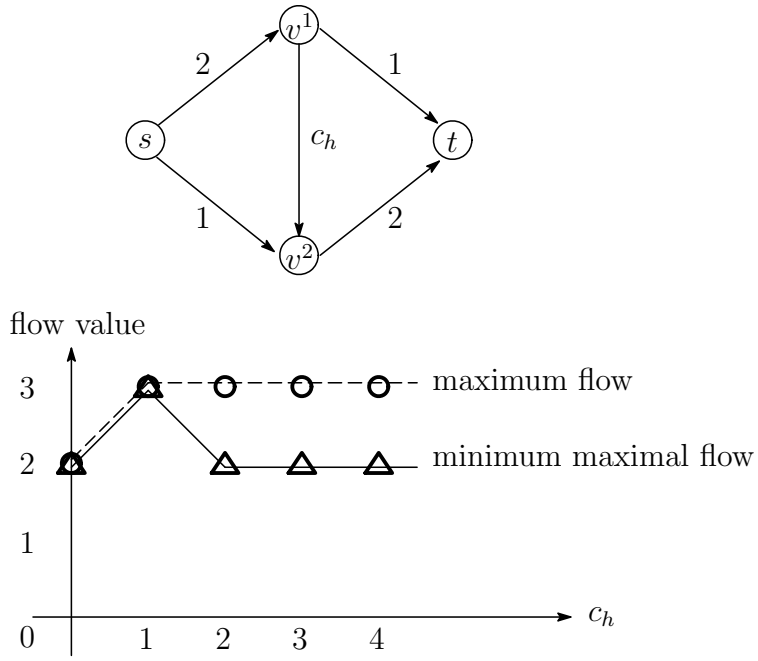


Figure 4.2: Maximum flow vs. minimum maximal flow

Figure 4.2, the network in which is given by Iri [29, 30], highlights the difference between maximum flow and minimum maximal flow. The maximum flow value monotonically grows as the arc capacity c_h increases, while the minimum maximal flow value does not.

The difficulty of the problem is due to the nonconvexity of the set of maximal flows, implying a lot of locally optimal solutions with different objective function values. Indeed, (mmF) embraces the minimum maximal matching problem, which is $\mathcal{NP}$-hard (See e.g., Garey-Johnson [23]).

The concept of uncontrollable flow raised by Iri [29–31] is closely related to but slightly different from the maximal flow (See Appendix A).

4.1.1 (mmF) is a special case of (P_E)

Consider

$$(MC') \qquad \left| \begin{array}{ll} \text{Vector Max} & I\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in X, \end{array} \right.$$

where X is given by (4.1). It is readily seen that X_M of (4.2) is the efficient set for (MC'), and hence (mmF) reduces to (P_E) for $\phi(\boldsymbol{x}) = -f_v(\boldsymbol{x})$. In (mmF), the gap function of (3.4) is

$$g(\boldsymbol{x}) = \max\{\, \boldsymbol{ey} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} \geqq \boldsymbol{x} \,\} - \boldsymbol{ex}, \tag{4.3}$$

and also the gap function with direction $\boldsymbol{\lambda}$ of (3.6) is

$$g_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \max\{\, \boldsymbol{\lambda y} \mid \boldsymbol{y} \in X \,\} - \boldsymbol{\lambda x}. \tag{4.4}$$

Note that (MC') has the objective functions as many as the variables, i.e., $p = n$, the algorithms that exploit the low dimensionality of p would not work efficiently.

4.2 Outer Approximation Algorithm of Parameter Set

In this section we sum up the algorithm for (mmF) proposed by Shigeno-Takahashi-Yamamoto [48] briefly. The method combines the local search procedure described in Section 3.2 with the global technique originated by Phong-Tuyen [41]. As stated in Section 3.2, (mmF) is equivalent to

$$\left|\begin{array}{ll} \min\limits_{(\boldsymbol{x},\boldsymbol{\lambda})} & \boldsymbol{dx} \\ \text{s.t.} & (\boldsymbol{x},\boldsymbol{\lambda}) \in X \times \Lambda \\ & \boldsymbol{\lambda x} \geqq \boldsymbol{\lambda x'} \text{ for all } \boldsymbol{x'} \in X, \end{array}\right.$$

where

$$X = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c} \,\}, \text{ and}$$
$$\Lambda = \{\, \boldsymbol{\lambda} \in \mathbb{R}_n \mid \boldsymbol{\lambda} \geqq \boldsymbol{e},\ \boldsymbol{\lambda 1} = M \,\}.$$

Shigeno-Takahashi-Yamamoto [48] proved that n^2 suffices for the above M. The outline of the algorithm is described as follows.

/** outline of the algorithm **/

⟨0⟩ (initialization) Find a feasible vertex $\boldsymbol{w}^0 \in X_M \cap X_V$. Set $\ell := 0$ and go to ⟨⟨ℓ⟩⟩.

$\langle\langle \ell \rangle\rangle$ (local search) Apply the local search procedure starting from $\boldsymbol{w}^\ell$ to obtain an locally optimal vertex $\bar{\boldsymbol{x}}^{\ell+1} \in X_M \cap X_V$ with the value $\boldsymbol{d}\bar{\boldsymbol{x}}^{\ell+1}$. Set $\alpha := \boldsymbol{d}\bar{\boldsymbol{x}}^{\ell+1} - 1$, $\ell := \ell + 1$ and go to $\langle k \rangle$.

$\langle k \rangle$ (global technique) If $\{ \boldsymbol{x} \in X_M \mid \boldsymbol{dx} \leqq \alpha \} = \emptyset$ then stop ($\bar{\boldsymbol{x}}^\ell$ solves (mmF)). Otherwise find a feasible vertex $\boldsymbol{w}^\ell \in X_M \cap X_V$ from this set, and go to $\langle\langle \ell \rangle\rangle$.

Note that the optimal value of (mmF) as well as the objective function value at vertex of X are integers, and hence $\bar{\boldsymbol{x}}^\ell \in X_M \cap X_V$ solves (mmF) if the set $\{ \boldsymbol{x} \in X_M \mid \boldsymbol{dx} \leqq \boldsymbol{d}\bar{\boldsymbol{x}}^\ell - 1 \}$ is empty. The key issues are

- how to check if $\{ \boldsymbol{x} \in X_M \mid \boldsymbol{dx} \leqq \alpha \} = \emptyset$, and
- how to find a feasible vertex $\boldsymbol{w}^\ell \in X_M \cap X_V$ from $\{ \boldsymbol{x} \in X_M \mid \boldsymbol{dx} \leqq \alpha \}$ when the set is not empty.

For a vector $\boldsymbol{\lambda} \in \mathbb{R}_{n++}$ and $\alpha \in \mathbb{R}$ we define two functions σ and τ_α, which play an crucial role in the global technique, as

$$\sigma(\boldsymbol{\lambda}) = \max\{ \boldsymbol{\lambda x} \mid \boldsymbol{x} \in X \}, \text{ and} \tag{4.5}$$

$$\tau_\alpha(\boldsymbol{\lambda}) = \max\{ \boldsymbol{\lambda x} \mid \boldsymbol{x} \in X, \ \boldsymbol{dx} \leqq \alpha \}. \tag{4.6}$$

The following lemma is readily seen (See also Lemma 3.27 and Lemma 3.28).

Lemma 4.4 (Properties of σ and τ_α)

(i) *σ and τ_α are piece-wise linear positively homogeneous convex functions on $\mathbb{R}_{n++}$.*

(ii) *For all $\boldsymbol{\lambda} \in \mathbb{R}_{n++}$ we have*

$$\sigma(\boldsymbol{\lambda}) = \max\{ \boldsymbol{\lambda v} \mid \boldsymbol{v} \in X_M \cap X_V \}, \text{ and}$$
$$\tau_\alpha(\boldsymbol{\lambda}) = \max\{ \boldsymbol{\lambda v} \mid \boldsymbol{v} \text{ is an efficient vertex of } \{ \boldsymbol{x} \in X \mid \boldsymbol{dx} \leqq \alpha \} \}.$$

(iii) *$\tau_\alpha(\boldsymbol{\lambda}) \leqq \sigma(\boldsymbol{\lambda})$ for any $\boldsymbol{\lambda} \in \mathbb{R}_{n++}$.*

(iv) *$\tau_\alpha(\boldsymbol{\lambda})$ is a nondecreasing function in $\alpha \in \mathbb{R}$.*

We use $\operatorname{epi}\sigma$ to denote the epigraph of σ on Λ, which is defined in the same way of (3.19), i.e.,

$$\operatorname{epi}\sigma = \{\, (\boldsymbol{\lambda}, \mu) \in \Lambda \times \mathbb{R} \mid \mu \geqq \sigma(\boldsymbol{\lambda}) \,\}. \tag{4.7}$$

Lemma 4.4 (ii) implies that

$$\operatorname{epi}\sigma = \{\, (\boldsymbol{\lambda}, \mu) \in \Lambda \times \mathbb{R} \mid \mu - \boldsymbol{\lambda}\boldsymbol{v} \geqq 0 \text{ for all } \boldsymbol{v} \in X_M \cap X_V \,\}. \tag{4.8}$$

Then we have the following theorem whose proof can be found in e.g., Shigeno-Takahashi-Yamamoto [48] (See also Theorem 3.29).

Theorem 4.5 *Then the following statements are equivalent.*

(a) $\{\, \boldsymbol{x} \in X_M \mid \boldsymbol{d}\boldsymbol{x} \leqq \alpha \,\} \neq \emptyset$.

(b) *There is* $\boldsymbol{\lambda} \in \Lambda$ *such that* $\sigma(\boldsymbol{\lambda}) = \tau_\alpha(\boldsymbol{\lambda})$.

(c) *There is a vertex* $(\boldsymbol{\lambda}, \mu)$ *of* $\operatorname{epi}\sigma$ *such that* $\mu = \tau_\alpha(\boldsymbol{\lambda})$.

By Lemma 4.4 and Theorem 4.5, if $\sigma(\boldsymbol{\lambda}) > \tau_\alpha(\boldsymbol{\lambda})$ for all $\boldsymbol{\lambda} \in \Lambda$ then we see that $\{\, \boldsymbol{x} \in X_M \mid \boldsymbol{d}\boldsymbol{x} \leqq \alpha \,\} = \emptyset$. When we find $\bar{\boldsymbol{\lambda}} \in \Lambda$ such that $\sigma(\bar{\boldsymbol{\lambda}}) = \tau_\alpha(\bar{\boldsymbol{\lambda}})$, the face $\{\, \boldsymbol{x} \in X \mid \bar{\boldsymbol{\lambda}}\boldsymbol{x} = \sigma(\bar{\boldsymbol{\lambda}}) \,\}$ of X contains a point $\boldsymbol{w} \in X_M \cap X_V$ with $\boldsymbol{d}\boldsymbol{w} \leqq \alpha$, and hence we will obtain this point by solving $\max\{\, \bar{\boldsymbol{\lambda}}\boldsymbol{x} \mid \boldsymbol{x} \in X,\ \boldsymbol{d}\boldsymbol{x} \leqq \alpha \,\}$ with additional computation if necessary.

Furthermore, by evaluating $\tau_\alpha(\boldsymbol{\lambda})$ for every vertex $(\boldsymbol{\lambda}, \mu)$ of $\operatorname{epi}\sigma$, we can either find $\bar{\boldsymbol{\lambda}} \in \Lambda$ such that $\sigma(\bar{\boldsymbol{\lambda}}) = \tau_\alpha(\bar{\boldsymbol{\lambda}})$ or identify that $\{\, \boldsymbol{x} \in X_M \mid \boldsymbol{d}\boldsymbol{x} \leqq \alpha \,\} = \emptyset$. In fact we do not need to enumerate every vertex $(\boldsymbol{\lambda}, \mu)$ of $\operatorname{epi}\sigma$ by applying an outer approximation method of $\operatorname{epi}\sigma$. For a nonempty subset W of $X_M \cap X_V$ and $\boldsymbol{\lambda} \in \Lambda$ let

$$\sigma_W(\boldsymbol{\lambda}) = \max\{\, \boldsymbol{\lambda}\boldsymbol{w} \mid \boldsymbol{w} \in W \,\}, \tag{4.9}$$

and

$$\operatorname{epi}\sigma_W = \{\, (\boldsymbol{\lambda}, \mu) \in \Lambda \times \mathbb{R} \mid \mu - \boldsymbol{\lambda}\boldsymbol{w} \geqq 0 \text{ for all } \boldsymbol{w} \in W \,\}. \tag{4.10}$$

Note that for a sequence $\{W_k\}_{k=0}^K$ of the subset of $X_M \cap X_V$ such that $\emptyset \neq W_0 \subsetneqq W_1 \subsetneqq \cdots \subsetneqq W_K = X_M \cap X_V$, we have $\sigma_{W_0}(\boldsymbol{\lambda}) \leqq \sigma_{W_1}(\boldsymbol{\lambda}) \leqq \cdots \leqq \sigma_{W_K}(\boldsymbol{\lambda}) = \sigma(\boldsymbol{\lambda})$ for all $\boldsymbol{\lambda} \in \Lambda$, and hence we have also that $\operatorname{epi}\sigma_{W_0} \supseteqq \operatorname{epi}\sigma_{W_1} \supseteqq \cdots \supseteqq \operatorname{epi}\sigma_{W_K} = \operatorname{epi}\sigma$. Suppose that we have known the vertex set of $\operatorname{epi}\sigma_{W_k}$. If $\operatorname{epi}\sigma_{W_k} \subsetneqq \operatorname{epi}\tau_\alpha$, that

is identified by evaluating $\tau_\alpha(\boldsymbol{\lambda})$ for every $(\boldsymbol{\lambda}, \mu)$ of the vertex of epi σ_{W_k}, then we conclude that $\sigma(\boldsymbol{\lambda}) > \tau_\alpha(\boldsymbol{\lambda})$ for all $\boldsymbol{\lambda} \in \Lambda$ since epi $\sigma \subsetneqq$ epi $\sigma_{W_k} \subsetneqq$ epi τ_α. Otherwise we choose $\boldsymbol{w} \in (X_M \cap X_V) \setminus W_k$, set W_{k+1} to $W_k \cup \{\boldsymbol{w}\}$, and then compute the vertex set of epi $\sigma_{W_{k+1}}$ from the knowledge of the vertex set of epi σ_{W_k}. We repeat the above computing until either we find $\bar{\boldsymbol{\lambda}} \in \Lambda$ such that $\sigma(\bar{\boldsymbol{\lambda}}) = \tau_\alpha(\bar{\boldsymbol{\lambda}})$ or we identify that epi $\sigma_{W_k} \subsetneqq$ epi τ_α.

Here we describe the algorithm as follows.

/** outer approximation algorithm of parameter set for (mmF) **/

$\langle 0 \rangle$ (initialization) Find a feasible vertex $\boldsymbol{w}^0 \in X_M \cap X_V$. Set $W_0 := \{\boldsymbol{w}^0\}$ and compute the vertex set V_0 of epi σ_{W_0}. Set $\ell := 0$, $k := 0$ and go to $\langle\langle \ell \rangle\rangle$.

$\langle\langle \ell \rangle\rangle$ (local search) Apply the local search procedure starting from $\boldsymbol{w}^\ell$ to obtain an locally optimal vertex $\bar{\boldsymbol{x}}^{\ell+1} \in X_M \cap X_V$ with the value $\boldsymbol{d}\bar{\boldsymbol{x}}^{\ell+1}$. Set $\alpha := \boldsymbol{d}\bar{\boldsymbol{x}}^{\ell+1} - 1$, $\ell := \ell + 1$ and go to $\langle k \rangle$.

$\langle k \rangle$ (global technique) If $\tau_\alpha(\boldsymbol{\lambda}) < \mu$ for all $(\boldsymbol{\lambda}, \mu) \in V_k$ then stop ($\bar{\boldsymbol{x}}^\ell$ solves (mmF)). Otherwise, choose $(\boldsymbol{\lambda}^k, \mu_k) \in V_k$ such that $\tau_\alpha(\boldsymbol{\lambda}^k) \geqq \mu_k$.

$\langle k2 \rangle$ If $\sigma(\boldsymbol{\lambda}^k) = \tau(\boldsymbol{\lambda}^k)$ then solve $\max\{\, \boldsymbol{\lambda}^k \boldsymbol{x} \mid \boldsymbol{x} \in X,\ \boldsymbol{d}\boldsymbol{x} \leqq \alpha \,\}$ to obtain a point $\boldsymbol{w}^\ell \in X_M \cap X_V \cap \{\, \boldsymbol{x} \mid \boldsymbol{d}\boldsymbol{x} \leqq \alpha \,\}$, and go to $\langle\langle \ell \rangle\rangle$.

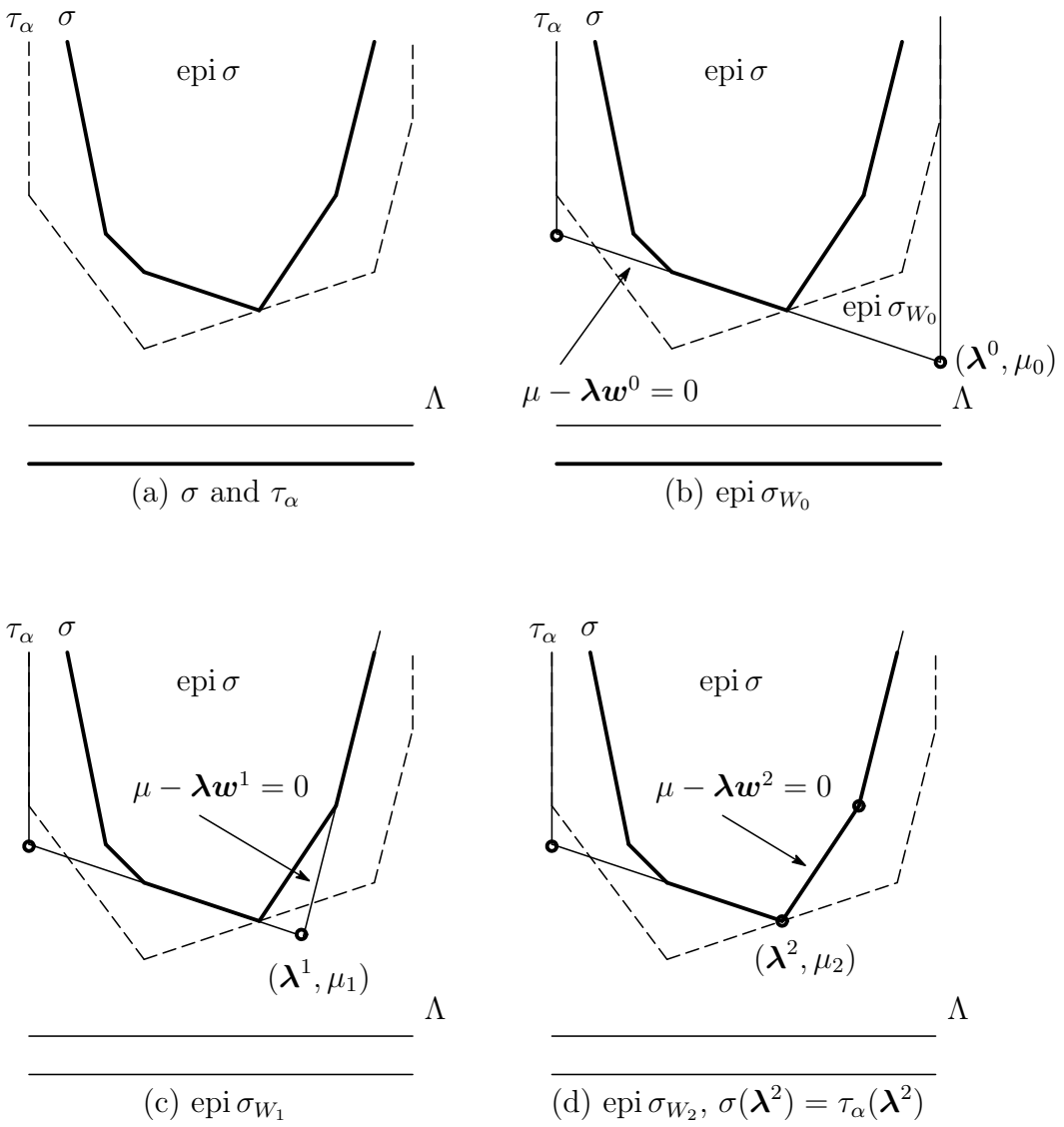
$\langle k3 \rangle$ Otherwise, solve $\max\{\, \boldsymbol{\lambda}^k \boldsymbol{x} \mid \boldsymbol{x} \in X \,\}$ to obtain a solution $\boldsymbol{v}^{k+1} \in X_M \cap X_V$. Set $W_{k+1} := W_k \cup \{\boldsymbol{v}^{k+1}\}$ and compute the vertex set V_{k+1} of epi $\sigma_{W_{k+1}}$. Set $k := k + 1$ and go to $\langle k \rangle$.

Theorem 4.6 *The outer approximation algorithm of parameter set for (mmF) works correctly and terminates after finitely many iteration.*

Proof: See Shigeno-Takahashi-Yamamoto [48] for the proof. □

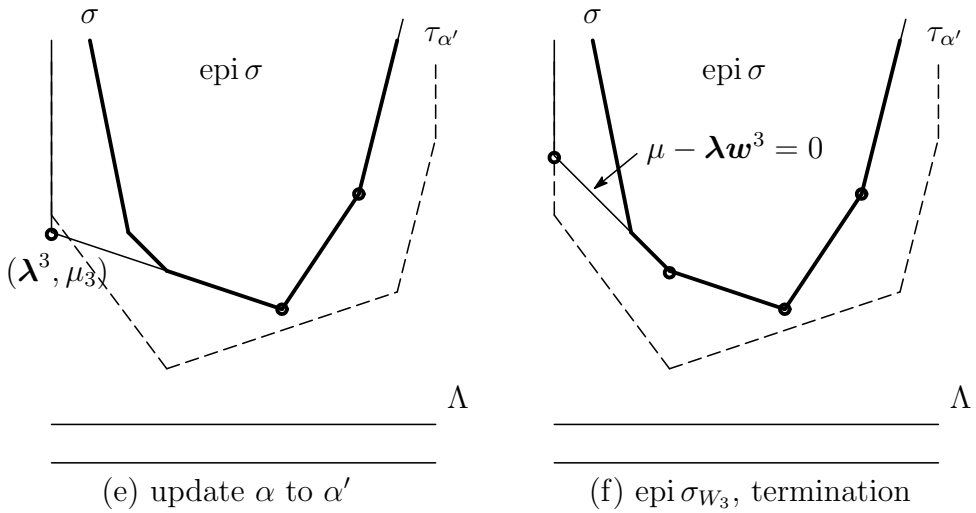
We illustrate the algorithm in Figure 4.3. For σ and τ_α of (a) and a feasible solution $\boldsymbol{w}^0$, we first construct epi σ_{W_0} by adding the constraint $\mu - \boldsymbol{\lambda}\boldsymbol{w}^0 \geqq 0$ to the set $\{\, (\boldsymbol{\lambda}, \mu) \mid (\boldsymbol{\lambda}, \mu) \in \Lambda \times \mathbb{R}_+ \,\}$ (See (b)). Since $\tau_\alpha(\boldsymbol{\lambda}^0) \geqq \mu_0$ and $\sigma(\boldsymbol{\lambda}^0) \neq \mu_0$, we obtain a solution $\boldsymbol{w}^1 \in \mathrm{argmax}\{\, \boldsymbol{\lambda}^k \boldsymbol{x} \mid \boldsymbol{x} \in X \,\}$, set $W_1 := W \cup \{\boldsymbol{w}^1\}$ and construct

epi σ_{W_1} (See (c)). In the similar way, we construct epi σ_{W_2} (See (d)). Since we find $(\boldsymbol{\lambda}^2, \mu_2)$ such that $\sigma(\boldsymbol{\lambda}^2) = \tau_\alpha(\boldsymbol{\lambda}^2)$, we update α to α' (See (e)). At last we stop the algorithm because we see that $\tau_\alpha(\boldsymbol{\lambda}) < \mu$ for every vertex $(\boldsymbol{\lambda}, \mu)$ of epi σ_{W_3} (See (f)).



4.3 D.C. Algorithm for (mmF)

Combining D.C. Algorithm (DCA) stated in Section 3.10, Muu-Shi [37] proposed the branch-and-bound based algorithm for (mmF). In this section we briefly review the algorithm.



(e) update α to α' (f) epi σ_{W_3}, termination

Figure 4.3: Outer approximation algorithm of parameter set for (mmF)

As stated in Section 3.8 and Subsection 4.1.1, (mmF) is equivalent to

$$(Q(\pi)) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{dx} - \pi(-g(\boldsymbol{x})) \\ \text{s.t.} & \boldsymbol{x} \in X, \end{array} \right.$$

for a sufficiently large $\pi > 0$. Muu-Shi [37] showed that (mmF) and $(Q(\pi))$ are equivalent for any $\pi > \pi^* := \max\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in X \,\} - \min\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in X \,\}$. Here we describe the algorithm.

/*** branch-and-bound algorithm with DCA for (mmF) ***/

$\langle 0 \rangle$ (initialization) Let S_0 be the simplex containing X. Find a feasible solution $\bar{\boldsymbol{x}} \in X_M$ and set $UB := \boldsymbol{d}\bar{\boldsymbol{x}}$. Set $LB := \min\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in X \,\}$, $\mathcal{R} := \mathcal{S} := \{S_0\}$ and $k := 0$.

$\langle k \rangle$ (iteration k)

$\langle k1 \rangle$ (upper bound) For each simplex $S \in \mathcal{S}$, find a locally optimal solution $\boldsymbol{x}^S$ for (mmF) on $X \cap S$ with $UB(S) := \boldsymbol{dx}^S$. If $UB(S) < UB$ then set $UB := UB(S)$ and $\bar{\boldsymbol{x}} := \boldsymbol{x}^S$.

$\langle k2 \rangle$ (lower bound) For each simplex $S \in \mathcal{S}$, calculate a lower bound $LB(S)$ for S. Set $LB := \min\{\, LB(S) \mid S \in \mathcal{R} \,\}$.

$\langle k3 \rangle$ (termination) Let $\mathcal{R} := \{ S \in \mathcal{R} \mid LB(S) < UB \}$. If either $UB - LB < \epsilon$ or $\mathcal{R} = \emptyset$ then stop ($\bar{\boldsymbol{x}}$ is an ϵ-optimal solution of (mmF)).

$\langle k4 \rangle$ (subdivision) Choose $S^* \in \mathcal{R}$ and divide S^* into subsimplices. Let $\mathcal{S}$ be the partition by this division. Set $\mathcal{R} := \mathcal{R} \setminus \{S^*\}$, $k := k + 1$ and go to $\langle k \rangle$.

Remark 4.7 *We apply DCA to the problem*

$$(Q(S)) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{d}\boldsymbol{x} - \pi(-g(\boldsymbol{x})) \\ s.t. & \boldsymbol{x} \in X \cap S, \end{array} \right.$$

to find a locally optimal solution $\boldsymbol{x}^S$

Remark 4.8 *Let* S_V *be the vertex set of a simplex* S*. To calculate a lower bound* $LB(S)$ *we find an affine function* $l_S(\boldsymbol{x})$ *such that* $l_S(\boldsymbol{v}) = \pi(-g(\boldsymbol{v}))$ *for all* $\boldsymbol{v} \in S_V$*, and solve*

$$(L(S)) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{d}\boldsymbol{x} - l_S(\boldsymbol{x}) \\ s.t. & \boldsymbol{x} \in X \cap S. \end{array} \right.$$

Chapter 5

New Algorithms for the Minimum Maximal Flow Problem

In this chapter we propose two algorithms for solving (mmF), that are based on the CS method and the OA method stated in Chapter 2. Most of the existing algorithms, including the algorithms stated in the previous chapter, for (mmF) are mainly based on the techniques for (P_E). These methods anticipate a small number p of criteria of Problem (MC) and convert the problem to a global optimization problem in variables of the number of criteria or so. The number of criteria in (mmF) is, however, equal to the number of arcs, i.e., $p = n$. Hence these methods usually do not work efficiently for (mmF). On the other hand our algorithm proposed in this chapter does not depend on the number of criteria. Then, compared with the existing algorithms, our algorithm is advantageous for (mmF).

We put assumptions on a given network in the next section, in which we also mention an implementation issue about generating problems. In Section 5.2 after extending the domain of the gap function characterizing the set of maximal flows, we state a local search procedure using the extended gap function. In Section 5.3 we propose the CS method for (mmF), which is studied in Yamamoto-Zenke [67, 68]. The convergence of the CS method is discussed. In Section 5.4 we propose the OA method for (mmF), which is studied in Yamamoto-Zenke [67, 69]. The OA method, based on the idea of ε-optimal solution and local search technique, terminates after finitely many iterations with the optimal value of the problem. Some implementation issues about these algorithms are also studied. Furthermore

we improve the algorithms by reducing the number of variables in Section 5.5. In Section 5.6 we report the results of the computational experiment. We propose a heuristics to locate an initial incumbent and report some numerical results of the improved CS method and improved OA method. We further compare the improved OA method with the method in Section 4.2. We observe that the improved OA method works efficiently for many problems and surpasses both an application of vertex enumeration method and the method in Section 4.2 in computational time, especially as the number of variables grows. We also observed that our heuristics using the modified local search procedure provides a pretty good initial incumbent.

5.1 Assumptions on a Given Network

For simplicity we assume throughout the chapter that a given network $\mathcal{N} = (V, E, \boldsymbol{c})$ satisfies the following three assumptions.

Assumption 5.1

(i) *Each capacity is a positive integer, i.e.,* $c_h \in \mathbb{Z}$ *and* $c_h > 0$ *for each* $h \in E$.

(ii) *There is no* t*-*s*-path.*

(iii) *For all arc* $h = (u, v)$*, there are a* s*-*u*-path and a* v*-*t*-path.*

Assumption 5.1 (i) and (ii) are not so restrictive. In Subsection 5.5.6 and Subsection 5.5.7, respectively, we will discuss the modification of our algorithms for the case where Assumption 5.1 (i) and (ii) do not hold. Assumption 5.1 (iii) might appear to confine the scope of application, however, performing the following operation to the given network, we can construct a new network $\mathcal{N}' = (V', E', \boldsymbol{c}')$ that satisfies this assumption without a substantial change of the problem. Let (mmF) and (mmF') be the minimum maximal flow problems defined for $\mathcal{N}$ and $\mathcal{N}'$, respectively. Problems (mmF') and (mmF) have the same optimal value, and an optimal solution of (mmF) can be easily obtained from an optimal solution of (mmF').

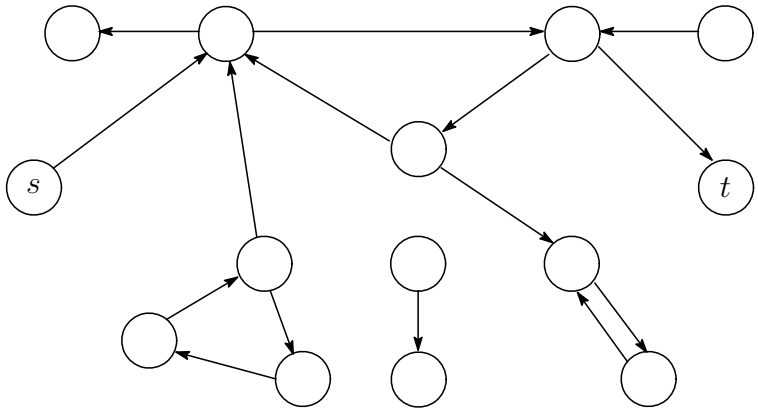
/** operation to obtain a network $\mathcal{N}'$ satisfying Assumption 5.1 (iii) **/

⟨1⟩ (construction of T_1) Construct the graph $G_1 := (V_1, E_1)$, where $V_1 := V \setminus \{t\}$ and $E_1 := E \setminus \{\, h \in E \mid \partial^- h = t \,\}$. Obtain the directed tree T_1 by applying the depth search algorithm with root s on the graph G_1.

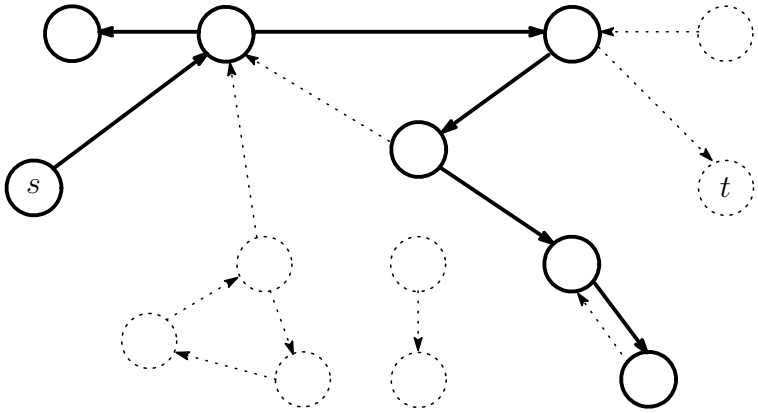
⟨2⟩ (construction of T_2) Let $\bar{E} := \{\, (v,u) \mid (u,v) \in E \,\}$. Construct the graph $G_2 := (V_2, E_2)$, where $V_2 := V \setminus \{s\}$ and $E_2 := \bar{E} \setminus \{\, h \in \bar{E} \mid \partial^- h = s \,\}$. Obtain the directed tree T_2 by applying the depth search algorithm with root t on the graph G_2.

⟨3⟩ (construction of G') Set $V' := \{\, v \in V \mid v$ is contained in both T_1 and $T_2 \,\} \cup \{s,t\}$, $E' := \{\, h \in E \mid \partial^+ h \in V'$ and $\partial^- h \in V' \,\}$, and $\boldsymbol{c}' := \left[c_h\right]_{h \in E'}$. (the network $\mathcal{N}' = (V', E', \boldsymbol{c}')$ is obtained).

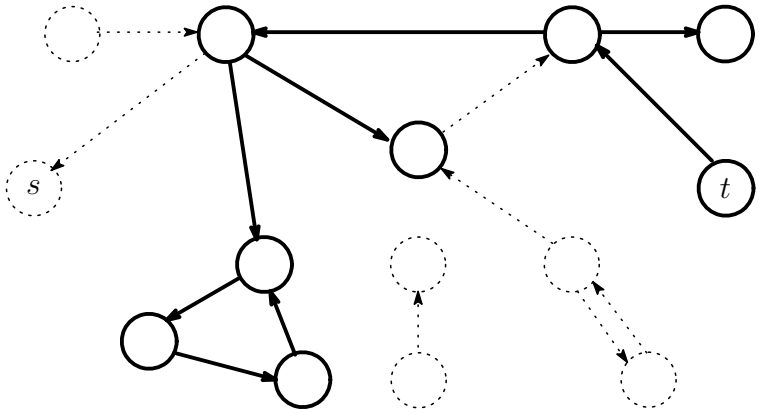
When we are given a directed network $\mathcal{N} = (V, E, \boldsymbol{c})$ in Figure 5.1 (a), we construct trees T_1 of (b) and T_2 of (c), and we then obtain the network $\mathcal{N}' = (V', E', \boldsymbol{c}')$ of (d). As readily seen from the way of the construction, $\mathcal{N}'$ satisfies Assumption 5.1 (iii). In Figure 5.1 (e), we have five connected components, say $C_1, \cdots, C_5$, deleted by the operation. Let $C_j = (V_j, E_j, \boldsymbol{c}^j)$ for each $j = 1, \ldots, 5$, and let $E(G') = \{\, h \in E \mid h$ links G' and one of $C_1, \cdots, C_5 \,\}$. Then the solution $\boldsymbol{x}$ of (mmF) is splitted as $\boldsymbol{x} = (\boldsymbol{x}', \boldsymbol{x}^1, \cdots, \boldsymbol{x}^5, \boldsymbol{x}^{E(G')})$, where $\boldsymbol{x}' = \left[x_h\right]_{h \in E'}$, $\boldsymbol{x}^j = \left[x_h\right]_{h \in E_j}$ for $j = 1, \ldots, 5$ and $\boldsymbol{x}^{E(G')} = \left[x_h\right]_{h \in E(G')}$. Obviously, $\boldsymbol{x}^{E(G')} = \boldsymbol{0}$ whenever $\boldsymbol{x}$ is a feasible flow. Hence $\boldsymbol{x}^j$ for each $j = 1, \ldots, 5$ has no affect on the optimal value of (mmF). Therefore once we obtain an optimal solution $\boldsymbol{x}'$ of (mmF'), we can easily obtain an optimal solution of (mmF). Namely, for each $j = 1, \ldots, 5$ we construct the network $\mathcal{N}_j = (V_j \cup \{s,t\}, E_j \cup \{(s,u_j), (v_j,t)\}, (\boldsymbol{c}^j, \infty, \infty))$ for arbitrarily chosen nodes $u_j, v_j \in V_j$ and solve the maximum flow problem on $\mathcal{N}_j$ to obtain the solution $\boldsymbol{x}^j$ (See (f)). In this way an optimal solution $\boldsymbol{x} = (\boldsymbol{x}', \boldsymbol{x}^1, \cdots, \boldsymbol{x}^5, \boldsymbol{0})$ of (mmF) is obtained.



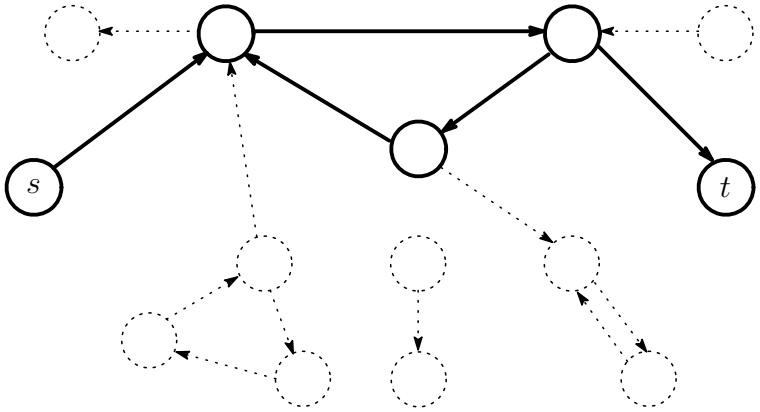
(a) Given network $\mathcal{N} = (V, E, \boldsymbol{c})$



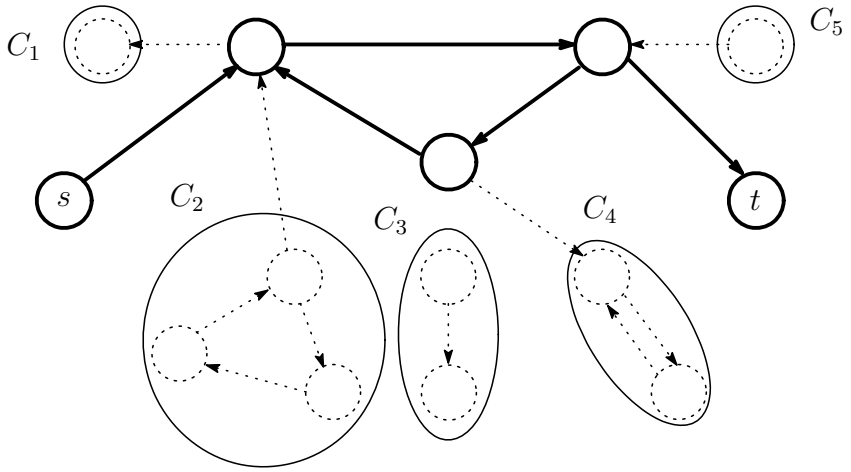
(b) Tree T_1



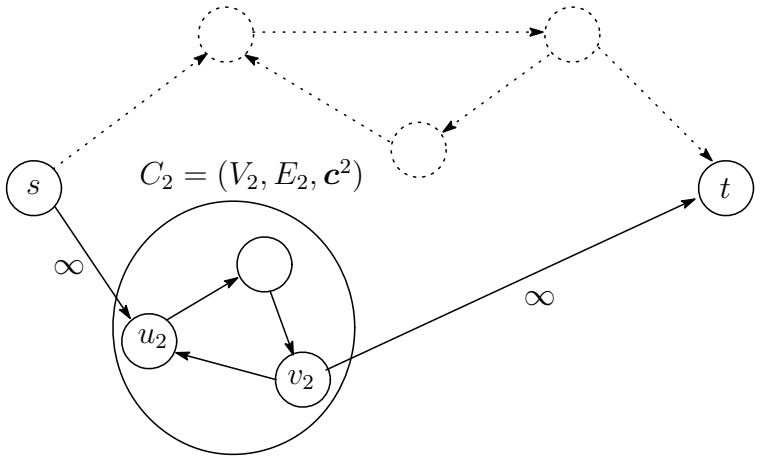
(c) Tree T_2



(d) Obtained network $\mathcal{N}' = (V', E', \boldsymbol{c}')$



(e) Connected components



(f) Network $\mathcal{N}_2 = (V_2 \cup \{s, t\}, E_2 \cup \{(s, u_2), (v_2, t)\}, (\boldsymbol{c}^2, \infty, \infty))$

Figure 5.1: Operation for Assumption 5.1 (iii)

Assumption 5.1 implies the following lemma.

Lemma 5.2

(*i*) *Each element of a vertex of* X *is integer, so is the optimal value of* (mmF).

(*ii*) $\min\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in X \,\} = 0$.

(*iii*) *There is a vector* $\boldsymbol{x}$ *such that* $A\boldsymbol{x} = \mathbf{0}$ *and* $\mathbf{0} < \boldsymbol{x} < \boldsymbol{c}$, *and hence we have:*

(*a*) $\mathbf{0} \notin X_M$.

(*b*) *The dimension of* X *is* $n - m$.

Proof: (*i*) Assumption 5.1 (i) ensures the integrality of vertices of X as well as the optimal value of (mmF).

(*ii*) The assertion is straightforward by Assumption 5.1 (ii).

(*iii*) By Assumption 5.1 (iii), for each arc $h = (u, v) \in E$, there are a s-u-path P_{su} and a v-t-path P_{vt}. For each arc $h = (u, v) \in E$, we define $\boldsymbol{x}^{uv} = \left[x_i^{uv}\right]$, $\boldsymbol{x}^{su} = \left[x_i^{su}\right]$ and $\boldsymbol{x}^{vt} = \left[x_i^{vt}\right]$, where

$$x_i^{uv} = \begin{cases} \varepsilon & \text{if arc } i = h \\ 0 & \text{otherwise,} \end{cases}$$

$$x_i^{su} = \begin{cases} \varepsilon & \text{if arc } i \text{ is contained in } P_{su} \\ 0 & \text{otherwise, and} \end{cases}$$

$$x_i^{vt} = \begin{cases} \varepsilon & \text{if arc } i \text{ is contained in } P_{vt} \\ 0 & \text{otherwise,} \end{cases}$$

for $\varepsilon = \min\{\, c_j \mid j = 1, \ldots, n \,\}/(2m)$, and we then define $\boldsymbol{x}^h = \boldsymbol{x}^{uv} + \boldsymbol{x}^{su} + \boldsymbol{x}^{vt}$ and $\boldsymbol{x} = \sum_{h \in E} \boldsymbol{x}^h$. It is easily seen that $A\boldsymbol{x} = \mathbf{0}$ and $\mathbf{0} < \boldsymbol{x} < \boldsymbol{c}$.

(*a*) Since there is a vector $\boldsymbol{x}$ such that $A\boldsymbol{x} = \mathbf{0}$ and $\mathbf{0} < \boldsymbol{x} < \boldsymbol{c}$, we see that there is $\boldsymbol{x} \in X$ such that $\boldsymbol{x} \geqq \mathbf{0}$ and $\boldsymbol{x} \neq \mathbf{0}$. This implies that $\mathbf{0} \notin X_M$.

(*b*) Assumption 5.1 (iii) ensures that the given network is connected, and hence the

incidence matrix has full rank, i.e., $\operatorname{rank} A = m$. Then the dimension of the affine space $L = \{ \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \mathbf{0} \}$ is $n - m$. Since $X \subseteqq L$, we have $\dim X \leqq n - m$. Let $\boldsymbol{x}$ be the n-dimensional vector such that $A\boldsymbol{x} = \mathbf{0}$ and $\mathbf{0} < \boldsymbol{x} < \boldsymbol{c}$. There is $\delta > 0$ such that $\mathbf{0} < \boldsymbol{y} < \boldsymbol{c}$ for all $\boldsymbol{y} \in N_\delta(\boldsymbol{x})$. Since $N_\delta(\boldsymbol{x}) \cap L \subseteqq X$ and $\dim (N_\delta(\boldsymbol{x}) \cap L) = n - m$, we have $\dim X \geqq n - m$. □

5.1.1 Implementation issue about generating problems

In this subsection we discuss an implementation issue about generating problems. Given integers m and n, we need to make a graph $G = (V, E)$ that satisfies Assumption 5.1 (ii)-(iii), with $|V| = m + 2$, $|E| = n$. For the credibility of computational experiments we should generate a graph at random. Out of lots of ideas of generating graphs, we adopt the following method in our study.

/** procedure of generating a graph **/

⟨0⟩ (initialization) Input the number of nodes m and the number of arcs n. Set $V := \{1, \ldots, m\} \cup \{s, t\}$ and $E := \emptyset$.

⟨k⟩ (adding phase) Choose two nodes $u, v \in V$ at random. If one of the following cases:

(a) $u = s$ and $v = t$

(b) $(u, v) \in E$

(c) the graph $G = (V, E \cup \{(u, v)\})$ has a t-s-path

occurs, go to ⟨k⟩. Otherwise, set $E := E \cup \{(u, v)\}$.

⟨$k1$⟩ (construction of T_1 and T_2) Set $G_1 := (V \setminus \{t\}, E_2)$ and $G_2 := (V \setminus \{s\}, E_2)$ be directed graphs, where $E_1 := \{ h \in E \mid \partial^- h \neq t \}$ and $E_2 := \{ (v, u) \mid h = (u, v) \in E,\ \partial^+ h \neq s \}$. Apply the depth search algorithm starting from s on G_1 and starting form t on G_2, to construct trees T_1 and T_2, respectively.

⟨$k2$⟩ (checking Assumption 5.1 (iii)) If both of the trees are spanning, meaning G holds Assumption 5.1 (iii), then go to ⟨⟨ℓ⟩⟩, otherwise go to ⟨k⟩.

$\langle\langle \ell \rangle\rangle$ (deleting phase) If $|E| = n$ then stop (we obtain a graph $G = (V, E)$). Otherwise, choose $(u, v) \in E$ at random, and set $E := E \setminus \{(u, v)\}$. Run $\langle k1 \rangle$ again to obtain T_1 and T_2. If both of the trees are spanning, meaning G holds Assumption 5.1 (iii), then go to $\langle\langle \ell \rangle\rangle$. Otherwise set $E := E \cup \{(u, v)\}$ and go to $\langle\langle \ell \rangle\rangle$.

5.2 Extension of Gap Function, and Local Search Procedure Using the Function

Since $g(\boldsymbol{v}) = -\infty$ if there is no point $\boldsymbol{y} \in X$ such that $\boldsymbol{y} \geqq \boldsymbol{v}$, no information is available about how far the point $\boldsymbol{v}$ is from the domain of g. Exploiting the totally unimodularity of the incidence matrix, Yamamoto-Zenke [67,68] extend the domain of the gap function g to $\mathbb{R}^n$. In this section we explain the extension of the gap function and describe a local search procedure using the extended gap function.

5.2.1 Extension of gap function

The gap function $g(\boldsymbol{x})$ of (4.3) is given by the optimal value of the problem

$$(P_G(\boldsymbol{x})) \quad \left| \begin{array}{ll} \max\limits_{\boldsymbol{y}} & \boldsymbol{e}\boldsymbol{y} - \boldsymbol{e}\boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{y} = \boldsymbol{0},\ \boldsymbol{0} \leqq \boldsymbol{y} \leqq \boldsymbol{c} \\ & \boldsymbol{y} \geqq \boldsymbol{x}, \end{array} \right.$$

whose dual problem is

$$(D_G(\boldsymbol{x})) \quad \left| \begin{array}{ll} \min\limits_{(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})} & \boldsymbol{\alpha}\boldsymbol{c} - \boldsymbol{\beta}\boldsymbol{x} - \boldsymbol{e}\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{\pi}A + \boldsymbol{\alpha} - \boldsymbol{\beta} \geqq \boldsymbol{e} \\ & \boldsymbol{\alpha}, \boldsymbol{\beta} \geqq \boldsymbol{0}. \end{array} \right.$$

Note that $(D_G(\boldsymbol{x}))$ is always feasible, e.g., take $\boldsymbol{\pi} = \boldsymbol{\beta} = \boldsymbol{0}$ and $\boldsymbol{\alpha} \geqq \boldsymbol{e}$. Therefore, $(P_G(\boldsymbol{x}))$ is infeasible if and only if $(D_G(\boldsymbol{x}))$ is unbounded. Adding the upper bound constraints $\boldsymbol{\beta} \leqq \bar{\boldsymbol{\beta}}$ to $(D_G(\boldsymbol{x}))$ yields the following problem

$$(\overline{D_G(\boldsymbol{x})}) \quad \left| \begin{array}{ll} \min\limits_{(\boldsymbol{\pi},\boldsymbol{\alpha},\boldsymbol{\beta})} & \boldsymbol{\alpha c} - \boldsymbol{\beta x} - \boldsymbol{ex} \\ \text{s.t.} & \boldsymbol{\pi} A + \boldsymbol{\alpha} - \boldsymbol{\beta} \geqq \boldsymbol{e} \\ & \boldsymbol{\alpha} \geqq \boldsymbol{0},\ \boldsymbol{0} \leqq \boldsymbol{\beta} \leqq \bar{\boldsymbol{\beta}}, \end{array} \right.$$

where $\bar{\boldsymbol{\beta}} \geqq \boldsymbol{0}$ will be specified in the following theorem. The dual problem of $(\overline{D_G(\boldsymbol{x})})$ is

$$(\overline{P_G(\boldsymbol{x})}) \quad \left| \begin{array}{ll} \max\limits_{(\boldsymbol{y},\boldsymbol{t})} & \boldsymbol{ey} - \boldsymbol{ex} - \bar{\boldsymbol{\beta}}\boldsymbol{t} \\ \text{s.t.} & A\boldsymbol{y} = \boldsymbol{0},\ \boldsymbol{0} \leqq \boldsymbol{y} \leqq \boldsymbol{c} \\ & \boldsymbol{y} + \boldsymbol{t} \geqq \boldsymbol{x},\ \boldsymbol{t} \geqq \boldsymbol{0}. \end{array} \right.$$

Then we define the *extended gap function* $\bar{g} : \mathbb{R}^n \to \mathbb{R}$ as

$$\bar{g}(\boldsymbol{x}) = \max\{\, \boldsymbol{ey} - \bar{\boldsymbol{\beta}}\boldsymbol{t} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} + \boldsymbol{t} \geqq \boldsymbol{x},\ \boldsymbol{t} \geqq \boldsymbol{0} \,\} - \boldsymbol{ex}. \tag{5.1}$$

Theorem 5.3 *(i) The domain of $\bar{g}$ is $\mathbb{R}^n$.*

(ii) If $\bar{\boldsymbol{\beta}} \geqq n\boldsymbol{e}$ then $\bar{g} = g$ on the domain of g.

Proof: *(i)* For any $\boldsymbol{x} \in \mathbb{R}^n$, $(\overline{D_G(\boldsymbol{x})})$ has a feasible solution and the objective function is bounded. By the duality theorem of linear programming there is an optimal value of $(\overline{P_G(\boldsymbol{x})})$, and hence $\bar{g}(\boldsymbol{x}) > -\infty$ for any $\boldsymbol{x} \in \mathbb{R}^n$.

(ii) Let Ω and $\bar{\Omega}$ denote the feasible sets of $(D_G(\boldsymbol{x}))$ and $(\overline{D_G(\boldsymbol{x})})$, respectively, i.e.,

$$\Omega = \{\, (\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{R}_{m+2n} \mid \boldsymbol{\pi} A + \boldsymbol{\alpha} - \boldsymbol{\beta} \geqq \boldsymbol{e},\ \boldsymbol{\alpha}, \boldsymbol{\beta} \geqq \boldsymbol{0} \,\}, \text{ and} \tag{5.2}$$

$$\bar{\Omega} = \{\, (\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{R}_{m+2n} \mid \boldsymbol{\pi} A + \boldsymbol{\alpha} - \boldsymbol{\beta} \geqq \boldsymbol{e},\ \boldsymbol{\alpha} \geqq \boldsymbol{0},\ \boldsymbol{0} \leqq \boldsymbol{\beta} \leqq \bar{\boldsymbol{\beta}} \,\}. \tag{5.3}$$

By the theory of linear programming, if $\bar{\boldsymbol{\beta}}$ is so large that every vertex $\boldsymbol{v} = (\boldsymbol{\pi}_v, \boldsymbol{\alpha}_v, \boldsymbol{\beta}_v)$ of Ω satisfies $\boldsymbol{\beta}_v \leqq \bar{\boldsymbol{\beta}}$ then $\bar{\Omega}$ contains every vertex of Ω, and hence we have $\bar{g}(\boldsymbol{x}) = g(\boldsymbol{x})$ for all $\boldsymbol{x}$ in the domain of g. Replacing $\boldsymbol{\pi}$ by $\boldsymbol{\pi}^1 - \boldsymbol{\pi}^2$ with $\boldsymbol{\pi}^1, \boldsymbol{\pi}^2 \geqq \boldsymbol{0}$ and introducing a slack variable vector $\boldsymbol{\gamma} \geqq \boldsymbol{0}$, we rewrite Ω as

$$\Omega = \left\{ \left[\begin{array}{c} (\boldsymbol{\pi}^1)^\top \\ (\boldsymbol{\pi}^2)^\top \\ \boldsymbol{\alpha}^\top \\ \boldsymbol{\beta}^\top \\ \boldsymbol{\gamma}^\top \end{array} \right] \middle| \left[\begin{array}{ccccc} A^\top & -A^\top & I & -I & -I \end{array} \right] \left[\begin{array}{c} (\boldsymbol{\pi}^1)^\top \\ (\boldsymbol{\pi}^2)^\top \\ \boldsymbol{\alpha}^\top \\ \boldsymbol{\beta}^\top \\ \boldsymbol{\gamma}^\top \end{array} \right] = \boldsymbol{1},\ \left[\begin{array}{c} (\boldsymbol{\pi}^1)^\top \\ (\boldsymbol{\pi}^2)^\top \\ \boldsymbol{\alpha}^\top \\ \boldsymbol{\beta}^\top \\ \boldsymbol{\gamma}^\top \end{array} \right] \geqq \boldsymbol{0} \right\}.$$

Let $\boldsymbol{v}$ be a vertex of Ω. Then it is a basic solution of the system defining Ω, i.e., $\boldsymbol{v} = (\boldsymbol{w}^B, \boldsymbol{w}^N) = (B^{-1}\mathbf{1}, \mathbf{0})$ for some nonsingular $n \times n$ submatrix B of $\begin{bmatrix} A^\top & -A^\top & I & -I & -I \end{bmatrix}$. Since the incidence matrix A is totally unimodular, i.e., each subdeterminant of A is -1, 0, or $+1$, so is $\begin{bmatrix} A^\top & -A^\top & I & -I & -I \end{bmatrix}$. Therefore the matrix B^{-1} is composed of -1, 0 and $+1$, and hence $B^{-1}\mathbf{1} \leqq n\mathbf{1}$. This completes the proof. □

By Theorem 5.3, we hereafter fix $\bar{\boldsymbol{\beta}} = n\boldsymbol{e}$, and we replace the constraint $g(\boldsymbol{x}) \leqq 0$ in (mmF) with $\bar{g}(\boldsymbol{x}) \leqq 0$ to obtain an equivalent formulation of (mmF):

$$(mmF) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{d}\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in X,\ \bar{g}(\boldsymbol{x}) \leqq 0, \end{array} \right.$$

which is further equivalent to

$$(mmF) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{d}\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in X \setminus \operatorname{int} \bar{G}, \end{array} \right.$$

where

$$\bar{G} = \{ \boldsymbol{x} \in \mathbb{R}^n \mid \bar{g}(\boldsymbol{x}) \geqq 0 \}. \tag{5.4}$$

5.2.2 Local search using the extended gap function

For $\boldsymbol{v} \in X_M \cap X_V$, we define the set of all efficient vertices linked to $\boldsymbol{v}$ by

$$\begin{aligned} N_M(\boldsymbol{v}) &= \{ \boldsymbol{v}' \in X_M \cap X_V \mid [\boldsymbol{v}, \boldsymbol{v}'] \text{ is an edge of } X \} \\ &= \{ \boldsymbol{v}' \in X_V \mid [\boldsymbol{v}, \boldsymbol{v}'] \text{ is an edge of } X \text{ and } \bar{g}(\boldsymbol{v}') \leqq 0 \}. \end{aligned} \tag{5.5}$$

Note that the above inequality $\bar{g}(\boldsymbol{v}') \leqq 0$ can be replaced with $g(\boldsymbol{v}') \leqq 0$, however, $\bar{g}(\boldsymbol{v}') \leqq 0$ is more convenient because we can easily obtain the initial basis for solving the linear programming problem $\bar{g}(\boldsymbol{v}')$ while we often need to run phase I of two-phase simplex method to obtain the initial basis for solving $g(\boldsymbol{v}')$.

When we find a feasible solution $\boldsymbol{w} \in X_M$, we apply the local search procedure starting with $\boldsymbol{w}$ (LS($\boldsymbol{w}$) for short) for further improvement. The procedure is described as follows.

/** LS($\boldsymbol{w}$) procedure **/

$\langle 0 \rangle$ (initialization) If $\boldsymbol{w} \notin X_V$ then solve $\min\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in F(\boldsymbol{w}) \,\}$, where $F(\boldsymbol{w})$ is the face of X containing $\boldsymbol{w}$ in its relative interior, to obtain a vertex $\boldsymbol{v}^0 \in X_M \cap X_V$, otherwise set $\boldsymbol{v}^0 := \boldsymbol{w}$. Set $k := 0$.

$\langle k \rangle$ (iteration k) If $\{\, \boldsymbol{v} \in N_M(\boldsymbol{v}^k) \mid \boldsymbol{dv} < \boldsymbol{dv}^k \,\} = \emptyset$ then stop ($\boldsymbol{v}^k$ is an locally optimal vertex of (mmF)). Otherwise choose $\boldsymbol{v}^{k+1}$ in the set, set $k := k + 1$ and go to $\langle k \rangle$.

Remark 5.4 *Whenever $\boldsymbol{w} \in X_M$, the face $F(\boldsymbol{w})$ of X containing $\boldsymbol{w}$ in its relative interior is contained in X_M by Theorem 3.6.*

Figure 5.2 shows a three-dimensional example of the LS($\boldsymbol{w}$) procedure. In this figure X_E is the union of faces of X framed by bold lines. We see that $\boldsymbol{w}$ is not a vertex of X (See (a)). Then we consider the face $F(\boldsymbol{w})$ of X containing $\boldsymbol{w}$ in its interior (See (b)). We solve $\min\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in F(\boldsymbol{w}) \,\}$ to obtain the vertex $\boldsymbol{v}^0 \in X_E \cap X_V$ (See (c)). Since there is the vertex $\boldsymbol{v}^1 \in N_M(\boldsymbol{v}^0)$ such that $\boldsymbol{dv}^1 < \boldsymbol{dv}^0$, we go to the next iteration. Finally we stop the procedure at $\boldsymbol{v}^1$ because $\{\, \boldsymbol{x} \in N_M(\boldsymbol{v}^1) \mid \boldsymbol{dx} < \boldsymbol{dv}^1 \,\} = \emptyset$ (See (d)).

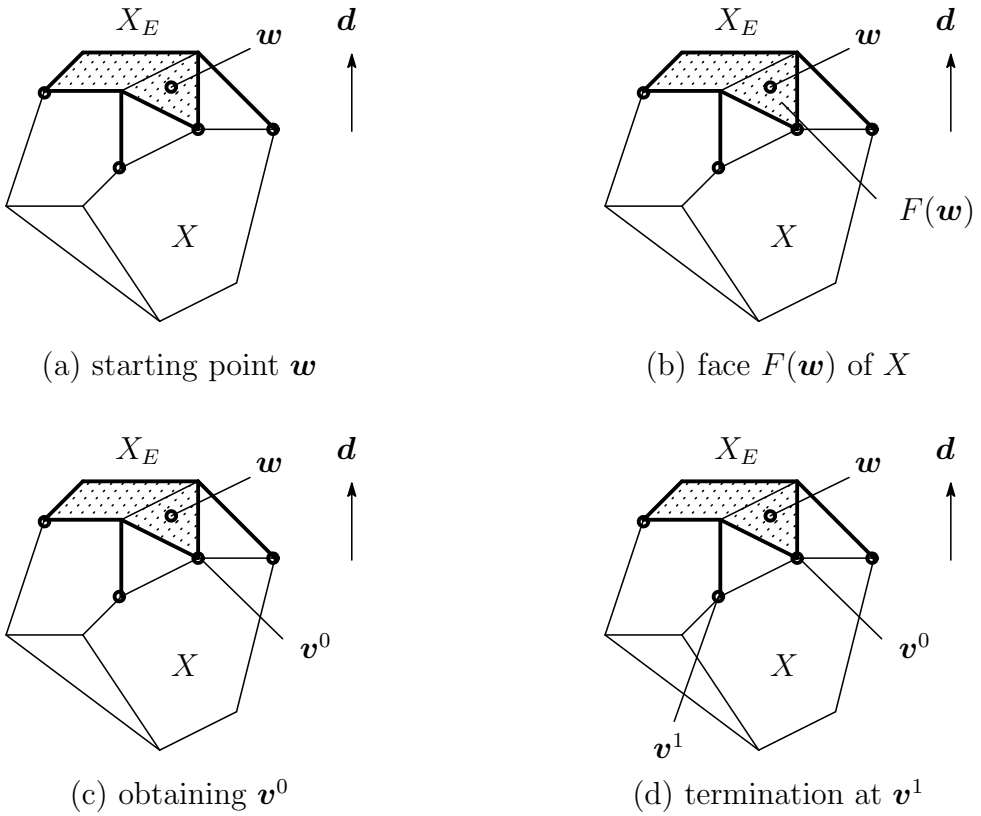
5.2.3 Implementation issues about a local search procedure

In Step $\langle k \rangle$ of the LS($\boldsymbol{w}$) procedure, we choose a point $\boldsymbol{v}^{k+1}$ in $\{\, \boldsymbol{v} \in N_M(\boldsymbol{v}^k) \mid \boldsymbol{dv} < \boldsymbol{dv}^k \,\}$ and set $k := k+1$ unless this set is empty. To do this, we consider the linear programming problem

$$\left|\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{dx} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{0} \\ & \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c}. \end{array}\right.$$

Introducing slack variables $\boldsymbol{s} = (s_1, \cdots, s_n)$, we convert this problem to

$$\left|\begin{array}{ll} \min\limits_{(\boldsymbol{x},\boldsymbol{s})} & \boldsymbol{dx} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{0} \\ & \boldsymbol{x} + \boldsymbol{s} = \boldsymbol{c} \\ & (\boldsymbol{x}, \boldsymbol{s}) \geqq \boldsymbol{0}, \end{array}\right.$$



(a) starting point $\boldsymbol{w}$ (b) face $F(\boldsymbol{w})$ of X

(c) obtaining $\boldsymbol{v}^0$ (d) termination at $\boldsymbol{v}^1$

Figure 5.2: Local search procedure LS($\boldsymbol{w}$)

which is further written as

$$\left|\begin{array}{ll} \min\limits_{\boldsymbol{y}} & \boldsymbol{d}_s\boldsymbol{y} \\ \text{s.t.} & A_s\boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{y} \geqq \boldsymbol{0}, \end{array}\right.$$

where

$$\boldsymbol{d}_s = \begin{bmatrix} \boldsymbol{d} & \boldsymbol{0} \end{bmatrix}, \quad A_s = \begin{bmatrix} A & O \\ I & I \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{s} \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{c} \end{bmatrix}.$$

The following form is called a *tableau* for the above problem.

A_s	$\boldsymbol{b}$
$\boldsymbol{d}_s$	min

Letting B and N be the basic and nonbasic matrices, respectively, of the system $A_s\boldsymbol{y} = \boldsymbol{b}$, the above tableau is rewritten as

B	N	$\boldsymbol{b}$
$\boldsymbol{d}^B$	$\boldsymbol{d}^N$	min

We assume, in Step $\langle k \rangle$, that $\boldsymbol{v}^k \in X_V$ and we have obtained the basic matrix B and the nonbasic matrix N corresponding to the solution $\boldsymbol{y} = (\boldsymbol{v}^k, \boldsymbol{c} - \boldsymbol{v}^k)$ at hand. Let $(\boldsymbol{y}^B, \boldsymbol{y}^N)$ be the partition of $\boldsymbol{y}$. Step $\langle k \rangle$ of the LS($\boldsymbol{w}$) procedure is implemented as follows. We compute the reduced cost $\bar{\boldsymbol{d}}^N = \boldsymbol{d}^N - \boldsymbol{d}^B B^{-1} N$. If $\bar{\boldsymbol{d}}^N \geqq \boldsymbol{0}$ we have done, otherwise we define

$$R = \{\, j \mid y_j^N \text{ is a nonbasic variable, } \bar{d}_j^N < 0 \,\}.$$

We choose $j \in R$, delete j from R and compute the following as long as R is not empty. We temporarily compute the basic solution $\boldsymbol{y}' = (\boldsymbol{x}', \boldsymbol{s}')$ adjacent to $\boldsymbol{y}$ by pivotting in the variable y_j^N. Let B' and N' be the basic matrix and the nonbasic matrix corresponding to the solution $\boldsymbol{y}'$. If $\bar{g}(\boldsymbol{x}') \leqq 0$, meaning $\boldsymbol{x}'$ is a maximal flow, then we update (B, N) to (B', N') and repeat Step $\langle k \rangle$.

If we have not obtained an initial basic matrix B and nonbasic matrix N corresponding to the solution $\boldsymbol{y}$, we need to solve $\min\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in F(\boldsymbol{w}) \,\}$ in Step $\langle 0 \rangle$. The detailed description of the problem $\min\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in F(\boldsymbol{w}) \,\}$ is

$$\left|\begin{array}{lll} \min\limits_{\boldsymbol{x}} & \boldsymbol{dx} & \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{0} & \\ & \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c} & \\ & \boldsymbol{e}^i\boldsymbol{x} \leqq 0 & \text{for } i \in I^z(\boldsymbol{w}) \\ & \boldsymbol{e}^j\boldsymbol{x} \geqq c_j & \text{for } j \in I^c(\boldsymbol{w}), \end{array}\right.$$

where

$$I^z(\boldsymbol{w}) = \{\, j \in \{1, \ldots, n\} \mid w_j = 0 \,\}, \quad \text{and}$$
$$I^c(\boldsymbol{w}) = \{\, j \in \{1, \ldots, n\} \mid w_j = c_j \,\}.$$

We apply the dual simplex method to solve the problem and obtain the matrices B and N to be used in Step $\langle k \rangle$. See Appendix B for the dual simplex method.

5.3 Cut-and-Split Method for (mmF)

In this section we propose an algorithm for (mmF), which is based on the CS method. Note that we can directly apply the CS method to (mmF) since it satisfies the assumptions of (2.6), i.e.,

$$X \subsetneqq \mathbb{R}^n_+ \quad \text{and} \quad \mathbf{0} \in X_V \cap \operatorname{int} \bar{G}. \tag{5.6}$$

To make the algorithm more efficient we combine the LS($\boldsymbol{w}$) procedure with the CS method, namely we apply the LS($\boldsymbol{w}$) procedure to obtain a tighter upper bound every time we find a feasible solution $\boldsymbol{w} \in X_M$.

5.3.1 Algorithm

The CS method for (mmF) is described as follows.

/** CS method for (mmF) **/

$\langle 0 \rangle$ (initialization) Find an initial feasible vertex $\boldsymbol{w}^0 \in X_M \cap X_V$ of (mmF). If $N_M(\boldsymbol{w}^0) = \emptyset$ then stop ($\boldsymbol{w}^0$ is the unique optimal solution of (mmF)). Otherwise, apply the LS($\boldsymbol{w}^0$) procedure to obtain a locally optimal vertex $\bar{\boldsymbol{x}} \in X_M \cap X_V$. Set $K_0 := \mathbb{R}^n_+$, $\mathcal{S} := \{K_0\}$, $\mathcal{R} := \mathcal{S}$, $\gamma := 0$ and $k := 0$.

$\langle k \rangle$ (iteration k) For each $K \in \mathcal{S}$, solve $\beta_K := \min\{\, \boldsymbol{d}\boldsymbol{x} \mid \boldsymbol{x} \in X \cap K,\ l_K(\boldsymbol{x}) \geqq 0 \,\}$ to obtain a solution $\boldsymbol{\omega}^K$ if the problem is feasible, set $\beta_k := \infty$ otherwise, where $l_K(\boldsymbol{x}) \geqq 0$ is the concavity cut for $K \setminus \bar{G}$.

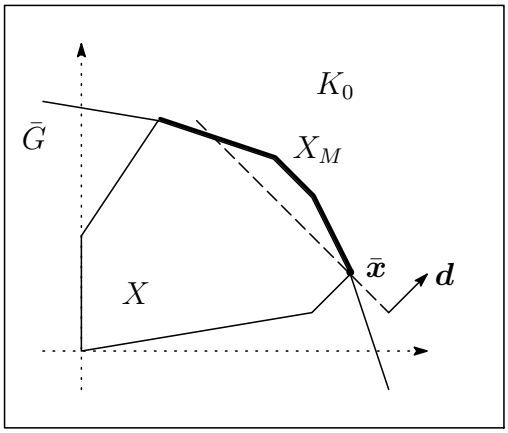
$\langle k1 \rangle$ (update) Set $\mathcal{L} := \{\, K \in \mathcal{S} \mid \boldsymbol{\omega}^K \in X_M \,\}$. If $\mathcal{L} \neq \emptyset$ then apply the LS($\boldsymbol{\omega}^K$) procedure to obtain a locally optimal vertex $\boldsymbol{v}^K$ for each $K \in \mathcal{L}$. Solve $\min\{\, \boldsymbol{d}\boldsymbol{v}^K \mid K \in \mathcal{L} \,\}$ to obtain the cone K^*. If $\boldsymbol{d}\boldsymbol{v}^{K^*} < \boldsymbol{d}\bar{\boldsymbol{x}}$, set $\bar{\boldsymbol{x}} := \boldsymbol{v}^{K^*}$.

$\langle k2 \rangle$ (termination) Set $\mathcal{R}' := \{\, K \in \mathcal{R} \mid \beta_K < \boldsymbol{d}\bar{\boldsymbol{x}} \,\}$. If $\mathcal{R}' = \emptyset$ or $\boldsymbol{d}\bar{\boldsymbol{x}} - \gamma < 1$ then stop ($\bar{\boldsymbol{x}}$ solves (mmF)).

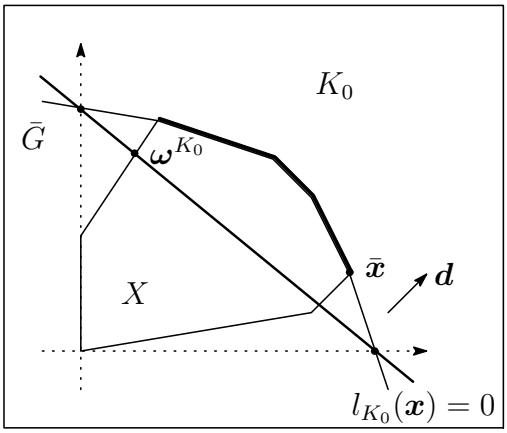
$\langle k3 \rangle$ (subdivision) Solve $\min\{\, \beta_K \mid K \in \mathcal{R}' \,\}$ to obtain the cone K^{**}. If $\beta_{K^{**}} > \gamma$ then set $\gamma := \beta_{K^{**}}$. Perform the $\boldsymbol{\omega}$-subdivision on K^{**} for some $\boldsymbol{\omega} \in K^{**}$,

and let $\mathcal{S}^{**}$ be the partition of K^{**}. Set $\mathcal{S} := \mathcal{S}^{**}$, $\mathcal{R} := \mathcal{S}^{**} \cup (\mathcal{R}' \setminus \{K^{**}\})$, $k := k + 1$ and go to $\langle k \rangle$.

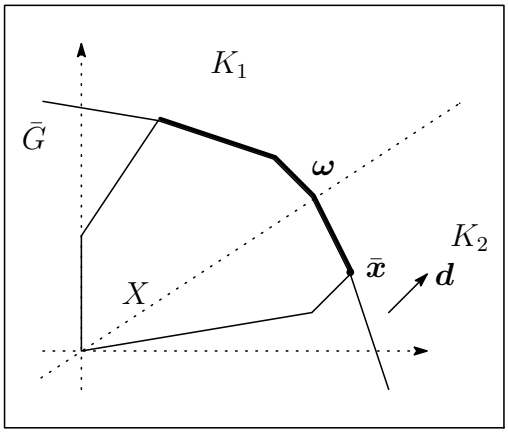
We illustrate the CS method for (mmF) in Figure 5.3, in which we use a two-dimensional general polyhedron $X = \{ \boldsymbol{x} \in \mathbb{R}^2 \mid A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0} \}$ instead of $X = \{ \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c} \}$ since the latter is unsuitable for illustration. We first obtain a locally optimal vertex $\bar{\boldsymbol{x}} \in X_M \cap X_V$ and set $K_0 := \mathbb{R}^n_+$ (See (a)). We determine the concavity cut $l_{K_0}(\boldsymbol{x}) \geqq 0$ for $K_0 \setminus \bar{G}$ and obtain a point $\boldsymbol{\omega}^{K_0}$ (See (b)). The value $\boldsymbol{d\omega}^{K_0}$ is a lower bound with respect to K_0. Since $\boldsymbol{\omega}^{K_0} \notin X_M$ and $\boldsymbol{d\omega}^{K_0} < \boldsymbol{d\bar{x}}$, we split the cone K_0 into K_1 and K_2 (See (c)). In the next iteration, points $\boldsymbol{\omega}^{K_1}$ and $\boldsymbol{\omega}^{K_2}$ are obtained (See (d)). We see that $\boldsymbol{\omega}^{K_2} \in X_M$, and hence apply the LS($\boldsymbol{\omega}^{K_2}$) procedure to obtain a better point $\boldsymbol{v}^{K_2}$ and update the incumbent $\bar{\boldsymbol{x}}$ to $\boldsymbol{v}^{K_2}$. The cone K_2 is discarded, since $\boldsymbol{d\omega}^{K_2} \geqq \boldsymbol{d\bar{x}}$. Meanwhile K_1 is split into K_3 and K_4 since $\boldsymbol{\omega}^{K_1} \notin X_M$ and $\boldsymbol{d\omega}^{K_1} < \boldsymbol{d\bar{x}}$ (See (e)). We obtain points $\boldsymbol{\omega}^{K_3}$ and $\boldsymbol{\omega}^{K_4}$ in the next iteration (See (f)) and continue the algorithm.



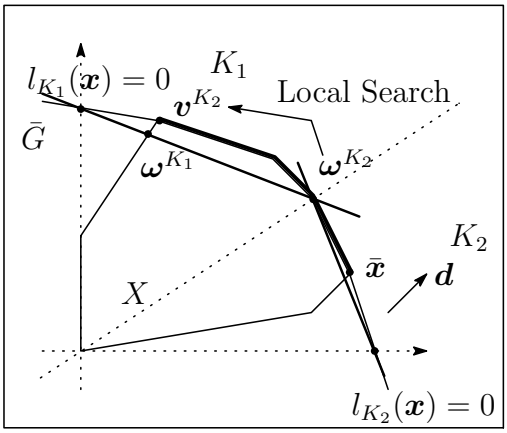
(a) initialization



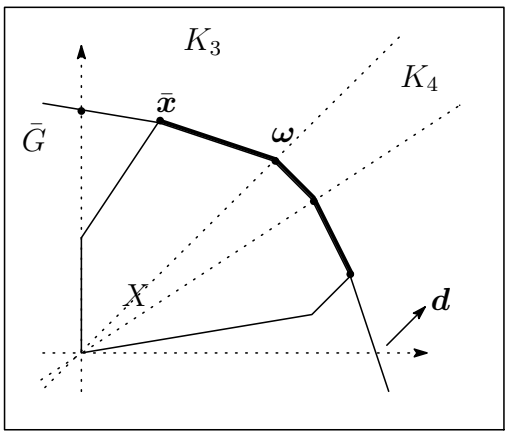
(b) obtaining $\boldsymbol{\omega}^{K_0}$



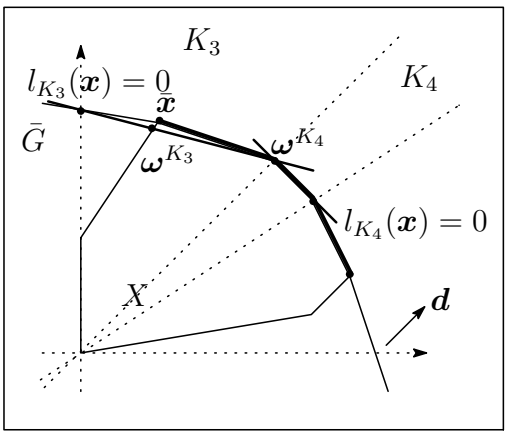
(c) splitting K_0 into K_1 and K_2



(d) obtaining $\boldsymbol{\omega}^{K_1}$ and $\boldsymbol{\omega}^{K_2}$



(e) splitting K_1 into K_3 and K_4



(f) obtaining $\boldsymbol{\omega}^{K_3}$ and $\boldsymbol{\omega}^{K_4}$

Figure 5.3: CS method for (mmF)

5.3.2 The way of deciding $\boldsymbol{\omega}$ for subdivision and convergence of the algorithm

In Step $\langle k3 \rangle$ the direction $\boldsymbol{\omega} \in K^{**}$ is obtained as follows. Suppose that K^{**} is generated by the directions $\boldsymbol{u}^1, \ldots, \boldsymbol{u}^n$ such that $\boldsymbol{u}^i \in \partial \bar{G}$, initially $\boldsymbol{u}^i = \alpha^* \boldsymbol{e}^i$ with $\alpha^* = \max\{\, \alpha \mid \bar{g}(\alpha \boldsymbol{e}^i) \geqq 0 \,\}$ for each $i = 1, \ldots, n$. The function $l_{K^{**}} : \mathbb{R}^n \to \mathbb{R}$ defining the concavity cut $l_{K^{**}}(\boldsymbol{x}) \geqq 0$ for $K^{**} \setminus \bar{G}$ is given by $l_{K^{**}}(\boldsymbol{x}) = \boldsymbol{e}U^{-1}\boldsymbol{x} - 1$, where $U = \begin{bmatrix} \boldsymbol{u}^1 & \cdots & \boldsymbol{u}^n \end{bmatrix}$. Here we solve

$$\eta = \max\{\, l_{K^{**}}(\boldsymbol{y}) \mid \boldsymbol{y} \in X \cap K^{**} \,\}, \tag{5.7}$$

to obtain a solution $\boldsymbol{y}^*$. Since K^{**} is in $\mathcal{R}'$, i.e., $\beta_{K^{**}} < \boldsymbol{d}\bar{\boldsymbol{x}}$, we see that $\eta \geqq 0$. If $\eta > 0$ then we perform $\boldsymbol{\omega}$-subdivision on K^{**} with $\boldsymbol{\omega} = \boldsymbol{y}^*$. If $\eta = 0$ then we discard K^{**} from $\mathcal{R}'$ and go back to Step $\langle k2 \rangle$. In this case there is no point $\boldsymbol{v} \in X_V \cap K^{**}$ such that $\boldsymbol{v} \neq \boldsymbol{u}^i$ for each $i = 1, \ldots, n$. Then we can discard K^{**} from further consideration, because at least one vertex of X solves (mmF).

Every time we obtain an optimal solution $\boldsymbol{y}^*$ of (5.7) with $\eta > 0$, we can perform $\boldsymbol{y}^*$-subdivision on K^{**}. This assertion follows from the following theorem.

Theorem 5.5 *Let K^{**} be a cone generated by the directions $\boldsymbol{u}^1, \ldots, \boldsymbol{u}^n$ such that $\boldsymbol{u}^i \in \partial \bar{G}$ for each $i = 1, \ldots, n$, and $\boldsymbol{y}^*$ be an optimal solution of (5.7) with $\eta > 0$. Then $\boldsymbol{y}^* \neq \alpha \boldsymbol{u}^i$ for any $i = 1, \ldots, n$ and for any $\alpha > 0$.*

Proof: Assume that $\boldsymbol{y}^*$ lies on an extreme ray of K^{**}, i.e., $\boldsymbol{y}^* = \alpha \boldsymbol{u}^j$ for some $\alpha > 0$ and $\boldsymbol{u}^j$. Since $0 < \eta = l_{K^{**}}(\boldsymbol{y}^*) = \boldsymbol{e}U^{-1}(\alpha \boldsymbol{u}^j) - 1 = \alpha - 1$, we have $\alpha > 1$. By the choice of $\boldsymbol{u}^j$, we have $\bar{g}((1+\varepsilon)\boldsymbol{u}^j) < 0$ for any $\varepsilon > 0$. Therefore we have $\bar{g}(\boldsymbol{y}^*) < 0$. On the other hand, $\boldsymbol{y}^* \in X \cap K^{**} \subseteqq K$, which implies $\bar{g}(\boldsymbol{y}^*) \geqq 0$. This is a contradiction. □

Furthermore the following assertion is also available in this subdivision rule.

Theorem 5.6 *Let K be the cone generated by the directions $\boldsymbol{u}^1, \ldots, \boldsymbol{u}^n$ and $U = \begin{bmatrix} \boldsymbol{u}^1 & \cdots & \boldsymbol{u}^n \end{bmatrix}$. If $\boldsymbol{u}^i \in X_M$ for each $i = 1, \ldots, n$ then some $\boldsymbol{u}^j$ solves $\min\{\, \boldsymbol{d}\boldsymbol{x} \mid \boldsymbol{x} \in X \cap K,\ l_K(\boldsymbol{x}) \geqq 0 \,\}$.*

Proof: Let $\boldsymbol{\omega}$ be an optimal solution of $\min\{\, \boldsymbol{d}\boldsymbol{x} \mid \boldsymbol{x} \in X \cap K,\ l_K(\boldsymbol{x}) \geqq 0 \,\}$. Since $\boldsymbol{\omega} \in K$, there are nonnegative numbers $\mu_1, \ldots, \mu_n$ such that $\boldsymbol{\omega} = \sum_{i=1}^n \mu_i \boldsymbol{u}^i$. Also

we see that $0 \leqq l_K(\boldsymbol{\omega}) = \boldsymbol{e}U^{-1}(\sum_{i=1}^n \mu_i \boldsymbol{u}^i) - 1 = \sum_{i=1}^n \mu_i - 1$, and hence $\sum_{i=1}^n \mu_i \geqq 1$. Note that $\boldsymbol{dx} \geqq 0$ for all $\boldsymbol{x} \in X$ by the assumption that a given network has no t-s-path. Let $\boldsymbol{u}^j$ attain $\min\{\, \boldsymbol{du}^i \mid i = 1, \ldots, n \,\}$. Then $\boldsymbol{du}^j \leqq \sum_{i=1}^n \mu_i \boldsymbol{du}^i = \boldsymbol{d\omega}$, in other words, $\boldsymbol{u}^j \in X_M$ solves $\min\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in X \cap K,\ l_K(\boldsymbol{x}) \geqq 0 \,\}$. □

We see that the set $\{\, \boldsymbol{x} \in K^{**} \mid l_{K^{**}}(\boldsymbol{x}) > 0 \,\}$ does not contain a vertex of X when η of (5.7) is zero. Suppose that an oracle is available that provides a vertex of X in $\{\, \boldsymbol{x} \in K^{**} \mid l_{K^{**}}(\boldsymbol{x}) > 0 \,\}$ whenever there are some, and take the vertex as the direction $\boldsymbol{\omega}$ in Step $\langle k3 \rangle$. Then, owing to the finiteness of X_V, the $\boldsymbol{\omega}$-subdivision is repeated at most $|X_V|$ times, and hence the CS method terminates after finitely many iterations. However, the oracle is costly and the authors estimate it NP-complete to check whether $X_V \cap \{\, \boldsymbol{x} \in K^{**} \mid l_{K^{**}}(\boldsymbol{x}) > 0 \,\}$ is not empty. See Freund-Orlin [20].

5.3.3 Implementation issue about the CS Method for (mmF)

In the CS method for (mmF) we need to obtain the point $\boldsymbol{u} \in \{\, \alpha\boldsymbol{\omega} \mid \alpha > 0 \,\} \cap \partial\bar{G}$ for a direction $\boldsymbol{\omega}$, by applying the parametric simplex method. The detail will be explained in Subsection 5.4.3.

Let K be the cone given by

$$K = \{\, U\boldsymbol{\mu} \mid \boldsymbol{\mu} \geqq \boldsymbol{0} \,\},$$

where the jth column $\boldsymbol{u}^j$ of U is contained in $\partial\bar{G}$. To check if the set $\{\, \boldsymbol{x} \in X \cap K \mid l_K(\boldsymbol{x}) \geqq 0 \,\}$ is empty, we solve

$$\eta_K = \max\{\, l_K(\boldsymbol{x}) \mid \boldsymbol{x} \in X \cap K \,\},$$

and we see that the set is empty if $\eta_K < 0$. This problem is

$$\left|\begin{array}{lll} \eta_K = & \max\limits_{(\boldsymbol{x},\boldsymbol{\mu})} & \boldsymbol{e}U^{-1}\boldsymbol{x} - 1 \\ & \text{s.t.} & A\boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c} \\ & & \boldsymbol{x} = U\boldsymbol{\mu},\ \boldsymbol{\mu} \geqq \boldsymbol{0}, \end{array}\right.$$

which is equivalent to

$$\left|\begin{array}{lll} \eta_K = & \max\limits_{\boldsymbol{\mu}} & \boldsymbol{e\mu} - 1 \\ & \text{s.t.} & AU\boldsymbol{\mu} = \mathbf{0},\ \mathbf{0} \leqq U\boldsymbol{\mu} \leqq \boldsymbol{c} \\ & & \boldsymbol{\mu} \geqq \mathbf{0}. \end{array}\right.$$

To obtain a locally optimal vertex $\boldsymbol{v}^K$, we first solve

$$\left|\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{dx} \\ \text{s.t.} & \boldsymbol{x} \in X \cap K, \end{array}\right.$$

and we then solve

$$\left|\begin{array}{lll} \beta_K = & \min\limits_{\boldsymbol{x}} & \boldsymbol{dx} \\ & \text{s.t.} & \boldsymbol{x} \in X \cap K,\ l_K(\boldsymbol{x}) \geqq 0, \end{array}\right.$$

by the dual simplex method, which will be explained in Appendix. Since the above problem is equivalent to

$$\left|\begin{array}{lll} \beta_K = & \min\limits_{\boldsymbol{\mu}} & \boldsymbol{d}U\boldsymbol{\mu} \\ & \text{s.t.} & AU\boldsymbol{\mu} = \mathbf{0},\ \mathbf{0} \leqq U\boldsymbol{\mu} \leqq \boldsymbol{c} \\ & & \boldsymbol{e\mu} \geqq 1,\ \boldsymbol{\mu} \geqq \mathbf{0}, \end{array}\right.$$

we need not calculate the inverse matrix U^{-1}.

If the set $X \cap K \cap \{\, \boldsymbol{x} \mid l_K(\boldsymbol{x}) \geqq 0 \,\}$ contains either no integer solution or only one integer solution, then we stop dividing the cone K. For each $i = 1, \ldots, n$, let

$$\begin{aligned} UB(K,i) &= \max\{u_i^1, \cdots, u_i^n, \eta_K u_i^1, \ldots, \eta_K u_i^n\}, \text{ and} \\ LB(K,i) &= \min\{u_i^1, \cdots, u_i^n, \eta_K u_i^1, \ldots, \eta_K u_i^n\}. \end{aligned}$$

It is readily seen that

$$\begin{aligned} X \cap K \cap \{\, \boldsymbol{x} \mid l_K(\boldsymbol{x}) \geqq 0 \,\} &\subseteqq \text{conv}\{\boldsymbol{u}^1, \cdots, \boldsymbol{u}^n, \eta_K \boldsymbol{u}^1, \cdots, \eta_K \boldsymbol{u}^n\} \\ &\subseteqq \Pi_{i=1}^n [LB(K,i), UB(K,i)], \end{aligned}$$

where conv$S$ means the convex full of a set S. If there is $i \in \{1, \ldots, n\}$ such that $\lfloor UB(K,i) \rfloor < \lceil LB(K,i) \rceil$ then the hypercube $\Pi_{i=1}^n [LB(K,i), UB(K,i)]$ contains no integer solution. Also if $\lfloor UB(K,i) \rfloor = \lceil LB(K,i) \rceil$ for all $i \in \{1, \ldots, n\}$ then the above hypercube contains only one integer solution whose ith element is $\lfloor UB(K,i) \rfloor$.

5.4 Outer Approximation Method for (mmF)

In this section we propose an algorithm for (mmF), which is based on the OA method. Obviously, X of (4.1) is bounded, and by Assumption 5.1 (ii)-(iii) we have

$$\mathbf{0} \in X \cap \operatorname{int} \bar{G}, \text{ and } \min\{\, \boldsymbol{dx} \mid \boldsymbol{x} \in X \,\} = 0, \tag{5.8}$$

which correspond to (2.9). Then we can apply the OA method to (mmF) if the regularity condition is met.

5.4.1 Regularity and optimality condition

Unfortunately, the problem (mmF) is not regular, i.e.,

$$X \setminus \operatorname{int} \bar{G} = \operatorname{cl}(X \setminus \bar{G}) \tag{5.9}$$

does not hold. To work out countermeasures we introduce a positive tolerance ε and consider, instead of (mmF),

$$(mmF_\varepsilon) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{dx} \\ \text{s.t.} & \boldsymbol{x} \in X \setminus \operatorname{int} \bar{G}_\varepsilon, \end{array} \right.$$

where

$$\bar{G}_\varepsilon = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \bar{g}(\boldsymbol{x}) \geqq \varepsilon \,\}. \tag{5.10}$$

We call an optimal solution of (mmF_ε) an *ε-optimal solution of* (mmF).

First we show that any positive ε ensures the regularity of (mmF_ε).

Theorem 5.7 *The problem (mmF_ε) is regular for any $\varepsilon > 0$.*

Proof: We show that

$$X \setminus \operatorname{int} \bar{G}_\varepsilon = \operatorname{cl}(X \setminus \bar{G}_\varepsilon) \tag{5.11}$$

holds for any $\varepsilon > 0$.

($\supseteqq$) Since $X \setminus \operatorname{int} \bar{G}_\varepsilon$ is closed and $X \setminus \operatorname{int} \bar{G}_\varepsilon \supseteqq X \setminus \bar{G}_\varepsilon$, we have

$$X \setminus \operatorname{int} \bar{G}_\varepsilon = \operatorname{cl}(X \setminus \operatorname{int} \bar{G}_\varepsilon) \supseteqq \operatorname{cl}(X \setminus \bar{G}_\varepsilon).$$

($\subseteqq$) Let $\boldsymbol{x}$ be an arbitrary point of $X \setminus \operatorname{int} \bar{G}_\varepsilon$. We show that there is always a point, say x' in $N_\delta(\boldsymbol{x}) \cap (X \setminus \bar{G}_\varepsilon)$. If $\bar{g}(\boldsymbol{x}) > \varepsilon$ then there exists $\gamma > 0$ such that

$\bar{g}(\boldsymbol{x}') > \varepsilon$ for any point $\boldsymbol{x}' \in N_\gamma(\boldsymbol{x})$ by the continuity of $\bar{g}$. This implies $N_\gamma(\boldsymbol{x}) \subseteqq \bar{G}_\varepsilon$, and hence $\boldsymbol{x} \in \operatorname{int} \bar{G}_\varepsilon$. Therefore the assumption $\boldsymbol{x} \in X \setminus \operatorname{int} \bar{G}_\varepsilon$ implies that $\boldsymbol{x} \in X$ and $\bar{g}(\boldsymbol{x}) \leqq \varepsilon$. By Theorem 5.3, we have $\bar{g}(\boldsymbol{x}) = g(\boldsymbol{x})$. When $\bar{g}(\boldsymbol{x}) < \varepsilon$, take $\boldsymbol{x}$ as $\boldsymbol{x}'$. Clearly $\boldsymbol{x}' = \boldsymbol{x} \notin \bar{G}_\varepsilon$ and $\boldsymbol{x}' = \boldsymbol{x} \in N_\delta(\boldsymbol{x})$, and we have done. When $g(\boldsymbol{x}) = \bar{g}(\boldsymbol{x}) = \varepsilon$, there is an optimal solution $\boldsymbol{y}^*$ of $\max\{\, \boldsymbol{e}\boldsymbol{y} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} \geqq \boldsymbol{x} \,\}$ such that $\boldsymbol{e}(\boldsymbol{y}^* - \boldsymbol{x}) = \varepsilon$, and hence $\boldsymbol{y}^* \neq \boldsymbol{x}$. Take λ such that $0 < \lambda < \min\{\, 1,\ \delta / \|\boldsymbol{y}^* - \boldsymbol{x}\| \,\}$ and let $\boldsymbol{x}' = \lambda \boldsymbol{y}^* + (1 - \lambda)\boldsymbol{x}$. Since $\|\boldsymbol{x}' - \boldsymbol{x}\| = \lambda \|\boldsymbol{y}^* - \boldsymbol{x}\| < \delta$, we see $\boldsymbol{x}' \in N_\delta(\boldsymbol{x})$. Also we see that $\boldsymbol{x}' \in X$ by the convexity of X, and hence $g(\boldsymbol{x}') = \bar{g}(\boldsymbol{x}')$ by applying Theorem 5.3 again. Since $\boldsymbol{x}' \geqq \boldsymbol{x}$ and $\boldsymbol{x}' \neq \boldsymbol{x}$, we have

$$\begin{aligned}
\bar{g}(\boldsymbol{x}') &= g(\boldsymbol{x}') \\
&= \max\{\, \boldsymbol{e}\boldsymbol{y} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} \geqq \boldsymbol{x}' \,\} - \boldsymbol{e}\boldsymbol{x}' \\
&< \max\{\, \boldsymbol{e}\boldsymbol{y} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} \geqq \boldsymbol{x} \,\} - \boldsymbol{e}\boldsymbol{x} \\
&= \boldsymbol{e}(\boldsymbol{y}^* - \boldsymbol{x}) = \varepsilon.
\end{aligned}$$

Therefore we see that $\boldsymbol{x}' \notin \bar{G}_\varepsilon$. This completes the proof. □

We illustrate a difference between (mmF) and (mmF_ε) in Figure 5.4, in which we see that $X \setminus \operatorname{int} \bar{G}_\varepsilon = \operatorname{cl}(X \setminus \bar{G}_\varepsilon)$ while $X \setminus \operatorname{int} \bar{G} \neq \operatorname{cl}(X \setminus \bar{G})$.

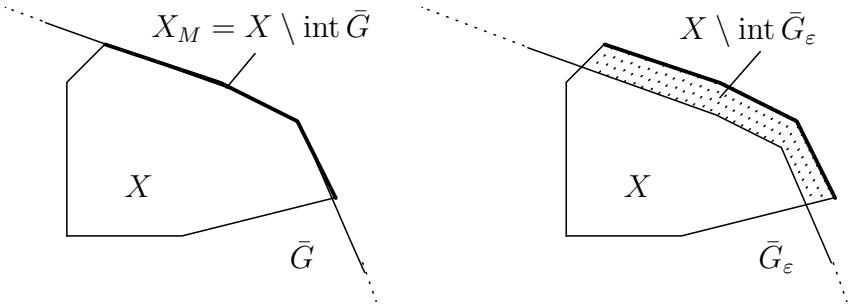


Figure 5.4: A difference between (mmF) and (mmF_ε)

Next we discuss an upper bound of ε, which will be crucial for the convergence of the algorithm.

Lemma 5.8 *If* $\varepsilon \in (0, 1)$ *then* $\mathbf{0} \in \operatorname{int} \bar{G}_\varepsilon$, *and* $(\mathbf{0}, \boldsymbol{v}) \cap \partial \bar{G}_\varepsilon \neq \emptyset$ *for any point* $\boldsymbol{v}$ *such that* $\bar{g}(\boldsymbol{v}) \leqq 0$.

Proof: We have $\bar{g}(\mathbf{0}) > 0$ since $\mathbf{0} \in \operatorname{int} \bar{G}$. Note that $\bar{g}(\mathbf{0})$, which coincides with $g(\mathbf{0})$, takes an integer value by the integrality property of X, and hence $\bar{g}(\mathbf{0}) \geqq 1$. Then we have $\bar{g}(\mathbf{0}) > \varepsilon$, i.e., $\mathbf{0} \in \operatorname{int} \bar{G}_\varepsilon$ for any $\varepsilon \in (0, 1)$. The continuity of $\bar{g}$ ensures the last assertion. □

In the following lemma, we use δ_s to denote the number of arcs leaving node s, i.e.,

$$\delta_s = |\{\, h \in E \mid d_h = +1 \,\}|. \tag{5.12}$$

Lemma 5.9 *Let $\boldsymbol{x}^*$ and $\boldsymbol{x}^*_\varepsilon$ be an optimal solution and an ε-optimal solution of (mmF), respectively. Then $0 \leqq \boldsymbol{d}\boldsymbol{x}^* - \boldsymbol{d}\boldsymbol{x}^*_\varepsilon \leqq \varepsilon\delta_s$.*

Proof: Since $\boldsymbol{x}^* \in X$ and $\bar{g}(\boldsymbol{x}^*) \leqq 0$, $\boldsymbol{x}^*$ is a feasible solution of (mmF_ε), and hence $\boldsymbol{d}\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}\boldsymbol{x}^*$. Let $\boldsymbol{y}^*_\varepsilon$ be an optimal solution of $\max\{\, \boldsymbol{e}\boldsymbol{y} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} \geqq \boldsymbol{x}^*_\varepsilon \,\}$. Clearly $\boldsymbol{y}^*_\varepsilon \in X_M$, i.e., $\boldsymbol{y}^*_\varepsilon$ is a feasible solution of (mmF), and hence $\boldsymbol{d}\boldsymbol{x}^* \leqq \boldsymbol{d}\boldsymbol{y}^*_\varepsilon$. We see that $(y^*_\varepsilon)_h - (x^*_\varepsilon)_h \leqq \varepsilon$ for each $h = 1, \ldots, n$, since $\boldsymbol{y}^*_\varepsilon - \boldsymbol{x}^*_\varepsilon \geqq \mathbf{0}$ and $\boldsymbol{e}(\boldsymbol{y}^*_\varepsilon - \boldsymbol{x}^*_\varepsilon) \leqq \varepsilon$. That implies $\boldsymbol{d}(\boldsymbol{y}^*_\varepsilon - \boldsymbol{x}^*_\varepsilon) \leqq \varepsilon|\{\, h \mid d_h = +1 \,\}| = \varepsilon\delta_s$, hence $\boldsymbol{d}\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}\boldsymbol{x}^* \leqq \boldsymbol{d}\boldsymbol{y}^*_\varepsilon \leqq \boldsymbol{d}\boldsymbol{x}^*_\varepsilon + \varepsilon\delta_s$. □

Theorem 5.10 *Let $\boldsymbol{x}^*_\varepsilon$ be an ε-optimal solution for some $\varepsilon \in (0,\, 1/\delta_s)$. Then $\lceil \boldsymbol{d}\boldsymbol{x}^*_\varepsilon \rceil$ coincides with the optimal value of (mmF).*

Proof: From Lemma 5.9 we see that $0 \leqq \boldsymbol{d}\boldsymbol{x}^* - \boldsymbol{d}\boldsymbol{x}^*_\varepsilon < 1$. This inequality and the integrality of $\boldsymbol{d}\boldsymbol{x}^*$ gives the assertion. □

In the sequel we choose ε from the open interval $(0,\, 1/\delta_s)$.

Note that $\bar{g}(\boldsymbol{x}^*_\varepsilon) \leqq \varepsilon$ holds for an ε-optimal solution $\boldsymbol{x}^*_\varepsilon$ of (mmF). Therefore $\bar{g}(\boldsymbol{x}) \leqq 0$ for any accumulation point $\boldsymbol{x}$ of $\{\boldsymbol{x}^*_\varepsilon\}_{\varepsilon \to 0+}$. This observation leads to the following corollary.

Corollary 5.11 *Let $\{\boldsymbol{x}^*_\varepsilon\}_{\varepsilon \to 0+}$ be a sequence of ε-optimal solutions of (mmF) for ε converging to 0 from above. Then any accumulation point of $\{\boldsymbol{x}^*_\varepsilon\}_{\varepsilon \to 0+}$ is an optimal solution of (mmF).*

For $\eta \in \mathbb{R}$ let

$$X(\eta) = \{\, \boldsymbol{x} \in X \mid \boldsymbol{d}\boldsymbol{x} \leqq \eta \,\}. \tag{5.13}$$

As seen in Theorem 2.5, the optimality condition of (mmF_ε) is $X(\boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon) \subseteqq \bar{G}_\varepsilon$ for some $\bar{\boldsymbol{x}}_\varepsilon \in X \setminus \operatorname{int} \bar{G}_\varepsilon$. We can further relax this condition.

Theorem 5.12 *Let $\bar{\boldsymbol{x}}_\varepsilon \in X \setminus int \bar{G}_\varepsilon$ for some $\varepsilon \in (0,\, 1/\delta_s)$. If $X(\lceil \boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon - 1 \rceil) \subseteqq \bar{G}_{\varepsilon'}$ for some $\varepsilon' > 0$ then $\lceil \boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon \rceil$ coincides with the optimal value of (mmF).*

Proof: Let $\boldsymbol{x}^*$ and $\boldsymbol{x}^*_\varepsilon$ be an optimal solution and an ε-optimal solution of (mmF), respectively. Since $\bar{\boldsymbol{x}}_\varepsilon$ is a feasible solution of (mmF_ε), we have $\boldsymbol{d}\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon$. It is also clear that $\boldsymbol{d}\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}\boldsymbol{x}^*$. If $\boldsymbol{d}\boldsymbol{x}^* < \boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon$ then we have $\boldsymbol{x}^* \in X(\lceil \boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon - 1 \rceil) \subseteqq \bar{G}_{\varepsilon'}$ since $\boldsymbol{d}\boldsymbol{x}^*$ is integer, and hence $\bar{g}(\boldsymbol{x}^*) \geqq \varepsilon' > 0$, which contradicts that $\bar{g}(\boldsymbol{x}^*) = 0$. Then we have $\boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon \leqq \boldsymbol{d}\boldsymbol{x}^*$. Hence by Lemma 5.9 we obtain $\boldsymbol{d}\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon \leqq \boldsymbol{d}\boldsymbol{x}^* \leqq \boldsymbol{d}\boldsymbol{x}^*_\varepsilon + \varepsilon\delta_s < \boldsymbol{d}\boldsymbol{x}^*_\varepsilon + 1$. This completes the proof. □

We construct a polytope P satisfying $X(\lceil \boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon - 1 \rceil) \subseteqq P$ for some $\bar{\boldsymbol{x}}_\varepsilon \in X \setminus \operatorname{int} \bar{G}_\varepsilon$. Let $\boldsymbol{v}^*$ be a vertex minimizing $\bar{g}(\boldsymbol{v})$ over P_V and $\varepsilon' = \bar{g}(\boldsymbol{v}^*)$. For any $\boldsymbol{x} \in P$ we have $\bar{g}(\boldsymbol{x}) \geqq \bar{g}(\boldsymbol{v}^*)$, i.e., $0 \leqq \bar{g}(\boldsymbol{x}) - \bar{g}(\boldsymbol{v}^*) = \bar{g}(\boldsymbol{x}) - \varepsilon'$, and hence $P \subseteqq \bar{G}_{\varepsilon'}$. This implies that $X(\lceil \boldsymbol{d}\bar{\boldsymbol{x}}_\varepsilon - 1 \rceil) \subseteqq \bar{G}_{\varepsilon'}$. Therefore if $\varepsilon' > 0$ then the optimal value of (mmF) is obtained by Theorem 5.12.

5.4.2 Algorithm and its convergence

We describe the OA method for (mmF) as follows.

/** OA method for (mmF) **/

⟨0⟩ (initialization) Find an initial feasible vertex $\boldsymbol{w}^0 \in X_M \cap X_V$ of (mmF). If $N_M(\boldsymbol{w}^0) = \emptyset$ then stop ($\boldsymbol{w}^0$ is the unique feasible solution of (mmF)). Otherwise, apply the LS($\boldsymbol{w}^0$) procedure to obtain a locally optimal vertex $\bar{\boldsymbol{x}}^0 \in X_M \cap X_V$. Solve $\zeta := \max\{\, \boldsymbol{e}\boldsymbol{x} \mid \boldsymbol{x} \in X,\ \boldsymbol{d}\boldsymbol{x} \leqq \boldsymbol{d}\bar{\boldsymbol{x}}^0 - 1 \,\}$ and construct an initial polytope $P^0 \supseteqq X(\boldsymbol{d}\bar{\boldsymbol{x}}^0 - 1)$ by setting $P^0 := \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{e}\boldsymbol{x} \leqq \zeta,\ \boldsymbol{d}\boldsymbol{x} \leqq \boldsymbol{d}\bar{\boldsymbol{x}}^0 - 1,\ \boldsymbol{x} \geqq \boldsymbol{0} \,\}$. Compute the vertex set P^0_V of P^0. Set $k := 0$.

⟨k⟩ (iteration k) Find a vertex $\boldsymbol{v}^k \in P^k_V$ such that $\bar{g}(\boldsymbol{v}^k) \leqq 0$ if any.

$\langle k1 \rangle$ (termination) If either $\boldsymbol{d}\bar{\boldsymbol{x}}^k = 0$ or $\bar{g}(\boldsymbol{v}) > 0$ for all $\boldsymbol{v} \in P_V^k$ then stop (the optimal value of (mmF) is $\lceil \boldsymbol{d}\bar{\boldsymbol{x}}^k \rceil$). Otherwise, obtain the point $\boldsymbol{x}_\varepsilon^k \in (\boldsymbol{0}, \boldsymbol{v}^k) \cap \partial \bar{G}_\varepsilon$. (Note that Lemma 5.8 ensures that $(\boldsymbol{0}, \boldsymbol{v}^k) \cap \partial \bar{G}_\varepsilon \neq \emptyset$).

$\langle k2 \rangle$ (update) If $\boldsymbol{x}_\varepsilon^k \in X$, obtain the point $\boldsymbol{x}^k \in (\boldsymbol{0}, \boldsymbol{v}^k] \cap \partial \bar{G}$.

$\langle k2.1 \rangle$ If $\boldsymbol{x}^k \in X$, meaning $\boldsymbol{x}^k \in X_M$, then obtain the locally optimal vertex $\boldsymbol{z}^k \in X_M \cap X_V$ by applying the LS($\boldsymbol{x}^k$) procedure, and further obtain the point $\boldsymbol{z}_\varepsilon^k \in (\boldsymbol{0}, \boldsymbol{z}^k) \cap \partial \bar{G}_\varepsilon$. Set $\bar{\boldsymbol{x}}^{k+1} :=$ argmin$\{ \boldsymbol{d}\boldsymbol{z}_\varepsilon^k,\ \boldsymbol{d}\boldsymbol{x}_\varepsilon^k \}$ and $P^{k+1} := P^k \cap \{ \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{d}\boldsymbol{x} \leqq \lceil \boldsymbol{d}\bar{\boldsymbol{x}}^{k+1} - 1 \rceil \}$.

$\langle k2.2 \rangle$ If $\boldsymbol{x}^k \notin X$, meaning $\boldsymbol{x}^k \notin X_M$, then set $\bar{\boldsymbol{x}}^{k+1} := \boldsymbol{x}_\varepsilon^k$ and $P^{k+1} := P^k \cap \{ \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{d}\boldsymbol{x} \leqq \lceil \boldsymbol{d}\bar{\boldsymbol{x}}^{k+1} - 1 \rceil,\ l(\boldsymbol{x}) \leqq 0 \}$, with some affine function $l : \mathbb{R}^n \to \mathbb{R}$.

$\langle k3 \rangle$ If $\boldsymbol{x}_\varepsilon^k \notin X$ then set $\bar{\boldsymbol{x}}^{k+1} := \bar{\boldsymbol{x}}^k$ and $P^{k+1} := P^k \cap \{ \boldsymbol{x} \in \mathbb{R}^n \mid l(\boldsymbol{x}) \leqq 0 \}$, with some affine function $l : \mathbb{R}^n \to \mathbb{R}$.

$\langle k4 \rangle$ Compute the vertex set P_V^{k+1} of P^{k+1}. Set $k := k + 1$ and go to $\langle k \rangle$.

Remark 5.13 *The inequality $l(\boldsymbol{x}) \leqq 0$ in Step $\langle k2.2 \rangle$ and Step $\langle k3 \rangle$ is given by one of the inequalities $\pm A\boldsymbol{x} \leqq 0$ and $\boldsymbol{x} \leqq \boldsymbol{c}$ not satisfied by the point $\boldsymbol{v}^k$, i.e.,*

(i) $l(\boldsymbol{x}) = \boldsymbol{e}^j \boldsymbol{x} - c_j$ for some $j \in \{1, \ldots, n\}$ such that $v_j^k > c_j$, or

(ii) $l(\boldsymbol{x}) = sgn(\boldsymbol{a}^i \boldsymbol{v}^k) \boldsymbol{a}^i \boldsymbol{x}$ for some $i \in \{1, \ldots, m\}$ such that $\boldsymbol{a}^i \boldsymbol{v}^k \neq 0$, where $\boldsymbol{a}^i$ is the ith row of A, and

$$sgn(\alpha) = \begin{cases} +1 & when\ \alpha > 0 \\ -1 & when\ \alpha < 0. \end{cases}$$

Lemma 5.14 *Let $\boldsymbol{z}^k$ be the locally optimal vertex obtained by applying the LS($\boldsymbol{x}^k$) procedure starting with $\boldsymbol{x}^k$ in Step $\langle k2.1 \rangle$ at iteration k, and suppose $\boldsymbol{d}\boldsymbol{z}^k > 0$. Then $\boldsymbol{d}\boldsymbol{z}^{k'} < \boldsymbol{d}\boldsymbol{z}^k$ for iteration k' such that $k' > k$.*

Proof: It suffices to show that $\boldsymbol{d}\boldsymbol{z}^{k+1} < \boldsymbol{d}\boldsymbol{z}^k$. By the construction of P^{k+1} we have $P^{k+1} \subseteqq \{ \boldsymbol{x} \mid \boldsymbol{d}\boldsymbol{x} \leqq \lceil \boldsymbol{d}\bar{\boldsymbol{x}}^{k+1} - 1 \rceil \}$. Since $\boldsymbol{x}^{k+1} \in (0, \boldsymbol{v}^{k+1}] \subseteqq P^{k+1}$ and $\boldsymbol{z}^{k+1}$ is obtained by LS($\boldsymbol{x}^{k+1}$), we have

$$\boldsymbol{d}\boldsymbol{z}^{k+1} \leqq \boldsymbol{d}\boldsymbol{x}^{k+1} \leqq \lceil \boldsymbol{d}\bar{\boldsymbol{x}}^{k+1} - 1 \rceil.$$

Since we assume that $\boldsymbol{dz}^k > 0$, we have $0 < \boldsymbol{dz}^k_\varepsilon < \boldsymbol{dz}^k$ by the choice of $\boldsymbol{z}^k_\varepsilon$. Therefore in Step $\langle k2.1\rangle$

$$\lceil \boldsymbol{d\bar{x}}^{k+1} - 1 \rceil < \boldsymbol{d\bar{x}}^{k+1} = \min\{\boldsymbol{dx}^k_\varepsilon, \boldsymbol{dz}^k_\varepsilon\} < \boldsymbol{dz}^k.$$

Combining the two inequalities yields the desired result. □

Theorem 5.15 *The OA method for (mmF) works correctly and terminates after finitely many iterations with the optimal value of (mmF).*

Proof: (correctness) If $N_M(\boldsymbol{w}^0) = \emptyset$ at the initialization step, we can conclude from the connectedness of X_M that $\boldsymbol{w}^0$ is a unique feasible solution of (mmF) and hence solves the problem. When the algorithm terminates in Step $\langle k1\rangle$, the optimal value of (mmF) is equal either to zero by Assumption 5.1 (iii), or to $\lceil \boldsymbol{d\bar{x}}^k \rceil$ by Theorem 5.12. So the optimal value is obtained whenever the algorithm terminates.

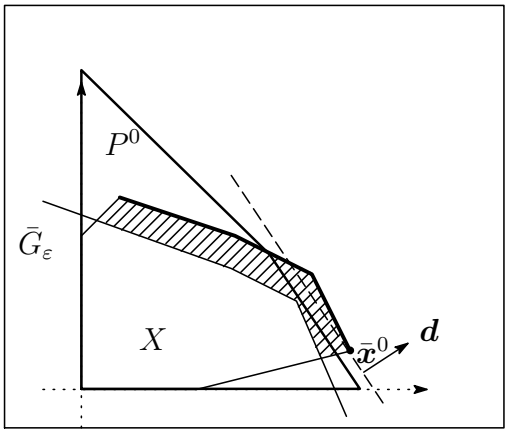
We suppose that the algorithm has not yet terminated at iteration k, i.e., $\boldsymbol{d\bar{x}}^k > 0$ and $\bar{g}(\boldsymbol{v}^k) \leqq 0$, and show that each step of the algorithm can be done. Lemma 5.8 ensures that there are points $\boldsymbol{x}^k_\varepsilon \in (\boldsymbol{0}, \boldsymbol{v}^k) \cap \partial\, \bar{G}_\varepsilon$ and $\boldsymbol{z}^k_\varepsilon \in (\boldsymbol{0}, \boldsymbol{z}^k) \cap \partial\, \bar{G}_\varepsilon$, in Step $\langle k1\rangle$ and Step $\langle k2.1\rangle$, respectively. Since $\boldsymbol{0} \in \operatorname{int} \bar{G}$ and $\boldsymbol{v}^k \notin \operatorname{int} \bar{G}$, there also exists a point $\boldsymbol{x}^k \in (\boldsymbol{0}, \boldsymbol{v}^k] \cap \partial\, \bar{G}$. When $\boldsymbol{x}^k_\varepsilon \notin X$, clearly $\boldsymbol{v}^k \notin X$, and hence the function $l : \mathbb{R}^n \to \mathbb{R}$ of Remark 5.13 can be found in Step $\langle k3\rangle$. To show that the function $l : \mathbb{R}^n \to \mathbb{R}$ is found in Step $\langle k2.2\rangle$ we have only to show that $\boldsymbol{v}^k \notin X$. Suppose the contrary, i.e., $\boldsymbol{v}^k \in X$. By the assumption that $\bar{g}(\boldsymbol{v}^k) \leqq 0$ and the fact that $\bar{g}(\boldsymbol{x}) \geqq 0$ for all $\boldsymbol{x} \in X$, we have $\bar{g}(\boldsymbol{v}^k) = 0$, i.e., $\boldsymbol{v}^k \in \partial\, \bar{G}$, and hence $\boldsymbol{v}^k \in X \setminus \operatorname{int} \bar{G} = X_M$. This implies $\boldsymbol{x}^k = \boldsymbol{v}^k \in X_M$ by the choice of $\boldsymbol{x}^k$, which contradicts that we are currently at Step $\langle k2.2\rangle$. Therefore we have seen that $\boldsymbol{v}^k \notin X$ in Step $\langle k2.2\rangle$.
(finiteness) Suppose that the polytope P^ν at iteration ν meets the condition

$$P^\nu \subseteqq X \text{ and } P^\nu \cap X_M = \emptyset, \tag{5.14}$$

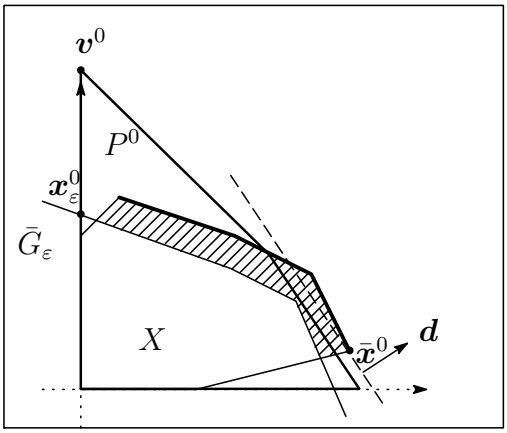
after updated either in Step $\langle k2\rangle$ or in Step $\langle k3\rangle$, and consider the next iteration. Since $\boldsymbol{v}^\nu$ is chosen from P^ν, we have $\boldsymbol{v}^\nu \in X \setminus X_M$ and consequently $\bar{g}(\boldsymbol{v}^\nu) > 0$. Then the algorithm stops at Step $\langle k1\rangle$. Therefore we have only to prove that (5.14) holds within a finite number of iterations. Note first that both Step $\langle k2.2\rangle$ and Step $\langle k3\rangle$ are done only a finite number of times. By the definition of affine function l, the

polytope, say $P^{k'}$, when $2m+n$ cuts $l(\boldsymbol{x}) \leqq 0$ have been added to the initial polytope P^0, is contained in X. Therefore $\boldsymbol{v}^{k'}$ as well as $\boldsymbol{x}_\varepsilon^{k'}$ lies in X, and hence we obtain that $\boldsymbol{x}^{k'} = \boldsymbol{v}^{k'} \in X_M$ in the same way as in the former part of this proof. Therefore we come to neither Step $\langle k2.2\rangle$ nor Step $\langle k3\rangle$ after iteration k'. Namely, Step $\langle k2.1\rangle$ followed by Step $\langle k4\rangle$ repeats itself after iteration k'. For iteration k with $k \geqq k'+1$, we have $\boldsymbol{x}^k \in X_M$. We then locate $\boldsymbol{z}^k \in X_M \cap X_V$ by applying the LS($\boldsymbol{x}^k$) procedure and obtain a point $\boldsymbol{z}_\varepsilon^k \in (\boldsymbol{0}, \boldsymbol{z}^k) \cap \partial \bar{G}_\varepsilon$. If $\boldsymbol{dz}^k = 0$ for some $k \geqq k'+1$ then we set $\bar{\boldsymbol{x}}^{k+1} := \boldsymbol{z}_\varepsilon^k$ since $\boldsymbol{dz}_\varepsilon^k = \boldsymbol{dz}^k = 0 \leqq \boldsymbol{dx}_\varepsilon^k$. Then the incumbent value $\boldsymbol{d}\bar{\boldsymbol{x}}^{k+1}$ becomes zero, and hence the algorithm stops in Step $\langle k1\rangle$ at the next iteration. If $\boldsymbol{dz}^k > 0$ for all k with $k \geqq k'+1$, we see that $\boldsymbol{dz}^{k+1} < \boldsymbol{dz}^k$ for all $k \geqq k'+1$ by Lemma 5.14. Since $|X_M \cap X_V|$ is finite, we eventually obtain a point $\boldsymbol{z}^{\nu-1} \in X_M \cap X_V$ such that $\boldsymbol{dz}^{\nu-1} \leqq \boldsymbol{dz}$ for all $\boldsymbol{z} \in X_M \cap X_V$. Also we have $\boldsymbol{dz}_\varepsilon^{\nu-1} < \boldsymbol{dz}^{\nu-1}$ by the choice of $\boldsymbol{z}_\varepsilon^{\nu-1}$. The polytope P^ν is then defined as $P^\nu := P^{\nu-1} \cap \{\, \boldsymbol{x} \mid \boldsymbol{dx} \leqq \lceil \boldsymbol{d}\bar{\boldsymbol{x}}^\nu - 1 \rceil \,\}$, where $\bar{\boldsymbol{x}}^\nu$ satisfies that $\boldsymbol{d}\bar{\boldsymbol{x}}^\nu = \min\{\, \boldsymbol{dx}_\varepsilon^{\nu-1}, \boldsymbol{dz}_\varepsilon^{\nu-1} \,\} < \boldsymbol{dz}^{\nu-1}$. This means that $P^\nu \cap (X_M \cap X_V) = \emptyset$. Since X_M is a connected union of several faces of X, we see that $\boldsymbol{dz}^{\nu-1} \leqq \boldsymbol{dx}$ for all $\boldsymbol{x} \in X_M$. Therefore we conclude that $P^\nu \cap X_M = \emptyset$. □

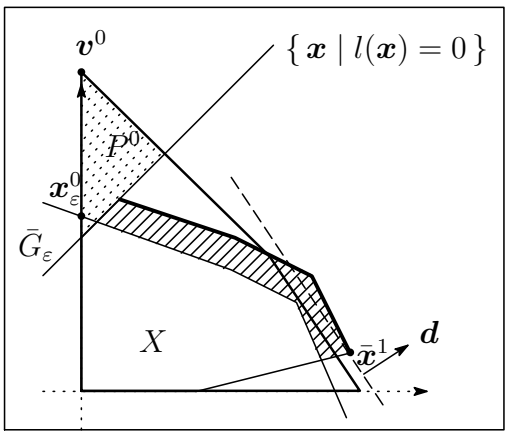
We illustrate the OA method for (mmF) in Figure 5.5. We obtain a locally optimal vertex $\bar{\boldsymbol{x}}^0 \in X_M \cap X_V$ and set up an initial polytope P^0 (See (a)). It is easy to enumerate all vertices of P^0 because this polytope is simply given by $P^0 := \{\, \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{ex} \leqq \zeta,\ \boldsymbol{dx} \leqq \boldsymbol{d}\bar{\boldsymbol{x}}^0 - 1,\ \boldsymbol{x} \geqq \boldsymbol{0} \,\}$. We obtain a point $\boldsymbol{v}^0 \in P_V^0$ such that $\bar{g}(\boldsymbol{v}) \leqq 0$ and a point $\boldsymbol{x}_\varepsilon^0 \in (\boldsymbol{0}, \boldsymbol{v}^0) \cap \partial \bar{G}_\varepsilon$ (See (b)). We see that $\boldsymbol{x}_\varepsilon^0 \notin X$, and hence set $\bar{\boldsymbol{x}}^1 := \bar{\boldsymbol{x}}^0$ and cut off $\boldsymbol{v}^0$ from P^0 (See (c)). In the next iteration, we obtain points $\boldsymbol{v}^1, \boldsymbol{x}_\varepsilon^1$ and $\boldsymbol{x}^1$. Since $\boldsymbol{x}^1 \in X_M$, we apply the LS($\boldsymbol{x}^1$) procedure to obtain a point $\boldsymbol{z}^1$, and obtain a point $\boldsymbol{z}_\varepsilon^1 \in (\boldsymbol{0}, \boldsymbol{z}^1) \cap \partial \bar{G}_\varepsilon$ (See (d)). We find a point $\boldsymbol{z}_\varepsilon^1 \in X \setminus \operatorname{int} \bar{G}_\varepsilon$ such that $\boldsymbol{dz}_\varepsilon^1 < \boldsymbol{dx}_\varepsilon^1$. We then set $\bar{\boldsymbol{x}}^2 := \boldsymbol{z}_\varepsilon^1$ and construct P^2 by adding the cut $\boldsymbol{dx} \leqq \lceil \boldsymbol{d}\bar{\boldsymbol{x}}^2 - 1 \rceil$ to P^1 (See (e)). Because $\bar{g}(\boldsymbol{v}) > 0$ for all vertices $\boldsymbol{v}$ of P^2 (See (f)), we terminate at the next iteration with the optimal value $\lceil \boldsymbol{d}\bar{\boldsymbol{x}}^2 \rceil$.



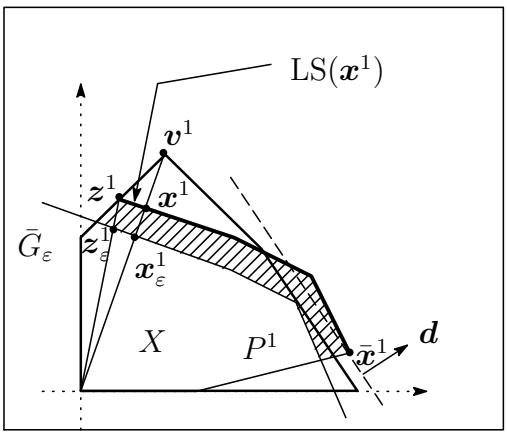
(a) polytope P^0



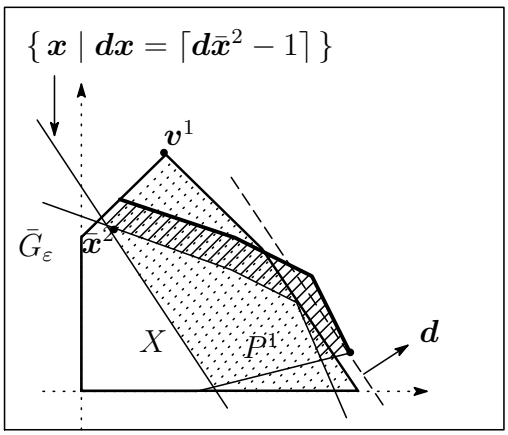
(b) obtaining $\boldsymbol{v}^0$ and $\boldsymbol{x}^0_\varepsilon$



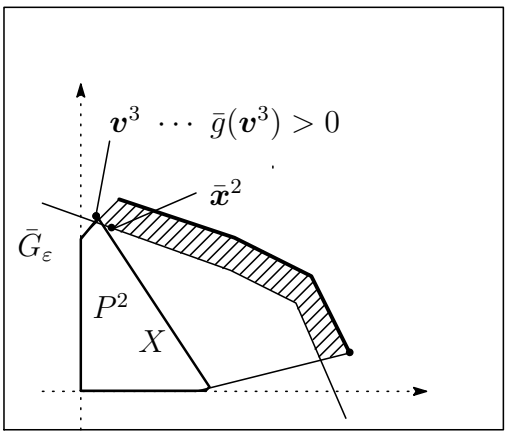
(c) cutting off $\boldsymbol{v}^0$ from P^0



(d) obtaining $\boldsymbol{v}^1$, $\boldsymbol{x}^1_\varepsilon$ and $\boldsymbol{z}^1_\varepsilon$



(e) cutting off $\boldsymbol{v}^1$ from P^1



(f) termination

Figure 5.5: OA method for (mmF)

5.4.3 Implementation issue about the OA method for (mmF)

The key implementation issue about OA method for (mmF) is the way of computing the point $\boldsymbol{x} \in (\boldsymbol{0}, \boldsymbol{v}) \cap \partial \bar{G}_\gamma$ for a given point $\boldsymbol{v} \in P_V^k$ such that $\bar{g}(\boldsymbol{v}) \leqq 0$, and $\gamma \in \{0, \varepsilon\}$. In other words, for a given $\boldsymbol{v}$ and γ, we compute $\theta \in (0, 1]$ such that

$$\bar{g}(\theta \boldsymbol{v}) = \gamma, \tag{5.15}$$

and set $\boldsymbol{x} := \theta \boldsymbol{v}$. As seen in Section 5.2.1 the extended gap function $\bar{g}(\theta \boldsymbol{v})$ is given by the optimal value of

$$(P(\theta \boldsymbol{v})) \quad \left| \begin{array}{ll} \max\limits_{(\boldsymbol{y}, \boldsymbol{t})} & \boldsymbol{e}\boldsymbol{y} - n\boldsymbol{e}\boldsymbol{t} - \theta \boldsymbol{e}\boldsymbol{v} \\ \text{s.t.} & A\boldsymbol{y} = \boldsymbol{0} \\ & \boldsymbol{y} \leqq \boldsymbol{c} \\ & \boldsymbol{y} + \boldsymbol{t} \geqq \theta \boldsymbol{v} \\ & (\boldsymbol{y}, \boldsymbol{t}) \geqq \boldsymbol{0}. \end{array} \right.$$

Introducing slack variables $\boldsymbol{s}$, $(P(\theta \boldsymbol{v}))$ is equivalent to

$$\left| \begin{array}{ll} \max\limits_{(\boldsymbol{y}, \boldsymbol{t}, \boldsymbol{s}^1, \boldsymbol{s}^2)} & \boldsymbol{e}\boldsymbol{y} - n\boldsymbol{e}\boldsymbol{t} - \theta \boldsymbol{e}\boldsymbol{v} \\ \text{s.t.} & A\boldsymbol{y} = \boldsymbol{0} \\ & \boldsymbol{y} + \boldsymbol{s}^1 = \boldsymbol{c} \\ & \boldsymbol{y} + \boldsymbol{t} - \boldsymbol{s}^2 = \theta \boldsymbol{v} \\ & (\boldsymbol{y}, \boldsymbol{t}, \boldsymbol{s}^1, \boldsymbol{s}^2) \geqq \boldsymbol{0}. \end{array} \right.$$

The tableau of this problem is

A	O	O	O	$\boldsymbol{0}$
I	O	I	O	$\boldsymbol{c}$
I	I	O	$-I$	$\theta \boldsymbol{v}$
$\boldsymbol{e}$	$-n\boldsymbol{e}$	$\boldsymbol{0}$	$\boldsymbol{0}$	$\max(-\theta \boldsymbol{e}\boldsymbol{v})$

(5.16)

For a basic matrix B and a nonbasic matrix N, let $(\boldsymbol{d}^B, \boldsymbol{d}^N)$ be the partition of

$\begin{bmatrix} \boldsymbol{e} & -n\boldsymbol{e} & \mathbf{0} & \mathbf{0} \end{bmatrix}$, and let $\boldsymbol{\pi}^B = \boldsymbol{d}^B B^{-1}$. We then define the linear function

$$\bar{g}^B(\theta) = \boldsymbol{\pi}^B \begin{bmatrix} \mathbf{0} \\ \boldsymbol{c} \\ \theta\boldsymbol{v} \end{bmatrix} - \theta \boldsymbol{e}\boldsymbol{v}, \tag{5.17}$$

with respect to the basic matrix B. Note that $\bar{g}^B(\theta)$ is a nonincreasing piece-wise linear concave function. Using the above function, we apply the parametric linear programming technique described as follows.

/** procedure of computing $\theta \in (0, 1]$ such that $\bar{g}(\theta\boldsymbol{v}) = \gamma$ **/

$\langle 0 \rangle$ Set $\theta_0 := 1$ and $k := 0$.

$\langle k \rangle$ Solve $(P(\theta_k \boldsymbol{v}))$ to obtain the optimal basic matrix B_k with the optimal value β. If $\beta = \gamma$ the stop (we obtain θ_k to be computed). Otherwise compute θ_{k+1} satisfying $\bar{g}^{B_k}(\theta_{k+1}) = \gamma$. Set $k := k + 1$ and go to $\langle k \rangle$.

We illustrate this procedure in Figure 5.6.

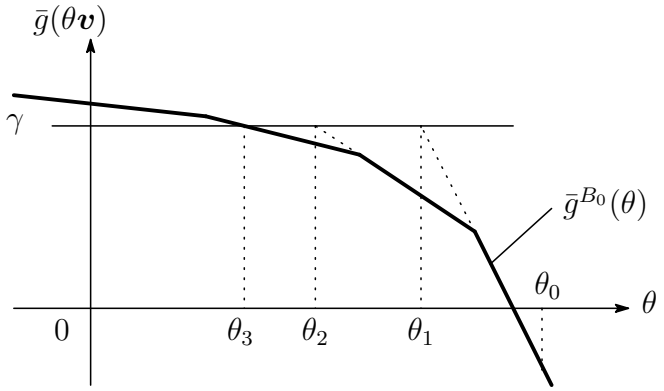


Figure 5.6: Procedure of computing $\theta \in (0, 1]$ such that $\bar{g}(\theta\boldsymbol{v}) = \gamma$

5.5 Reducing the Number of Variables in (mmF)

If we reduce the number of variables in (mmF), we can make algorithms more efficiently. The dimension of $X = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \mathbf{0},\ \mathbf{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c} \,\}$ is much smaller

than the number of variables in many cases. This property seems to be useful, however, there was some difficulties to overcome (See Appendix C).

In this section we improve the algorithms by reducing the number of variables. To do this we introduce an idea of decomposition by the basic matrix. After we modify the extended gap function and local search procedure, we state the improved CS method and improved OA method. We discuss the effect of the decomposition, and how to modify these algorithms in the case where a network does not hold Assumption5.1 (i).

5.5.1 Decomposition by the basic matrix

For an incidence matrix A, let B and N be a basic matrix and a nonbasic matrix, respectively, of the system $A\boldsymbol{x} = \mathbf{0}$. Note that B has the inverse matrix B^{-1}. After we divide $\boldsymbol{c}$ and $\boldsymbol{d}$ into two part, say $(\boldsymbol{c}^B, \boldsymbol{c}^N)$ and $(\boldsymbol{d}^B, \boldsymbol{d}^N)$, corresponding to B and N, respectively, we rearrange $\boldsymbol{c}$, $\boldsymbol{d}$ and columns of A so that $\boldsymbol{c} = (\boldsymbol{c}^B, \boldsymbol{c}^N)$, $\boldsymbol{d} = (\boldsymbol{d}^B, \boldsymbol{d}^N)$ and $A = \begin{bmatrix} B & N \end{bmatrix}$. Note also that the above matrix B is fixed throughout the algorithm. We then define

$$
\begin{aligned}
X^R &= \{\, \boldsymbol{x}^N \in \mathbb{R}^{n-m} \mid \boldsymbol{x} \in X,\ \boldsymbol{x} = (\boldsymbol{x}^B, \boldsymbol{x}^N) \,\} \qquad (5.18)\\
&= \{\, \boldsymbol{x}^N \in \mathbb{R}^{n-m} \mid B\boldsymbol{x}^B + N\boldsymbol{x}^N = \mathbf{0},\ \mathbf{0} \leqq \boldsymbol{x}^B \leqq \boldsymbol{c}^B,\ \mathbf{0} \leqq \boldsymbol{x}^N \leqq \boldsymbol{c}^N \,\}\\
&= \{\, \boldsymbol{x}^N \in \mathbb{R}^{n-m} \mid \boldsymbol{x}^B = -B^{-1}N\boldsymbol{x}^N,\ \mathbf{0} \leqq \boldsymbol{x}^B \leqq \boldsymbol{c}^B,\ \mathbf{0} \leqq \boldsymbol{x}^N \leqq \boldsymbol{c}^N \,\}\\
&= \{\, \boldsymbol{x}^N \in \mathbb{R}^{n-m} \mid \mathbf{0} \leqq -B^{-1}N\boldsymbol{x}^N \leqq \boldsymbol{c}^B,\ \mathbf{0} \leqq \boldsymbol{x}^N \leqq \boldsymbol{c}^N \,\}\\
&= \{\, \boldsymbol{x}^N \in \mathbb{R}^{n-m} \mid \mathbf{0} \leqq T\boldsymbol{x}^N \leqq \boldsymbol{c} \,\},
\end{aligned}
$$

where

$$
T = \begin{bmatrix} -B^{-1}N \\ I \end{bmatrix}. \qquad (5.19)
$$

Clearly $\mathbf{0} \in X_V^R$ and $X^R \subseteqq \{\, \boldsymbol{x} \in \mathbb{R}^{n-m} \mid \boldsymbol{x} \geqq \mathbf{0} \,\}$. Furthermore we have the following lemmas.

Lemma 5.16

(i) $(\boldsymbol{x}^B, \boldsymbol{x}^N) \in \{\, \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \mathbf{0} \,\} \Leftrightarrow \boldsymbol{x}^B = -B^{-1}N\boldsymbol{x}^N$ *and* $\boldsymbol{x}^N \in \mathbb{R}^{n-m}$.

(ii) $(\boldsymbol{x}^B, \boldsymbol{x}^N) \in X \Leftrightarrow \boldsymbol{x}^B = -B^{-1}N\boldsymbol{x}^N$ *and* $\boldsymbol{x}^N \in X^R$.

(iii) $(\boldsymbol{v}^B, \boldsymbol{v}^N) \in X_V \Leftrightarrow \boldsymbol{v}^B = -B^{-1}N\boldsymbol{v}^N$ *and* $\boldsymbol{v}^N \in X_V^R$.

Proof: Since (i) and (ii) are clear, we only show the assertion (iii).
(iii) $(\Leftarrow)$ Assume that $(\boldsymbol{v}^B, \boldsymbol{v}^N) \notin X_V$. If $(\boldsymbol{v}^B, \boldsymbol{v}^N) \notin X$ then $\boldsymbol{v}^B \neq -B^{-1}N\boldsymbol{v}^N$ or $\boldsymbol{v}^N \notin X^R$ by (ii), and hence we have done. Otherwise, there are at least two or more distinct vertices of X, say $\boldsymbol{v}^1, \cdots, \boldsymbol{v}^k \in X_V$, such that $(\boldsymbol{v}^B, \boldsymbol{v}^N) = \sum_{j=1}^k \theta_k \boldsymbol{v}^k$ for some $\theta_j > 0$ $(j = 1, \ldots, k)$. Let $(\boldsymbol{v}^{(j,B)}, \boldsymbol{v}^{(j,N)})$ be the partition of $\boldsymbol{v}^j$ corresponding to B and N for each $j = 1, \ldots, k$. Clearly $\boldsymbol{v}^{(j,N)} \in X^R$ for each $j = 1, \ldots, k$ by using (ii). When $i \neq j$ we see that $\boldsymbol{v}^{(i,N)} \neq \boldsymbol{v}^{(j,N)}$, since $\boldsymbol{v}^i \neq \boldsymbol{v}^j$. Therefore we have $\boldsymbol{v}^{(j,N)} \in X^R$ and $\theta_j > 0$ $(j = 1, \ldots, k)$ such that $\boldsymbol{v}^N = \sum_{j=1}^k \theta_j \boldsymbol{v}^{(j,N)}$, and $\boldsymbol{v}^{(i,N)} \neq \boldsymbol{v}^{(j,N)}$ if $i \neq j$. This means $\boldsymbol{v}^N \notin X_V^R$.
$(\Rightarrow)$ If $\boldsymbol{v}^B \neq -B^{-1}N\boldsymbol{v}^N$ or $\boldsymbol{v}^N \notin X^R$, we have $(\boldsymbol{v}^B, \boldsymbol{v}^N) \notin X$ by (ii), then we have done. Assume that $\boldsymbol{v}^N \notin X_V^R$ and $\boldsymbol{v}^B = -B^{-1}N\boldsymbol{v}^N$, and let $\boldsymbol{v} = (\boldsymbol{v}^B, \boldsymbol{v}^N)$. There are at least two or more distinct vertices of X^R, say $\boldsymbol{v}^{(1,N)}, \cdots, \boldsymbol{v}^{(k,N)} \in X_V^R$, such that $\boldsymbol{v}^N = \sum_{j=1}^k \theta_k \boldsymbol{v}^{(k,N)}$ for some $\theta_j > 0$ $(j = 1, \ldots, k)$. Let $\boldsymbol{v}^{(j,B)} = -B^{-1}N\boldsymbol{v}^{(j,N)}$ and $\boldsymbol{v}^j = (\boldsymbol{v}^{(j,B)}, \boldsymbol{v}^{(j,N)})$ for each $j = 1, \ldots, k$. Clearly $\boldsymbol{v}^j \in X$ for each $j = 1, \ldots, k$ by using (ii). When $i \neq j$ we see that $\boldsymbol{v}^i \neq \boldsymbol{v}^j$, since $\boldsymbol{v}^{(i,N)} \neq \boldsymbol{v}^{(j,N)}$. Therefore we have $\boldsymbol{v}^j \in X$ and $\theta_j > 0$ $(j = 1, \ldots, k)$ such that $\boldsymbol{v} = \sum_{j=1}^k \theta_j \boldsymbol{v}^j$, and $\boldsymbol{v}^i \neq \boldsymbol{v}^j$ if $i \neq j$. This means $\boldsymbol{v} \notin X_V$. □

Lemma 5.17 (Full dimensionality) *X^R has full dimension, i.e., $\dim X^R = n - m$.*

Proof: Lemma 5.2 ensures that there is a vector $\boldsymbol{x} \in X$ such that $\boldsymbol{0} < \boldsymbol{x} < \boldsymbol{c}$. By Lemma 5.16 there is a vector $\boldsymbol{x}^N \in X^R$ such that $\boldsymbol{0} < T\boldsymbol{x}^N < \boldsymbol{c}$, i.e., $\boldsymbol{x}^N \in \operatorname{int} X^R$. This implies that $\dim X^R = n - m$. □

For X^R we define

$$X_M^R = \{\, \boldsymbol{x}^N \in X^R \mid \nexists \boldsymbol{y}^N \in X^R : T\boldsymbol{y}^N \geqq T\boldsymbol{x}^N,\ T\boldsymbol{y}^N \neq T\boldsymbol{x}^N \,\}. \tag{5.20}$$

Theorem 5.18

(i) $(\boldsymbol{x}^B, \boldsymbol{x}^N) \in X_M \quad \Leftrightarrow \quad \boldsymbol{x}^B = -B^{-1}N\boldsymbol{x}^N$ *and* $\boldsymbol{x}^N \in X_M^R$.

(ii) $\min\{\, \boldsymbol{d}\boldsymbol{x} \mid \boldsymbol{x} \in X_M \,\} = \min\{\, \boldsymbol{d}T\boldsymbol{x}^N \mid \boldsymbol{x}^N \in X_M^R \,\}$.

Proof: (i) $(\Leftarrow)$ If $(\boldsymbol{x}^B, \boldsymbol{x}^N) \notin X$ then $\boldsymbol{x}^B \neq -B^{-1}N\boldsymbol{x}^N$ or $\boldsymbol{x}^N \notin X^R$ by Lemma 5.16 (ii), hence we have done. If $(\boldsymbol{x}^B, \boldsymbol{x}^N) \in X \setminus X_M$, there is $(\boldsymbol{y}^B, \boldsymbol{y}^N) \in X$

such that $(\boldsymbol{y}^B, \boldsymbol{y}^N) \geqq (\boldsymbol{x}^B, \boldsymbol{x}^N)$ and $(\boldsymbol{y}^B, \boldsymbol{y}^N) \neq (\boldsymbol{x}^B, \boldsymbol{x}^N)$. Using Lemma 5.16 (ii) again, we see that $\boldsymbol{x}^B = -B^{-1}N\boldsymbol{x}^N$, $\boldsymbol{x}^N \in X^R$, $\boldsymbol{y}^B = -B^{-1}N\boldsymbol{y}^N$ and $\boldsymbol{y}^N \in X^R$. We then have $\boldsymbol{y}^N \in X^R$ such that $T\boldsymbol{y}^N \geqq T\boldsymbol{x}^N$ and $T\boldsymbol{y}^N \neq T\boldsymbol{x}^N$, which implies $\boldsymbol{x}^N \notin X_M^R$.

$(\Rightarrow)$ If $\boldsymbol{x}^B \neq -B^{-1}N\boldsymbol{x}^N$ then $(\boldsymbol{x}^B, \boldsymbol{x}^N) \notin X$ by Lemma 5.16 (ii), hence we have done. If $\boldsymbol{x}^N \notin X_M^R$, there is $\boldsymbol{y}^N \in X^R$ such that $T\boldsymbol{y}^N \geqq T\boldsymbol{x}^N$, $T\boldsymbol{y}^N \neq T\boldsymbol{x}^N$. Letting $\boldsymbol{x}^B = -B^{-1}N\boldsymbol{x}^N$ and $\boldsymbol{y}^B = -B^{-1}N\boldsymbol{y}^N$, we have $(\boldsymbol{y}^B, \boldsymbol{y}^N) \in X$ such that $(\boldsymbol{y}^B, \boldsymbol{y}^N) \geqq (\boldsymbol{x}^B, \boldsymbol{x}^N)$ and $(\boldsymbol{y}^B, \boldsymbol{y}^N) \neq (\boldsymbol{x}^B, \boldsymbol{x}^N)$. Then $(\boldsymbol{x}^B, \boldsymbol{x}^N) \notin X_M$.

(ii) Applying (i), it is clear that

$$\begin{aligned}
\min\{\, \boldsymbol{d}\boldsymbol{x} \mid \boldsymbol{x} \in X_M \,\} &= \min\{\, \boldsymbol{d}^B\boldsymbol{x}^B + \boldsymbol{d}^N\boldsymbol{x}^N \mid (\boldsymbol{x}^B, \boldsymbol{x}^N) \in X_M \,\} \\
&= \min\{\, \boldsymbol{d}^B\boldsymbol{x}^B + \boldsymbol{d}^N\boldsymbol{x}^N \mid \boldsymbol{x}^B = -B^{-1}N\boldsymbol{x}^N,\ \boldsymbol{x}^N \in X_M^R \,\} \\
&= \min\{\, (-\boldsymbol{d}^B B^{-1}N + \boldsymbol{d}^N)\boldsymbol{x}^N \mid \boldsymbol{x}^N \in X_M^R \,\} \\
&= \min\{\, \boldsymbol{d}T\boldsymbol{x}^N \mid \boldsymbol{x}^N \in X_M^R \,\}.
\end{aligned}$$

□

By Theorem 5.18, (mmF) can be cast into the following problem.

Definition 5.19 (Modified minimum maximal flow problem)

$$(mmF^R) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x} \in \mathbb{R}^{n-m}} & \boldsymbol{d}T\boldsymbol{x} \\ s.t. & \boldsymbol{x} \in X_M^R. \end{array} \right.$$

Let (MC) be defined for $C = T$ and $X = X^R$, and let $\phi(\boldsymbol{x}) = \boldsymbol{d}T\boldsymbol{x}$. Then (mmF^R) also reduces to Problem (P_E).

5.5.2 Modification of extended gap function and a local search procedure

As seen in Section 3.2, the set X_M^R of (5.20) can be written as $X_M^R = \{\, \boldsymbol{x} \in X^R \mid g^R(\boldsymbol{x}) \leqq 0 \,\}$ by using

$$g^R(\boldsymbol{x}) = \max\{\, \boldsymbol{e}T\boldsymbol{y} \mid \boldsymbol{y} \in X^R,\ T\boldsymbol{y} \geqq T\boldsymbol{x} \,\} - \boldsymbol{e}T\boldsymbol{x}. \tag{5.21}$$

Furthermore defining

$$\bar{g}^R(\boldsymbol{x}) = \max\{\, \boldsymbol{e}T\boldsymbol{y} - n\boldsymbol{e}\boldsymbol{t} \mid \boldsymbol{y} \in X^R,\ T\boldsymbol{y} + \boldsymbol{t} \geqq T\boldsymbol{x},\ \boldsymbol{t} \geqq \boldsymbol{0} \,\} - \boldsymbol{e}T\boldsymbol{x}, \tag{5.22}$$

we have

Theorem 5.20

(*i*) *The domain of* $\bar{g}^R$ *is* $\mathbb{R}^{n-m}$.

(*ii*) $\bar{g}^R(\boldsymbol{x}^N) = g^R(\boldsymbol{x}^N)$ *for all* $\boldsymbol{x}^N \in X^R$.

Proof: For any $\boldsymbol{x}^N \in \mathbb{R}^{n-m}$ we have

$$\begin{aligned}
g(T\boldsymbol{x}^N) &= \max\{\, \boldsymbol{e}\boldsymbol{y} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} \geqq T\boldsymbol{x}^N \,\} - \boldsymbol{e}T\boldsymbol{x}^N \\
&= \max\{\, \boldsymbol{e}(\boldsymbol{y}^B + \boldsymbol{y}^N) \mid (\boldsymbol{y}^B, \boldsymbol{y}^N) \in X,\ (\boldsymbol{y}^B, \boldsymbol{y}^N) \geqq T\boldsymbol{x}^N \,\} - \boldsymbol{e}T\boldsymbol{x}^N \\
&= \max\{\, \boldsymbol{e}(\boldsymbol{y}^B + \boldsymbol{y}^N) \mid \boldsymbol{y}^B = -B^{-1}N\boldsymbol{y}^N,\ \boldsymbol{y}^N \in X^R,\ (\boldsymbol{y}^B, \boldsymbol{y}^N) \geqq T\boldsymbol{x}^N \,\} - \boldsymbol{e}T\boldsymbol{x}^N \\
&= \max\{\, \boldsymbol{e}T\boldsymbol{y}^N \mid \boldsymbol{y}^N \in X^R,\ T\boldsymbol{y}^N \geqq T\boldsymbol{x}^N \,\} - \boldsymbol{e}T\boldsymbol{x}^N \\
&= g^R(\boldsymbol{x}^N),
\end{aligned}$$

and

$$\begin{aligned}
\bar{g}(T\boldsymbol{x}^N) &= \max\{\, \boldsymbol{e}\boldsymbol{y} - n\boldsymbol{e}\boldsymbol{t} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} + \boldsymbol{t} \geqq T\boldsymbol{x}^N,\ \boldsymbol{t} \geqq \boldsymbol{0} \,\} - \boldsymbol{e}T\boldsymbol{x}^N \\
&= \max\{\, \boldsymbol{e}(\boldsymbol{y}^B + \boldsymbol{y}^N) - n\boldsymbol{e}\boldsymbol{t} \mid (\boldsymbol{y}^B, \boldsymbol{y}^N) \in X,\ (\boldsymbol{y}^B, \boldsymbol{y}^N) + \boldsymbol{t} \geqq T\boldsymbol{x}^N,\ \boldsymbol{t} \geqq \boldsymbol{0} \,\} - \boldsymbol{e}T\boldsymbol{x}^N \\
&= \max\{\, \boldsymbol{e}T\boldsymbol{y}^N - n\boldsymbol{e}\boldsymbol{t} \mid \boldsymbol{y}^N \in X^R,\ T\boldsymbol{y}^N + \boldsymbol{t} \geqq T\boldsymbol{x}^N,\ \boldsymbol{t} \geqq \boldsymbol{0} \,\} - \boldsymbol{e}T\boldsymbol{x}^N \\
&= \bar{g}^R(\boldsymbol{x}^N).
\end{aligned}$$

(*i*) By the above observation and the domain of $\bar{g}$ is $\mathbb{R}^n$, we see that $\bar{g}^R(\boldsymbol{x}^N) = \bar{g}(T\boldsymbol{x}^N) > -\infty$ for any $\boldsymbol{x}^N \in \mathbb{R}^{n-m}$, and hence the domain of $\bar{g}^R$ is $\mathbb{R}^{n-m}$.
(*ii*) If $\boldsymbol{x}^N \in X^R$ then $T\boldsymbol{x}^N \in X$ by Lemma 5.16 (*ii*). Since $\bar{g} = g$ on domain of g and $X \subseteqq \operatorname{dom} g$, we have $\bar{g}^R(\boldsymbol{x}^N) = \bar{g}(T\boldsymbol{x}^N) = g(T\boldsymbol{x}^N) = g^R(\boldsymbol{x}^N)$ for all $\boldsymbol{x}^N \in X^R$.
□

Note that g^R and $\bar{g}^R$ are also piece-wise linear concave functions. Note also that $g^R(\boldsymbol{0}) = \bar{g}^R(\boldsymbol{0}) > 0$. By Theorem 5.20, to solve (mmF) we consider

$$(mmF^R) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x} \in \mathbb{R}^{n-m}} & \boldsymbol{d}T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in X^R \setminus \operatorname{int} \bar{G}^R, \end{array} \right.$$

where

$$\bar{G}^R = \{\, \boldsymbol{x} \in X^R \mid \bar{g}^R(\boldsymbol{x}^N) \geqq 0 \,\}. \tag{5.23}$$

For $\boldsymbol{v} \in X_M^R \cap X_V^R$, we define the set of efficient vertices linked to $\boldsymbol{v}$ by an edge as

$$\begin{aligned} N_M^R(\boldsymbol{v}) &= \{\, \boldsymbol{v}' \in X_M^R \cap X_V^R \mid [\boldsymbol{v}, \boldsymbol{v}'] \text{ is an edge of } X^R \,\} \\ &= \{\, \boldsymbol{v}' \in X_V^R \mid [\boldsymbol{v}, \boldsymbol{v}'] \text{ is an edge of } X^R \text{ and } \bar{g}^R(\boldsymbol{v}') \leqq 0 \,\}. \end{aligned} \tag{5.24}$$

Whenever we find a feasible solution $\boldsymbol{w} \in X_M^R$, we apply the modified local search procedure starting with $\boldsymbol{w}$ (modified LS($\boldsymbol{w}$) for short) for further improvement.

The procedure is described as follows.

/** modified LS($\boldsymbol{w}$) procedure **/

$\langle 0 \rangle$ (initialization) If $\boldsymbol{w} \notin X_V^R$ then solve $\min\{\, \boldsymbol{d}T\boldsymbol{x} \mid \boldsymbol{x} \in F^R(\boldsymbol{w}) \,\}$, where $F^R(\boldsymbol{w})$ is the face of X^R containing $\boldsymbol{w}$ in its relative interior, to obtain a vertex $\boldsymbol{v}^0 \in X_M^R \cap X_V^R$, otherwise set $\boldsymbol{v}^0 := \boldsymbol{w}$. Set $k := 0$.

$\langle k \rangle$ (iteration k) If $\{\, \boldsymbol{v} \in N_M^R(\boldsymbol{v}^k) \mid \boldsymbol{d}T\boldsymbol{v} < \boldsymbol{d}T\boldsymbol{v}^k \,\} = \emptyset$ then stop ($\boldsymbol{v}^k$ is the locally optimal vertex of (mmF^R)). Otherwise choose $\boldsymbol{v}^{k+1}$ in the set, set $k := k + 1$ and go to $\langle k \rangle$.

Remark 5.21 *Whenever $\boldsymbol{w} \in X_M^R$, the face $F^R(\boldsymbol{w})$ of X^R containing $\boldsymbol{w}$ in its relative interior is contained in X_M^R since X_M^R is a connected union of several faces of X^R.*

5.5.3 Improved CS method for (mmF)

Problem (mmF^R) holds every conditions for the CS method and hence we can directly apply this algorithm to (mmF^R). The improved CS method for (mmF) is described as follows.

/** improved CS method for (mmF) **/

$\langle 0 \rangle$ (initialization) Find an initial feasible vertex $\boldsymbol{w}^0 \in X_M^R \cap X_V^R$ of (mmF^R). If $N_M^R(\boldsymbol{w}^0) = \emptyset$ then stop ($T\boldsymbol{w}^0$ is the unique optimal solution of (mmF)). Otherwise, apply the modified LS($\boldsymbol{w}^0$) procedure to obtain a locally optimal vertex $\bar{\boldsymbol{x}} \in X_M^R \cap X_V^R$. Set $K_0 := \mathbb{R}_+^{n-m}$, $\mathcal{S} := \{K_0\}$, $\mathcal{R} := \mathcal{S}$, $\gamma := 0$ and $k := 0$.

$\langle k \rangle$ (iteration k) For each $K \in \mathcal{S}$, solve $\beta_K := \min\{\, \boldsymbol{d}T\boldsymbol{x} \mid \boldsymbol{x} \in X^R \cap K,\ l_K(\boldsymbol{x}) \geqq 0 \,\}$ to obtain a solution $\boldsymbol{\omega}^K$ if the problem is feasible, set $\beta_k := \infty$ otherwise, where $l_K(\boldsymbol{x}) \geqq 0$ is the concavity cut for $K \setminus \bar{G}^R$.

$\langle k1 \rangle$ (update) Set $\mathcal{L} := \{\, K \in \mathcal{S} \mid \boldsymbol{\omega}^K \in X_M^R \,\}$. If $\mathcal{L} \neq \emptyset$ then apply the modified LS($\boldsymbol{\omega}^K$) procedure to obtain a locally optimal vertex $\boldsymbol{v}^K$ for each $K \in \mathcal{L}$. Solve $\min\{\, \boldsymbol{d}T\boldsymbol{v}^K \mid K \in \mathcal{L} \,\}$ to obtain the cone K^*. If $\boldsymbol{d}T\boldsymbol{v}^{K^*} < \boldsymbol{d}T\bar{\boldsymbol{x}}$, set $\bar{\boldsymbol{x}} := \boldsymbol{v}^{K^*}$.

$\langle k2 \rangle$ (termination) Set $\mathcal{R}' := \{\, K \in \mathcal{R} \mid \beta_K < \boldsymbol{d}T\bar{\boldsymbol{x}} \,\}$. If $\mathcal{R}' = \emptyset$ or $\boldsymbol{d}T\bar{\boldsymbol{x}} - \gamma < 1$ then stop ($T\bar{\boldsymbol{x}}$ solves (mmF)).

$\langle k3 \rangle$ (subdivision) Solve $\min\{\, \beta_K \mid K \in \mathcal{R}' \,\}$ to obtain the cone K^{**}. If $\beta_{K^{**}} > \gamma$ then set $\gamma := \beta_{K^{**}}$. Perform the $\boldsymbol{\omega}$-subdivision on K^{**} for some $\boldsymbol{\omega} \in K^{**}$, and let $\mathcal{S}^{**}$ be the partition of K^{**}. Set $\mathcal{S} := \mathcal{S}^{**}$, $\mathcal{R} := \mathcal{S}^{**} \cup (\mathcal{R}' \setminus \{K^{**}\})$, $k := k + 1$ and go to $\langle k \rangle$.

5.5.4 Improved OA method for (mmF)

To be applied the OA method (mmF^R) must be hold regularity condition, however, (mmF^R) is not regular by the same reason of (mmF). Thus we introduce an idea of ε-optimal solution again. Additionally we need to study the range of ε and optimality condition, and prove finite convergence of the improved OA algorithm.

In the similar way stated in Section 5.4, we introduce $\varepsilon > 0$ and consider

$$(mmF_\varepsilon^R) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{x} \in \mathbb{R}^{n-m}} & \boldsymbol{d}T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in X^R \setminus \operatorname{int} \bar{G}_\varepsilon^R, \end{array} \right.$$

where

$$\bar{G}_\varepsilon^R = \{\, \boldsymbol{x} \in X^R \mid \bar{g}^R(\boldsymbol{x}) \geqq \varepsilon \,\}. \tag{5.25}$$

Lemma 5.22 *For any $\varepsilon > 0$, (mmF_ε^R) is regular.*

Proof: We show that

$$X^R \setminus \operatorname{int} \bar{G}^R_\varepsilon = \operatorname{cl}(X^R \setminus \bar{G}^R_\varepsilon), \tag{5.26}$$

holds for any $\varepsilon > 0$.

($\supseteqq$) Since $X^R \setminus \operatorname{int} \bar{G}^R_\varepsilon$ is closed and $X^R \setminus \operatorname{int} \bar{G}^R_\varepsilon \supseteqq X^R \setminus \bar{G}^R_\varepsilon$, we have

$$X^R \setminus \operatorname{int} \bar{G}^R_\varepsilon = \operatorname{cl}(X^R \setminus \operatorname{int} \bar{G}^R_\varepsilon) \supseteqq \operatorname{cl}(X^R \setminus \bar{G}^R_\varepsilon).$$

($\subseteqq$) Let $\boldsymbol{x}$ be an arbitrary point of $X^R \setminus \operatorname{int} \bar{G}^R_\varepsilon$. We show that there is always a point, say x' in $N_\delta(\boldsymbol{x}) \cap (X^R \setminus \bar{G}^R_\varepsilon)$, where $N_\delta(\boldsymbol{x}) = \{\, \boldsymbol{x}' \in \mathbb{R}^{n-m} \mid \|\boldsymbol{x}' - \boldsymbol{x}\| < \delta \,\}$. If $\bar{g}^R(\boldsymbol{x}) > \varepsilon$ then there exists $\gamma > 0$ such that $\bar{g}^R(\boldsymbol{x}') > \varepsilon$ for any point $\boldsymbol{x}' \in N_\gamma(\boldsymbol{x})$ by the continuity of $\bar{g}^R$. This implies $N_\gamma(\boldsymbol{x}) \subseteqq \bar{G}^R_\varepsilon$, and hence $\boldsymbol{x} \in \operatorname{int} \bar{G}^R_\varepsilon$. Therefore the assumption $\boldsymbol{x} \in X^R \setminus \operatorname{int} \bar{G}^R_\varepsilon$ implies that $\boldsymbol{x} \in X^R$ and $\bar{g}^R(\boldsymbol{x}) \leqq \varepsilon$. By Theorem 5.20, we have $\bar{g}^R(\boldsymbol{x}) = g^R(\boldsymbol{x})$. When $\bar{g}^R(\boldsymbol{x}) < \varepsilon$, take $\boldsymbol{x}$ as $\boldsymbol{x}'$. Clearly $\boldsymbol{x}' = \boldsymbol{x} \notin \bar{G}^R_\varepsilon$ and $\boldsymbol{x}' = \boldsymbol{x} \in N_\delta(\boldsymbol{x})$, and we have done. When $g^R(\boldsymbol{x}) = \bar{g}^R(\boldsymbol{x}) = \varepsilon$, there is an optimal solution $\boldsymbol{y}^*$ of $\max\{\, \boldsymbol{e}T\boldsymbol{y} \mid \boldsymbol{y} \in X^R,\ T\boldsymbol{y} \geqq T\boldsymbol{x} \,\}$ such that $\boldsymbol{e}T(\boldsymbol{y}^* - \boldsymbol{x}) = \varepsilon$, and hence $\boldsymbol{y}^* \neq \boldsymbol{x}$. Take λ such that $0 < \lambda < \min\{\, 1,\ \delta / \|\boldsymbol{y}^* - \boldsymbol{x}\| \,\}$ and let $\boldsymbol{x}' = \lambda \boldsymbol{y}^* + (1-\lambda)\boldsymbol{x}$. Since $\|\boldsymbol{x}' - \boldsymbol{x}\| = \lambda \|\boldsymbol{y}^* - \boldsymbol{x}\| < \delta$, we see $\boldsymbol{x}' \in N_\delta(\boldsymbol{x})$. Also we see that $\boldsymbol{x}' \in X^R$ by the convexity of X^R, and hence $g^R(\boldsymbol{x}') = \bar{g}^R(\boldsymbol{x}')$ by applying Theorem 5.20 again. Since $\boldsymbol{x}' \geqq \boldsymbol{x}$ and $\boldsymbol{x}' \neq \boldsymbol{x}$, we have

$$\begin{aligned}
\bar{g}^R(\boldsymbol{x}') &= g^R(\boldsymbol{x}') \\
&= \max\{\, \boldsymbol{e}T\boldsymbol{y} \mid \boldsymbol{y} \in X^R,\ T\boldsymbol{y} \geqq T\boldsymbol{x}' \,\} - \boldsymbol{e}T\boldsymbol{x}' \\
&< \max\{\, \boldsymbol{e}T\boldsymbol{y} \mid \boldsymbol{y} \in X^R,\ T\boldsymbol{y} \geqq T\boldsymbol{x} \,\} - \boldsymbol{e}T\boldsymbol{x} \\
&= \boldsymbol{e}T(\boldsymbol{y}^* - \boldsymbol{x}) = \varepsilon.
\end{aligned}$$

Therefore we see that $\boldsymbol{x}' \notin \bar{G}^R_\varepsilon$. This completes the proof. □

Lemma 5.23 *If $\varepsilon \in (0,1)$ then $\mathbf{0} \in \operatorname{int} \bar{G}^R_\varepsilon$, and $(\mathbf{0}, \boldsymbol{v}) \cap \partial \bar{G}^R_\varepsilon \neq \emptyset$ for any point $\boldsymbol{v}$ such that $\bar{g}^R(\boldsymbol{v}) \leqq 0$.*

Proof: We have $\bar{g}^R(\mathbf{0}) > 0$ since $\mathbf{0} \in \operatorname{int} \bar{G}^R$. Note that $\bar{g}^R(\mathbf{0})$, which coincides with $g^R(\mathbf{0})$, takes an integer value by the integrality property of X^R, and hence $\bar{g}^R(\mathbf{0}) \geqq 1$. Then we have $\bar{g}^R(\mathbf{0}) > \varepsilon$, i.e., $\mathbf{0} \in \operatorname{int} \bar{G}^R_\varepsilon$ for any $\varepsilon \in (0,1)$. The continuity of $\bar{g}^R$ ensures the last assertion. □

Using the same δ_s given by (5.12), we have the following lemma.

Lemma 5.24 *Let $\boldsymbol{x}^*$ and $\boldsymbol{x}^*_\varepsilon$ be optimal solutions of (mmF^R) and (mmF^R_ε), respectively. Then $\boldsymbol{d}T\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}T\boldsymbol{x}^* \leqq \boldsymbol{d}T\boldsymbol{x}^*_\varepsilon + \varepsilon\delta_s$.*

Proof: Since $\boldsymbol{x}^* \in X^R$ and $g^R(\boldsymbol{x}^*) \leqq 0$, $\boldsymbol{x}^*$ is a feasible solution of (mmF^R_ε), and hence $\boldsymbol{d}T\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}T\boldsymbol{x}^*$. Let $\boldsymbol{y}^*_\varepsilon$ be an optimal solution of $\max\{\, \boldsymbol{e}T\boldsymbol{y} \mid \boldsymbol{y} \in X^R,\ T\boldsymbol{y} \geqq T\boldsymbol{x}^*_\varepsilon \,\}$. Clearly $\boldsymbol{y}^*_\varepsilon \in X^R_M$, i.e., $\boldsymbol{y}^*_\varepsilon$ is a feasible solution of (mmF^R), and hence $\boldsymbol{d}T\boldsymbol{x}^* \leqq \boldsymbol{d}T\boldsymbol{y}^*_\varepsilon$. Since $T\boldsymbol{y}^*_\varepsilon - T\boldsymbol{x}^*_\varepsilon \geqq \boldsymbol{0}$ and $\boldsymbol{e}(T\boldsymbol{y}^*_\varepsilon - T\boldsymbol{x}^*_\varepsilon) \leqq \varepsilon$, we see that $\boldsymbol{t}^h\boldsymbol{y}^*_\varepsilon - \boldsymbol{t}^h\boldsymbol{x}^*_\varepsilon \leqq \varepsilon$, where $\boldsymbol{t}^h$ is the hth row of T, for each $h = 1, \ldots, n$. That implies $\boldsymbol{d}(T\boldsymbol{y}^*_\varepsilon - T\boldsymbol{x}^*_\varepsilon) \leqq \varepsilon|\{\, h \mid d_h = +1 \,\}| = \varepsilon\delta_s$, hence $\boldsymbol{d}T\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}T\boldsymbol{x}^* \leqq \boldsymbol{d}T\boldsymbol{y}^*_\varepsilon \leqq \boldsymbol{d}T\boldsymbol{x}^*_\varepsilon + \varepsilon\delta_s$. □

Theorem 5.25 *Let $\boldsymbol{x}^*_\varepsilon$ be an optimal solution of (mmF^R_ε) for some $\varepsilon \in (0,\, 1/\delta_s)$. Then $\lceil \boldsymbol{d}T\boldsymbol{x}^*_\varepsilon \rceil$ coincides with the optimal value of (mmF^R).*

Proof: From Lemma 5.24 we see that $0 \leqq \boldsymbol{d}T\boldsymbol{x}^* - \boldsymbol{d}T\boldsymbol{x}^*_\varepsilon < 1$. This inequality and the integrality of $\boldsymbol{d}T\boldsymbol{x}^*$ gives the assertion. □

Letting

$$X^R(\eta) = \{\, \boldsymbol{x} \in X^R \mid \boldsymbol{d}T\boldsymbol{x} \leqq \eta \,\}, \tag{5.27}$$

for $\eta \in \mathbb{R}$, we have the following theorem.

Theorem 5.26 *Let $\bar{\boldsymbol{x}}_\varepsilon \in X^R \setminus \operatorname{int} \bar{G}^R_\varepsilon$ for some $\varepsilon \in (0,\, 1/\delta_s)$. If $X^R(\lceil \boldsymbol{d}T\bar{\boldsymbol{x}}_\varepsilon - 1 \rceil) \subseteq \bar{G}^R_{\varepsilon'}$ for some $\varepsilon' > 0$ then $\lceil \boldsymbol{d}T\bar{\boldsymbol{x}}_\varepsilon \rceil$ coincides with the optimal value of (mmF^R).*

Proof: Let $\boldsymbol{x}^*$ and $\boldsymbol{x}^*_\varepsilon$ be an optimal solution of (mmF^R) and (mmF^R_ε), respectively. Since $\bar{\boldsymbol{x}}_\varepsilon$ is a feasible solution of (mmF^R_ε), we have $\boldsymbol{d}T\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}T\bar{\boldsymbol{x}}_\varepsilon$. It is also clear that $\boldsymbol{d}T\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}T\boldsymbol{x}^*$. If $\boldsymbol{d}T\boldsymbol{x}^* < \boldsymbol{d}T\bar{\boldsymbol{x}}_\varepsilon$ then we have $\boldsymbol{x}^* \in X(\lceil \boldsymbol{d}T\bar{\boldsymbol{x}}_\varepsilon - 1 \rceil) \subseteq \bar{G}^R_{\varepsilon'}$ since $\boldsymbol{d}T\boldsymbol{x}^*$ is integer, and hence $\bar{g}^R(\boldsymbol{x}^*) \geqq \varepsilon' > 0$, which contradicts that $\bar{g}^R(\boldsymbol{x}^*) = 0$. Then we have $\boldsymbol{d}T\bar{\boldsymbol{x}}_\varepsilon \leqq \boldsymbol{d}T\boldsymbol{x}^*$. Hence by Lemma 5.24 we obtain $\boldsymbol{d}T\boldsymbol{x}^*_\varepsilon \leqq \boldsymbol{d}T\bar{\boldsymbol{x}}_\varepsilon \leqq \boldsymbol{d}T\boldsymbol{x}^* \leqq \boldsymbol{d}T\boldsymbol{x}^*_\varepsilon + \varepsilon\delta_s < \boldsymbol{d}T\boldsymbol{x}^*_\varepsilon + 1$. This completes the proof. □

We describe the improved OA method for (mmF) as follows.

/** improved OA method for (mmF^R) **/

$\langle 0 \rangle$ (initialization) Find an initial feasible vertex $\boldsymbol{w}^0 \in X_M^R \cap X_V^R$ of (mmF). If $N_M^R(\boldsymbol{w}^0) = \emptyset$ then stop ($\boldsymbol{w}^0$ is a unique feasible solution of (mmF^R)). Otherwise, apply the modified LS($\boldsymbol{w}^0$) procedure to obtain a locally optimal vertex $\bar{\boldsymbol{x}}^0 \in X_M^R \cap X_V^R$. Solve $\zeta := \max\{\, \boldsymbol{e}\boldsymbol{x} \mid \boldsymbol{x} \in X^R,\ \boldsymbol{d}T\boldsymbol{x} \leqq \boldsymbol{d}T\bar{\boldsymbol{x}}^0 - 1 \,\}$ and construct an initial polytope $P^0 \supseteqq X(\boldsymbol{d}T\bar{\boldsymbol{x}}^0 - 1)$ by setting $P^0 := \{\, \boldsymbol{x} \in \mathbb{R}^{n-m} \mid \boldsymbol{e}\boldsymbol{x} \leqq \zeta,\ \boldsymbol{d}T\boldsymbol{x} \leqq \boldsymbol{d}T\bar{\boldsymbol{x}}^0 - 1,\ \boldsymbol{x} \geqq \boldsymbol{0} \,\}$. Compute the vertex set P_V^0 of P^0. Set $k := 0$.

$\langle k \rangle$ (iteration k) Find a vertex $\boldsymbol{v}^k \in P_V^k$ such that $\bar{g}^R(\boldsymbol{v}^k) \leqq 0$.

$\langle k1 \rangle$ (termination) If either $\boldsymbol{d}T\bar{\boldsymbol{x}}^k = 0$ or $\bar{g}^R(\boldsymbol{v}) > 0$ for all $\boldsymbol{v} \in P_V^k$ then stop (the optimal value of (mmF^R) is $\lceil \boldsymbol{d}T\bar{\boldsymbol{x}}^k \rceil$). Otherwise, obtain the point $\boldsymbol{x}_\varepsilon^k \in (\boldsymbol{0}, \boldsymbol{v}^k) \cap \partial\, \bar{G}_\varepsilon^R$. Note that Lemma 5.23 ensures that $(\boldsymbol{0}, \boldsymbol{v}^k) \cap \partial\, \bar{G}_\varepsilon^R \neq \emptyset$.

$\langle k2 \rangle$ (update) If $\boldsymbol{x}_\varepsilon^k \in X^R$, obtain the point $\boldsymbol{x}^k \in (\boldsymbol{0}, \boldsymbol{v}^k] \cap \partial\, \bar{G}^R$.

$\langle k2.1 \rangle$ If $\boldsymbol{x}^k \in X^R$, meaning $\boldsymbol{x}^k \in X_M^R$, then obtain the locally optimal vertex $\boldsymbol{z}^k \in X_M^R \cap X_V^R$ by applying the modified LS($\boldsymbol{x}^k$) procedure, and further obtain the point $\boldsymbol{z}_\varepsilon^k \in (\boldsymbol{0}, \boldsymbol{z}^k) \cap \partial\, \bar{G}_\varepsilon^R$. Set $\bar{\boldsymbol{x}}^{k+1} := \operatorname{argmin}\{\, \boldsymbol{d}T\boldsymbol{z}_\varepsilon^k,\ \boldsymbol{d}T\boldsymbol{x}_\varepsilon^k \,\}$ and $P^{k+1} := P^k \cap \{\, \boldsymbol{x} \in \mathbb{R}^{n-m} \mid \boldsymbol{d}T\boldsymbol{x} \leqq \lceil \boldsymbol{d}T\bar{\boldsymbol{x}}^{k+1} - 1 \rceil \,\}$.

$\langle k2.2 \rangle$ If $\boldsymbol{x}^k \notin X^R$, meaning $\boldsymbol{x}^k \notin X_M^R$, then set $\bar{\boldsymbol{x}}^{k+1} := \boldsymbol{x}_\varepsilon^k$ and $P^{k+1} := P^k \cap \{\, \boldsymbol{x} \in \mathbb{R}^{n-m} \mid \boldsymbol{d}T\boldsymbol{x} \leqq \lceil \boldsymbol{d}T\bar{\boldsymbol{x}}^{k+1} - 1 \rceil,\ l(\boldsymbol{x}) \leqq 0 \,\}$, with some affine function $l : \mathbb{R}^{n-m} \to \mathbb{R}$.

$\langle k3 \rangle$ If $\boldsymbol{x}_\varepsilon^k \notin X^R$ then set $\bar{\boldsymbol{x}}^{k+1} := \bar{\boldsymbol{x}}^k$ and $P^{k+1} := P^k \cap \{\, \boldsymbol{x} \in \mathbb{R}^{n-m} \mid l(\boldsymbol{x}) \leqq 0 \,\}$, with some affine function $l : \mathbb{R}^{n-m} \to \mathbb{R}$.

$\langle k4 \rangle$ Compute the vertex set P_V^{k+1} of P^{k+1}. Set $k := k + 1$ and go to $\langle k \rangle$.

Remark 5.27 *The inequality $l(\boldsymbol{x}) \leqq 0$ in Step $\langle k2.2 \rangle$ and Step $\langle k3 \rangle$ is given by one of the inequalities $B^{-1}N\boldsymbol{x} \leqq \boldsymbol{0}$, $-B^{-1}N\boldsymbol{x} \leqq \boldsymbol{c}^B$ and $\boldsymbol{x} \leqq \boldsymbol{c}^N$ not satisfied by the point $\boldsymbol{v}^k$.*

Lemma 5.28 *Let $\boldsymbol{z}^k$ be the locally optimal vertex obtained by applying the LS($\boldsymbol{x}^k$) procedure starting with $\boldsymbol{x}^k$ in Step $\langle k2.1 \rangle$ at iteration k, and suppose $\boldsymbol{d}T\boldsymbol{z}^k > 0$. Then $\boldsymbol{d}T\boldsymbol{z}^{k'} < \boldsymbol{d}T\boldsymbol{z}^k$ for iteration k' such that $k' > k$.*

Proof: It suffices to show that $\boldsymbol{dTz}^{k+1} < \boldsymbol{dTz}^k$. By the construction of P^{k+1} we have $P^{k+1} \subseteqq \{\, \boldsymbol{x} \mid \boldsymbol{dTx} \leqq \lceil \boldsymbol{dT}\bar{\boldsymbol{x}}^{k+1} - 1 \rceil \,\}$. Since $\boldsymbol{x}^{k+1} \in (0, \boldsymbol{v}^{k+1}] \subseteqq P^{k+1}$ and $\boldsymbol{z}^{k+1}$ is obtained by LS($\boldsymbol{x}^{k+1}$), we have

$$\boldsymbol{dTz}^{k+1} \leqq \boldsymbol{dTx}^{k+1} \leqq \lceil \boldsymbol{dT}\bar{\boldsymbol{x}}^{k+1} - 1 \rceil.$$

Since we assume that $\boldsymbol{dTz}^k > 0$, we have $0 < \boldsymbol{dTz}^k_\varepsilon < \boldsymbol{dTz}^k$ by the choice of $\boldsymbol{z}^k_\varepsilon$. Therefore in Step $\langle k2.1 \rangle$

$$\lceil \boldsymbol{dT}\bar{\boldsymbol{x}}^{k+1} - 1 \rceil < \boldsymbol{dT}\bar{\boldsymbol{x}}^{k+1} = \min\{\, \boldsymbol{dTx}^k_\varepsilon, \boldsymbol{dTz}^k_\varepsilon \,\} < \boldsymbol{dTz}^k.$$

Combining the two inequalities yields the desired result. □

Theorem 5.29 *The OA method for* (mmF) *works correctly and terminates after finitely many iterations with the optimal value of* (mmF).

Proof: (correctness) If $N^R_M(\boldsymbol{w}^0) = \emptyset$ at the initialization step, we can conclude from the connectedness of X^R_M that $\boldsymbol{w}^0$ is a unique feasible solution of (mmF^R) and hence solves the problem. When the algorithm terminates in Step $\langle k1 \rangle$, the optimal value of (mmF^R) is equal either to zero by Assumption 5.1 (ii), or to $\lceil \boldsymbol{dT}\bar{\boldsymbol{x}}^k \rceil$ by Theorem 5.12. So the optimal value is obtained whenever the algorithm terminates.

We suppose that the algorithm has not yet terminated at iteration k, i.e., $\boldsymbol{dT}\bar{\boldsymbol{x}}^k > 0$ and $\bar{g}^R(\boldsymbol{v}^k) \leqq 0$, and show that each step of the algorithm can be done. Lemma 5.23 ensures that there are points $\boldsymbol{x}^k_\varepsilon \in (\boldsymbol{0}, \boldsymbol{v}^k) \cap \partial\, \bar{G}^R_\varepsilon$ and $\boldsymbol{z}^k_\varepsilon \in (\boldsymbol{0}, \boldsymbol{z}^k) \cap \partial\, \bar{G}^R_\varepsilon$, in Step $\langle k1 \rangle$ and Step $\langle k2.1 \rangle$, respectively. Since $\boldsymbol{0} \in \operatorname{int} \bar{G}^R$ and $\boldsymbol{v}^k \notin \operatorname{int} \bar{G}^R$, there also exists a point $\boldsymbol{x}^k \in (\boldsymbol{0}, \boldsymbol{v}^k] \cap \partial\, \bar{G}^R$. When $\boldsymbol{x}^k_\varepsilon \notin X^R$, clearly $\boldsymbol{v}^k \notin X^R$, and hence the function $l : \mathbb{R}^{n-m} \to \mathbb{R}$ of Remark 5.27 can be found in Step $\langle k3 \rangle$. To show that the function $l : \mathbb{R}^{n-m} \to \mathbb{R}$ is found in Step $\langle k2.2 \rangle$ we have only to show that $\boldsymbol{v}^k \notin X^R$. Suppose the contrary, i.e., $\boldsymbol{v}^k \in X^R$. By the assumption that $\bar{g}^R(\boldsymbol{v}^k) \leqq 0$ and the fact that $\bar{g}^R(\boldsymbol{x}) \geqq 0$ for all $\boldsymbol{x} \in X^R$, we have $\bar{g}^R(\boldsymbol{v}^k) = 0$, i.e., $\boldsymbol{v}^k \in \partial\, \bar{G}^R$, and hence $\boldsymbol{v}^k \in X^R \setminus \operatorname{int} \bar{G}^R = X^R_M$. This implies $\boldsymbol{x}^k = \boldsymbol{v}^k \in X^R_M$ by the choice of $\boldsymbol{x}^k$, which contradicts that we are currently at iteration $k2.2$. Therefore we have seen that $\boldsymbol{v}^k \notin X^R$ in Step $\langle k2.2 \rangle$.

(finiteness) Suppose that the polytope P^ν at iteration ν meets the condition

$$P^\nu \subseteqq X^R \text{ and } P^\nu \cap X^R_M = \emptyset, \tag{5.28}$$

after updated either in Step $\langle k2\rangle$ or in Step $\langle k3\rangle$, and consider the next iteration. Since $\boldsymbol{v}^\nu$ is chosen from P^ν, we have $\boldsymbol{v}^\nu \in X^R \setminus X^R_M$ and consequently $\bar{g}^R(\boldsymbol{v}^\nu) > 0$. Then the algorithm stops at Step $\langle k1\rangle$. Therefore we have only to prove that (5.28) holds within a finite number of iterations. Note first that both Step $\langle k2.2\rangle$ and Step $\langle k3\rangle$ are done only a finite number of times. By the definition of l, the polytope, say $P^{k'}$, when $m+n$ cuts $l(\boldsymbol{x}) \leqq 0$ have been added to the initial polytope P^0, is contained in X^R. Therefore $\boldsymbol{v}^{k'}$ as well as $\boldsymbol{x}^{k'}_\varepsilon$ lies in X^R, and hence we obtain that $\boldsymbol{x}^{k'} = \boldsymbol{v}^{k'} \in X^R_M$ in the same way as in the former part of this proof. Therefore we come to neither Step $\langle k2.2\rangle$ nor Step $\langle k3\rangle$ after iteration k'. Namely, Step $\langle k2.1\rangle$ followed by Step $\langle k4\rangle$ repeats itself after iteration k'. For iteration k with $k \geqq k'+1$, we have $\boldsymbol{x}^k \in X^R_M$. We then locate $\boldsymbol{z}^k \in X^R_M \cap X^R_V$ by applying the LS($\boldsymbol{x}^k$) procedure and obtain a point $\boldsymbol{z}^k_\varepsilon \in (\boldsymbol{0}, \boldsymbol{z}^k) \cap \partial \bar{G}^R_\varepsilon$. If $\boldsymbol{d}T\boldsymbol{z}^k = 0$ for some $k \geqq k'+1$ then we set $\bar{\boldsymbol{x}}^{k+1} := \boldsymbol{z}^k_\varepsilon$ since $\boldsymbol{d}T\boldsymbol{z}^k_\varepsilon = \boldsymbol{d}T\boldsymbol{z}^k = 0 \leqq \boldsymbol{d}T\boldsymbol{x}^k_\varepsilon$. Then the incumbent value $\boldsymbol{d}T\bar{\boldsymbol{x}}^{k+1}$ becomes zero, and hence the algorithm stops in Step $\langle k1\rangle$ at the next iteration. If $\boldsymbol{d}T\boldsymbol{z}^k > 0$ for all k with $k \geqq k'+1$, we see that $\boldsymbol{d}T\boldsymbol{z}^{k+1} < \boldsymbol{d}T\boldsymbol{z}^k$ for all $k \geqq k'+1$ by Lemma 5.28. Since $|X^R_M \cap X^R_V|$ is finite, we eventually obtain a point $\boldsymbol{z}^{\nu-1} \in X^R_M \cap X^R_V$ such that $\boldsymbol{d}T\boldsymbol{z}^{\nu-1} \leqq \boldsymbol{d}T\boldsymbol{z}$ for all $\boldsymbol{z} \in X^R_M \cap X^R_V$. Also we have $\boldsymbol{d}T\boldsymbol{z}^{\nu-1}_\varepsilon < \boldsymbol{d}T\boldsymbol{z}^{\nu-1}$ by the choice of $\boldsymbol{z}^{\nu-1}_\varepsilon$. The polytope P^ν is then defined as $P^\nu := P^{\nu-1} \cap \{\, \boldsymbol{x} \mid \boldsymbol{d}T\boldsymbol{x} \leqq \lceil \boldsymbol{d}T\bar{\boldsymbol{x}}^\nu - 1 \rceil \,\}$, where $\bar{\boldsymbol{x}}^\nu$ satisfies that $\boldsymbol{d}T\bar{\boldsymbol{x}}^\nu = \min\{\, \boldsymbol{d}T\boldsymbol{x}^{\nu-1}_\varepsilon, \boldsymbol{d}T\boldsymbol{z}^{\nu-1}_\varepsilon \,\} < \boldsymbol{d}T\boldsymbol{z}^{\nu-1}$. This means that $P^\nu \cap (X^R_M \cap X^R_V) = \emptyset$. Since X^R_M is a connected union of several faces of X^R, we see that $\boldsymbol{d}T\boldsymbol{z}^{\nu-1} \leqq \boldsymbol{d}T\boldsymbol{x}$ for all $\boldsymbol{x} \in X^R_M$. Therefore we conclude that $P^\nu \cap X^R_M = \emptyset$. □

5.5.5 Effect of the decomposition by a basic matrix

In this section we would like to emphasize the point that the CS method for (mmF) and the OA method for (mmF) are well suited to the decomposition by a basic matrix. In the improved CS method, we reduce the dimension of the cone to be cut and splitted. Also, in the improved OA method, we enumerate vertices of the polytope P with lower dimension. Furthermore, since X^R has full dimension, it is more likely that the incumbent is more frequently updated. This makes a sharp contrast to the OA method for (mmF). Note that in the OA method for (mmF), there is no possibilities that $\boldsymbol{x}^k_\varepsilon \in (\boldsymbol{0}, \boldsymbol{v}^k) \cap \partial \bar{G}_\varepsilon$ is in X whenever $\boldsymbol{v}^k$ is not contained

in $\{\, \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{0} \,\}$ (See Figure 5.7).

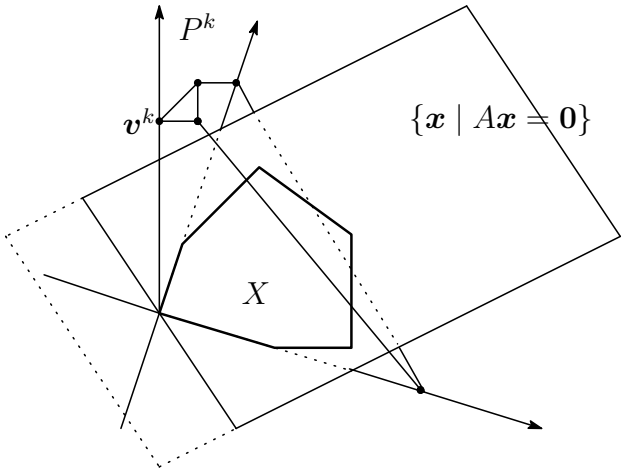


Figure 5.7: An explanation; there is no possiblilities that $\boldsymbol{x}^k_\varepsilon \in X$.

In the OA method, the parameter ε should be as large as possible but chosen from $(0, 1/\delta_s)$. Adding a dummy source that only one arc goes out of, we can make $\delta_s = 1$ and then set ε approximately to one, e.g., $\varepsilon = 0.99$. Namely, defining

$$\bar{A} = \begin{bmatrix} -1 & \boldsymbol{d} \\ \boldsymbol{0} & A \end{bmatrix}, \ \bar{\boldsymbol{c}} = \begin{bmatrix} \sum_{j=1}^n c_j \\ \boldsymbol{c} \end{bmatrix}, \quad \text{and } \bar{\boldsymbol{d}} = \begin{bmatrix} 1 & \boldsymbol{0} \end{bmatrix}, \tag{5.29}$$

and replacing $(A, \boldsymbol{c}, \boldsymbol{d})$ with $(\bar{A}, \bar{\boldsymbol{c}}, \bar{\boldsymbol{d}})$, we have $\delta_s = 1$, and hence we can take 0.99 as ε. This operation would increase the dimension of X, which can be a drawback of the OA method for (mmF), however, it would not change the dimension of X^R.

We should notice that basically this decomposition technique does not reduce the number of variables. Namely the number of variables in (mmF^R) seems to be reduced in comparison with (mmF). However, taking into account slack variables to be introduced, the effect of the decomposition by a basic matrix deteriorates. As for implementation, we usually introduce slack variables and replace inequality constraints by equality constraints. Namely we consider

$$X_{\boldsymbol{s}} = \{\, (\boldsymbol{x}, \boldsymbol{s}) \in \mathbb{R}^{n+n} \mid A\boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{x} + \boldsymbol{s} = \boldsymbol{c},\ (\boldsymbol{x}, \boldsymbol{s}) \geqq \boldsymbol{0} \,\}, \tag{5.30}$$

instead of X of (4.1). Similarly in (mmF^R), we consider

$$X^R_{\boldsymbol{s}} = \{\, (\boldsymbol{x}, \boldsymbol{s}^1, \boldsymbol{s}^2) \in \mathbb{R}^{(n-m)+m+n} \mid B^{-1}N\boldsymbol{x} + \boldsymbol{s}^1 = \boldsymbol{0},\ T\boldsymbol{x} + \boldsymbol{s}^2 = \boldsymbol{c},\ (\boldsymbol{x}, \boldsymbol{s}^1, \boldsymbol{s}^2) \geqq \boldsymbol{0} \,\}, \tag{5.31}$$

instead of X^R of (5.18). Therefore, counting the slack variables, the number of variables in (mmF^R) equals that in (mmF).

We also notice that since the variables of X^R as well as that of X are bounded, the simplex method with bounded variables is available when we solve optimization problems over X^R, e.g., when we evaluate the gap function $\bar{g}^R(\boldsymbol{v})$ for some $\boldsymbol{v} \in P_V^k$.

5.5.6 Modification for non-integral capacity

In this thesis as well as in other studies for (mmF), we assume that each capacity is integer (See Assumption 5.1 (i)). In this subsection we remove this assumption and explain the modification of our algorithms. When a network does not meet Assumption 5.1 (i), the feasible region X does not enjoy the integrality property, which played a crucial role in obtaining the optimal value. Then we need to modify the algorithms so that the algorithms provide a solution $\bar{\boldsymbol{x}} \in X^R$ such that $\boldsymbol{d}T\bar{\boldsymbol{x}} < \boldsymbol{d}T\boldsymbol{x}+\epsilon$ for all $\boldsymbol{x} \in X_M^R$ and for a fixed tolerance $\epsilon > 0$. Fortunately this modification is easily done as follows.

For the improved CS method we just replace Step $\langle k2 \rangle$ with the following:

$\langle k2 \rangle$ (termination) Set $\mathcal{R}' := \{\, K \in \mathcal{R} \mid \beta_K < \boldsymbol{d}T\bar{\boldsymbol{x}} \,\}$. If $\mathcal{R}' = \emptyset$ or $\boldsymbol{d}T\bar{\boldsymbol{x}} - \gamma < \epsilon$ then stop.

For the improved OA method we set $\varepsilon := \epsilon/\delta_s$ to assure that $\boldsymbol{d}T\boldsymbol{x}_\varepsilon^* \leqq \boldsymbol{d}T\boldsymbol{x}^* \leqq \boldsymbol{d}T\boldsymbol{x}_\varepsilon^* + \epsilon$ of Lemma 5.24. We construct the initial polytope P^0 as $P^0 := \{\, \boldsymbol{x} \in \mathbb{R}^{n-m} \mid \boldsymbol{e}\boldsymbol{x} \leqq \zeta,\ \boldsymbol{d}T\boldsymbol{x} \leqq \boldsymbol{d}T\bar{\boldsymbol{x}}^0 - \epsilon,\ \boldsymbol{x} \geqq \boldsymbol{0} \,\}$, where $\zeta := \max\{\, \boldsymbol{e}\boldsymbol{x} \mid \boldsymbol{x} \in X^R,\ \boldsymbol{d}T\boldsymbol{x} \leqq \boldsymbol{d}T\bar{\boldsymbol{x}}^0 - \epsilon \,\}$, so that we have $P^0 \supseteqq X(\boldsymbol{d}T\bar{\boldsymbol{x}}^0 - \epsilon)$. Also when we cut the current polytope P^k by using new incumbent solution $\bar{\boldsymbol{x}}^{k+1}$, we set $P^{k+1} := P^k \cap \{\, \boldsymbol{x} \in \mathbb{R}^{n-m} \mid \boldsymbol{d}T\boldsymbol{x} \leqq \boldsymbol{d}T\bar{\boldsymbol{x}}^{k+1} - \epsilon \,\}$. It is readily seen that this modified algorithm also terminates after finitely many iterations.

5.5.7 Modification for an existence of t-s-path

We assume in this thesis that there is no t-s-path (See Assumption 5.1 (ii)). In this subsection we remove this assumption and explain the modification of our algorithms. Let $\boldsymbol{x}^o$ be a basic optimal solution of $\min\{\, \boldsymbol{d}\boldsymbol{x} \mid \boldsymbol{x} \in X \,\}$. When there is the t-s-path in a network, the optimal value $\boldsymbol{d}\boldsymbol{x}^o$ is negative.

For the improved CS method we just replace $\gamma := 0$ with $\gamma := \boldsymbol{d}\boldsymbol{x}^o$ at the initialization step.

The modification of the improved OA method is also straightforward. At the termination step $\langle k1 \rangle$ we replace $\boldsymbol{d}T\bar{\boldsymbol{x}}^k = 0$ with $\boldsymbol{d}T\bar{\boldsymbol{x}}^k = \gamma$. And we replace $(\boldsymbol{0}, \boldsymbol{v}^k)$, $(\boldsymbol{0}, \boldsymbol{v}^k]$ and $(\boldsymbol{0}, \boldsymbol{z}^k)$ with $(\boldsymbol{x}^o, \boldsymbol{v}^k)$, $(\boldsymbol{x}^o, \boldsymbol{v}^k]$ and $(\boldsymbol{x}^o, \boldsymbol{z}^k)$, respectively, at step $\langle k \rangle$. It is easy to prove the existences of $\boldsymbol{x}^k_\varepsilon \in (\boldsymbol{x}^o, \boldsymbol{v}^k) \cap \partial\, \bar{G}^R_\varepsilon$, $\boldsymbol{x}^k \in (\boldsymbol{x}^o, \boldsymbol{v}^k] \cap \partial\, \bar{G}^R$ and $\boldsymbol{z}^k_\varepsilon \in (\boldsymbol{x}^o, \boldsymbol{z}^k) \cap \partial\, \bar{G}^R_\varepsilon$. The finite convergence of the algorithm is also clear.

5.6 Computational Experiments

We generated ten instances for each combination of m $(= |V \setminus \{s, t\}|)$ and n $(= |E|)$ by using the generation procedure stated in Subsection 5.1.1. We fixed m to 14 and varied n from 20 to 32. Each capacity c_h is randomly chosen from $\{1, 2, \ldots, 10\}$. The program was coded in Octave 2.1.71 with Atlas 3.6.0 and C language compiled by gcc 3.3.3 on cygwin 1.5.14-1, and run on DELL WORKSTATION PWS370 with Intel Pentium 4 (R) 3.20 GHz, 1.00GB of RAM and Windows XP professional version 2002 service pack 2.

5.6.1 Heuristics to locate an initial incumbent

By Theorem 3.5 and the definition of X^R_M, we see that a solution $\boldsymbol{x} \in X^R$ is in X^R_M if and only if there is a vector $\boldsymbol{\lambda} \in \Lambda$ such that $\boldsymbol{x}$ solves $\max\{\, \boldsymbol{\lambda}T\boldsymbol{x} \mid \boldsymbol{x} \in X^R \,\}$, where $\Lambda = \{\, \boldsymbol{\lambda} \in \mathbb{R}_n \mid \boldsymbol{\lambda} \geqq \boldsymbol{e},\ \boldsymbol{\lambda}\boldsymbol{1} = M \,\}$ for a sufficiently large $M > 0$. Shigeno-Takahashi-Yamamoto [48] proved that n^2 suffices for M. By the above observation, in the initialization step of both the improved OA method and the improved CS method, we easily obtain n starting points $\boldsymbol{w}^1, \cdots, \boldsymbol{w}^n$ by solving

$$(SC(\boldsymbol{\lambda}^j)) \qquad \left| \begin{array}{ll} \max\limits_{\boldsymbol{x} \in \mathbb{R}^{n-m}} & \boldsymbol{\lambda}^j T \boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in X^R, \end{array} \right.$$

where $\boldsymbol{\lambda}^j = (n^2 - n)\boldsymbol{e}^j + \boldsymbol{e}$ for $j = 1, \ldots, n$. Note that $\boldsymbol{\lambda}^j$ is the jth vertex of Λ. Applying the modified LS($\boldsymbol{w}^j$) procedure, we obtain n locally optimal solutions. We then set the incumbent to the best solution among the above solutions.

When we obtain a solution $\bar{\boldsymbol{x}} \in X^R_M$ such that $\boldsymbol{d}T\bar{\boldsymbol{x}} = 0$ in the initialization step,

we can immediately stop the algorithm, and hence we conclude that the iteration of the algorithm is zero for this problem.

5.6.2 Result of the improved CS method for (mmF)

We show the numerical result of the improved CS method for $n = 20$ in Table 5.1. The notations used in the table are as follows:

name	:	problem name
maxF	:	maximum flow value
mmF	:	minimum maximal flow value
itr	:	number of iterations needed
#prob	:	number of subproblems generated by $\boldsymbol{\omega}$-subdivision
dep	:	maximum depth of enumeration tree
LStime	:	CPU time in second to obtain the best locally optimal solution
time	:	total CPU time in second

Table 5.1: Result of the improved CS method $(m = 14, n = 20)$

name	*maxF*	*mmF*	*itr*	*#prob*	*dep*	*LStime*	*time*
p20_0	8	5	60	168	9	9.734	55.594
p20_1	1	1	200	557	8	4.578	212.219
p20_2	1	0	0	0	0	1.156	1.156
p20_3	4	3	288	957	9	5.344	291.063
p20_4	3	1	75	229	8	8.734	79.813
p20_5	8	8	858	2423	11	4.875	1086.766
p20_6	2	2	10	38	4	2.515	17.390
p20_7	4	4	805	2170	14	4.734	875.844
p20_8	1	0	0	0	0	2.702	2.702
p20_9	7	6	442	1225	13	5.157	460.813

Note that for test problems p20_2 and p20_8 we obtained a solution $\bar{\boldsymbol{x}} \in X_M^R$ such that $\boldsymbol{d}T\bar{\boldsymbol{x}} = 0$ in the initialization step. For every test problem it turned out we

had already obtained the optimal value in the initialization step, and hence the first incumbent was not updated. However, we need to solve many subproblems to prove optimality. While problems in Table 5.1 are relatively easy, long computational time was needed. Since computation is estimated to become more costly as n grows, we regrettably conclude that the improved CS method is not satisfactory. Further improvements of the algorithm, especially improvement of the concavity cut to strengthen the lower bound, would be required.

5.6.3 Result of the improved OA method for (mmF)

We show the numerical result of the improved OA method for $n = 20, \cdots, 32$ in Table 5.2-Table 5.6. The notations there are as follows:

$name$	:	problem name
$\#X_V^R$	:	number of vertices of X^R
$\#X_M^R$	:	number of maximal flows in X_V^R
$maxF$	:	maximum flow value
mmF	:	minimum maximal flow value
itr	:	number of iterations needed
$\#\bar{g}^R(\boldsymbol{v})$	:	number of evaluations of the function $\bar{g}^R$
$Vtime$	:	CPU time in second to calculate the vertex set of P^k
$time$	:	total CPU time in second

One of the most simple and primitive method for solving (mmF^R) is an application of vertex enumeration method. Namely to solve (mmF^R), we enumerate all vertices of X^R, check if the vertex is a maximal flow by evaluating the gap function and compare their flow values. Obviously the computational time needed is proportional to the number of vertices of X^R. On the other hand, the computational time of the improved OA method is also needed in proportion to the number of evaluations of the function $\bar{g}^R(\boldsymbol{v})$ for $\boldsymbol{v} \in P_V^k$. Then we can say that the improved OA method works efficiently when $\#\bar{g}^R(\boldsymbol{v})$ is much smaller than $\#X_V^R$. We observe from Table 5.2-Table 5.6 that the improved OA method surpasses an application of vertex enumeration method in computational time, especially as the number n of

arcs grows. Take the test problem p28_8 for example. For this test problem only 172 evaluations of the function $\bar{g}^R(\boldsymbol{v})$ ensured the optimality while X has about 5000 vertices. We also observe that for every test problems we have already obtained the optimal value in the initialization step, and hence the first incumbent was not updated. Therefore we can say that our local search procedure is considerably good. Furthermore we observe that the minimum maximal flow value is likely to be zero as n grows.

Table 5.2: Result of the improved OA method ($m = 14, n = 20, 21, 22$)

name	$\#X_V^R$	$\#X_M^R$	*maxF*	*mmF*	*itr*	$\#\bar{g}^R(\boldsymbol{v})$	*Vtime*	*time*
p20_0	32	11	8	5	8	41	0.125	19.265
p20_1	48	4	1	1	8	29	0.032	12.437
p20_2	32	7	1	0	0	0	0	0.485
p20_3	44	8	4	3	7	42	0.046	15.281
p20_4	55	8	3	1	8	39	0.016	18.343
p20_5	42	14	8	8	7	56	0.032	16.313
p20_6	20	4	2	2	7	29	0.000	9.952
p20_7	31	12	4	4	7	43	0.015	13.641
p20_8	12	5	1	0	0	0	0	0.468
p20_9	39	16	7	6	7	44	0.048	14.359
p21_0	152	15	8	7	12	191	0.125	55.344
p21_1	38	8	2	0	0	0	0	0.484
p21_2	40	10	2	0	0	0	0	0.687
p21_3	124	12	6	6	12	168	0.032	42.875
p21_4	96	4	4	2	7	47	0.063	24.891
p21_5	60	6	16	13	11	84	0.015	30.781
p21_6	26	7	3	1	4	7	0.032	15.079
p21_7	64	8	4	2	6	42	0.047	35.703
p21_8	72	16	3	3	7	46	0.030	17.563
p21_9	64	6	5	5	8	78	0.048	21.782
p22_0	22	19	2	0	0	0	0	0.703
p22_1	13	12	1	0	0	0	0	0.515
p22_2	324	5	8	7	14	448	0.139	122.343
p22_3	21	13	1	1	4	19	0.000	9.657
p22_4	122	45	7	4	10	121	0.031	35.500
p22_5	212	30	7	4	12	212	0.092	58.329
p22_6	35	11	2	1	6	19	0.047	13.391
p22_7	44	19	2	2	6	39	0.000	14.859
p22_8	212	38	10	10	14	227	0.016	57.250
p22_9	23	12	2	2	4	20	0.016	11.344

Table 5.3: Result of the improved OA method ($m = 14, n = 23, 24, 25$)

name	$\#X_V^R$	$\#X_M^R$	*maxF*	*mmF*	*itr*	$\#\bar{g}^R(\boldsymbol{v})$	*Vtime*	*time*
p23_0	224	12	1	0	0	0	0	0.233
p23_1	59	18	1	0	0	0	0	0.531
p23_2	152	32	5	0	0	0	0	8.673
p23_3	80	12	7	7	11	140	0.015	42.563
p23_4	200	32	3	2	10	114	0.046	40.750
p23_5	214	37	2	0	0	0	0	0.562
p23_6	32	7	1	0	0	0	0	0.563
p23_7	124	30	7	7	10	125	0.032	35.499
p23_8	458	81	5	4	16	478	0.063	124.328
p23_9	311	70	4	3	11	210	0.078	64.515
p24_0	39	8	2	2	9	53	0.016	25.922
p24_1	38	10	1	0	0	0	0	0.625
p24_2	60	20	5	2	7	40	0.048	42.610
p24_3	348	131	4	0	0	0	0	0.718
p24_4	120	16	4	2	9	77	0.062	32.626
p24_5	37	16	4	1	8	37	0.032	34.047
p24_6	326	80	10	6	10	201	0.063	59.094
p24_7	84	28	4	3	8	134	0.000	47.094
p24_8	68	20	4	0	0	0	0	1.047
p24_9	380	16	2	0	0	0	0	0.750
p25_0	536	148	9	4	10	227	0.063	73.891
p25_1	439	180	4	0	0	0	0	1.422
p25_2	384	60	4	2	12	193	0.126	69.203
p25_3	2250	258	7	4	12	758	0.171	220.719
p25_4	1024	36	3	1	14	260	0.108	112.750
p25_5	816	291	8	1	12	70	0.063	60.484
p25_6	246	92	6	6	11	353	0.077	95.234
p25_7	426	68	3	2	12	253	0.110	81.328
p25_8	512	18	5	5	8	521	0.077	128.171
p25_9	1232	259	7	5	16	842	0.155	226.000

Table 5.4: Result of the improved OA method ($m = 14, n = 26, 27, 28$)

name	$\#X_V^R$	$\#X_M^R$	*maxF*	*mmF*	*itr*	$\#\bar{g}^R(\boldsymbol{v})$	*Vtime*	*time*
p26_0	556	44	5	4	14	512	0.079	152.610
p26_1	352	111	1	0	0	0	0	0.656
p26_2	44	21	5	4	9	52	0.061	29.126
p26_3	330	128	2	0	0	0	0	0.265
p26_4	380	60	6	3	10	168	0.141	65.954
p26_5	1178	216	2	0	0	0	0	0.640
p26_6	192	14	7	6	13	268	0.109	94.813
p26_7	1052	171	15	10	17	1211	0.469	377.844
p26_8	2128	94	5	5	16	2867	0.483	728.000
p26_9	192	51	2	0	0	0	0	0.860
p27_0	1812	70	7	2	12	501	0.141	203.188
p27_1	2844	893	4	0	0	0	0	1.359
p27_2	584	71	4	2	13	445	0.172	157.687
p27_3	6603	574	5	0	0	0	0	18.687
p27_4	640	348	4	0	0	0	0	1.046
p27_5	560	464	2	0	0	0	0	0.937
p27_6	120	36	4	2	9	147	0.094	58.828
p27_7	56	13	1	0	0	0	0	0.251
p27_8	166	32	6	5	11	188	0.079	91.297
p27_9	120	94	3	0	0	0	0	0.609
p28_0	2309	1250	2	2	15	1483	0.438	435.360
p28_1	970	407	6	0	0	0	0	1.235
p28_2	352	99	1	0	0	0	0	1.001
p28_3	392	70	4	0	0	0	0	2.609
p28_4	2056	341	2	2	16	1314	0.377	414.859
p28_5	5532	1198	12	10	20	5589	1.905	1668.359
p28_6	96	33	3	0	0	0	0	0.218
p28_7	1106	477	2	0	0	0	0	0.797
p28_8	5934	401	4	1	15	172	0.077	94.078
p28_9	12138	2024	14	14	28	12588	106.534	3847.781

Table 5.5: Result of the improved OA method ($m = 14, n = 29, 30, 31$)

name	$\#X_V^R$	$\#X_M^R$	*maxF*	*mmF*	*itr*	$\#\bar{g}^R(\boldsymbol{v})$	*Vtime*	*time*
p29_0	21328	2856	2	0	0	0	0	1.187
p29_1	36	25	1	0	0	0	0	0.813
p29_2	88	24	3	1	11	20	0.110	39.563
p29_3	1388	379	5	2	15	570	0.347	231.048
p29_4	1948	281	13	9	18	1976	0.908	673.625
p29_5	193	141	4	3	12	107	0.109	68.046
p29_6	2412	308	6	1	15	588	0.203	232.250
p29_7	584	88	8	1	10	57	0.064	59.844
p29_8	3013	398	5	0	0	0	0	1.141
p29_9	424	90	11	8	13	421	0.185	192.155
p30_0	7126	1113	1	1	23	1105	0.472	448.017
p30_1	40	19	1	0	0	0	0	0.266
p30_2	29682	1774	13	7	27	26838	111.998	9696.844
p30_3	2288	434	4	0	0	0	0	1.937
p30_4	86	32	2	1	10	34	0.094	29.063
p30_5	2751	1249	2	0	0	0	0	1.140
p30_6	22336	3608	6	6	24	22029	42.922	7679.844
p30_7	17940	858	5	0	0	0	0	2.063
p30_8	1124	178	3	0	0	0	0	0.250
p30_9	476	215	2	0	0	0	0	1.406
p31_0	17330	1837	6	4	20	17459	28.453	6305.281
p31_1	292	93	3	0	0	0	0	1.453
p31_2	648	56	3	0	0	0	0	1.999
p31_3	5030	872	3	0	0	0	0	0.828
p31_4	4767	290	1	1	22	2114	1.174	768.328
p31_5	352	56	4	0	0	0	0	1.547
p31_6	65178	2254	3	0	0	0	0	0.298
p31_7	5998	2244	2	0	0	0	0	1.234
p31_8	660	66	4	0	0	0	0	13.188
p31_9	3384	144	14	6	19	3066	1.403	1184.359

Table 5.6: Result of the improved OA method ($m = 14, n = 32$)

name	$\#X_V^R$	$\#X_M^R$	*maxF*	*mmF*	*itr*	$\#\bar{g}^R(\boldsymbol{v})$	*Vtime*	*time*
p32_0	8172	1289	2	0	0	0	0	0.281
p32_1	4901	545	2	0	0	0	0	3.109
p32_2	275432	11332	5	2	28	162125	13180.059	103012.640
p32_3	52910	8047	5	0	0	0	0	3.422
p32_4	84799	10442	9	0	0	0	0	17.688
p32_5	28314	5272	10	4	21	15638	34.827	8805.343
p32_6	101027	6630	5	0	0	0	0	1.312
p32_7	8682	1235	5	1	18	814	0.580	636.437
p32_8	139377	10678	7	0	0	0	0	2.110
p32_9	3544	404	7	7	17	4550	2.970	2476.688

5.6.4 Result of the method in Section 4.2

To compare the improved OA method and the method in Section 4.2, we show the numerical result for $n = 20, \cdots, 22$ in Table 5.7. The notations used in the table are as follows:

name	:	problem name
$\#X_V^R$	:	$\#X_V^R$ in Table 5.2
$\#\bar{g}^R(\boldsymbol{v})$	:	$\#\bar{g}^R(\boldsymbol{v})$ in Table 5.2
maxF	:	maximum flow value
mmF	:	minimum maximal flow value
itr	:	number of iterations needed
#up	:	number of update
$\#\tau_\alpha(\boldsymbol{\lambda})$	:	number of evaluations of the function τ_α
Vtime	:	CPU time in second to calculate the vertex set of epi σ_W
time	:	total CPU time in second

We observe from Table 5.7 that the method in Section 4.2 requires more eval-

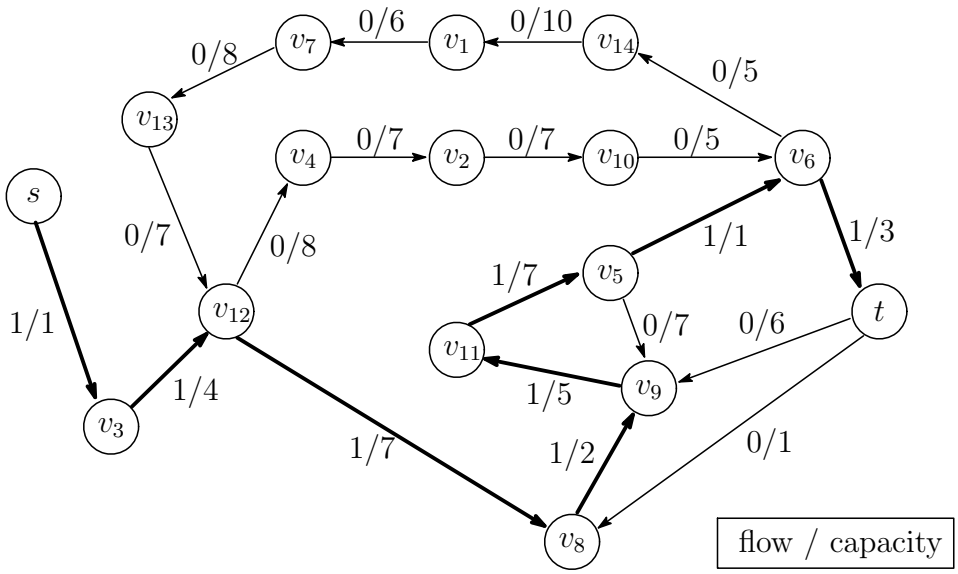
uations of the function τ_α to check the optimality than the improved OA method requires the evaluation of $\bar{g}^R$. Since the computational time is needed in proportion to $\#\tau_\alpha(\boldsymbol{\lambda})$, we can say that the improved OA method surpasses the method in Section 4.2. Since the dimension of $\operatorname{epi}\sigma_W$ is $n+1$, the computational time begins to grow rapidly when n is about 22. We also observe that there are some test problems for which we have not obtained the optimal value in the initialization step, and hence the first incumbent was updated (See rows of p20_0, p20_3, p20_9 and p22_4 in Table 5.7). Such differences between the improved methods and the method in Section 4.2 are due to the modification of the local search procedure. This is also a merit of the decomposition technique by the basic matrix.

5.6.5 Figures of test problems p20_2 and p20_5

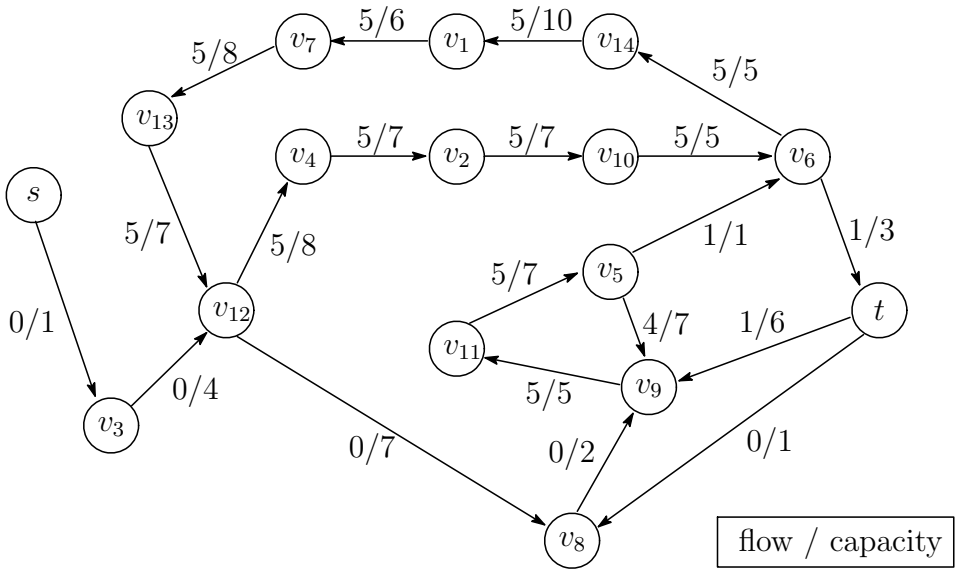
From the numerical results we observe that the maximum flow value and the minimum maximal flow value are different for a good many instances. However it is not easy to say if these values are different for a given network. Take problems p20_2 and p20_5 for examples. For p20_2 these values are different: the minimum maximal flow value is 0 while the maximum flow value is 1. On the other hand, for p20_5 both values are 8. We show examples of a maximal flow and a minimum maximal flow for p20_2 in Figure 5.8 and for p20_5 in Figure 5.9, respectively.

Table 5.7: Result of the method in Section 4.2 ($m = 14, n = 20, 21, 22$)

name	$\#X_V$	$\#\bar{g}^R(\boldsymbol{v})$	*maxF*	*mmF*	*itr*	*#up*	$\#\tau_\alpha(\boldsymbol{\lambda})$	*Vtime*	*time*
p20_0	32	41	8	5	11	1	760	5.906	49.436
p20_1	48	29	1	1	2	0	47	0.000	6.406
p20_2	32	0	1	0	0	0	0	0	0.547
p20_3	44	42	4	3	5	1	93	0.079	9.844
p20_4	55	39	3	1	5	0	360	0.235	25.906
p20_5	42	56	8	8	5	0	137	0.234	11.469
p20_6	20	29	2	2	4	0	158	0.110	9.579
p20_7	31	43	4	4	9	0	619	4.953	42.203
p20_8	12	0	1	0	0	0	0	0	0.532
p20_9	39	44	7	6	13	1	1085	74.111	133.298
p21_0	152	191	8	7	7	0	235	4.374	33.062
p21_1	38	0	2	0	0	0	0	0	0.188
p21_2	40	0	2	0	0	0	0	0	0.688
p21_3	124	168	6	6	5	0	131	0.139	15.796
p21_4	96	47	4	2	2	0	65	0.016	13.859
p21_5	60	84	16	13	3	0	81	0.061	16.109
p21_6	26	7	3	1	1	0	44	0.000	15.890
p21_7	64	42	4	2	6	0	116	0.187	32.719
p21_8	72	46	3	3	8	0	959	1.888	58.546
p21_9	64	78	5	5	4	0	144	0.094	12.298
p22_0	22	0	2	0	0	0	0	0	0.577
p22_1	13	0	1	0	0	0	0	0	0.438
p22_2	324	448	8	7	2	0	61	0.016	27.235
p22_3	21	19	1	1	5	0	322	0.234	19.890
p22_4	122	121	7	4	46	1	4448	7438.077	7686.390
p22_5	212	212	7	4	12	0	4208	1160.486	1411.188
p22_6	35	19	2	1	3	0	82	0.016	9.484
p22_7	44	39	2	2	6	0	1700	24.733	126.093
p22_8	212	227	10	10	17	0	10918	3141.468	3791.937
p22_9	23	20	2	2	6	0	2112	3.533	103.297

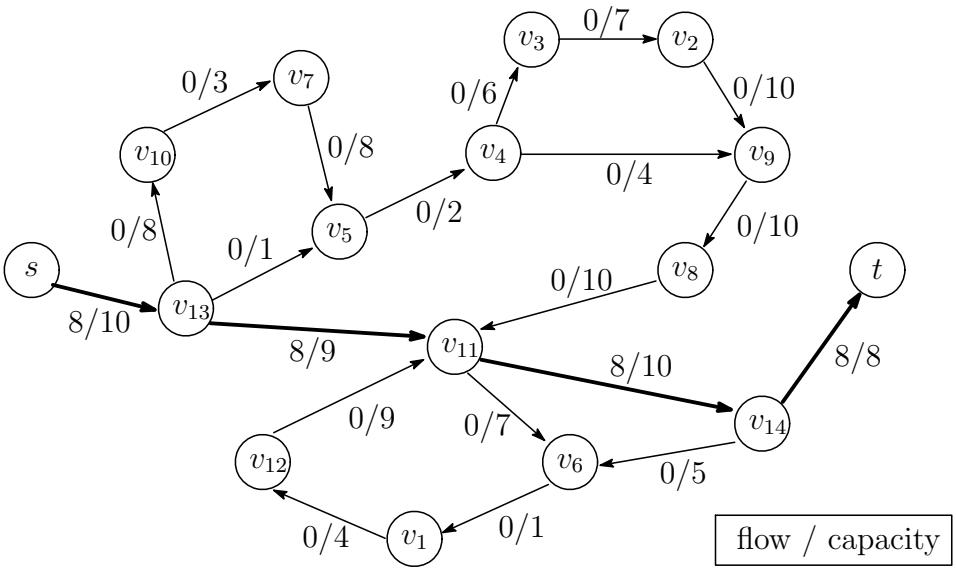


(a) maximum flow with value 1

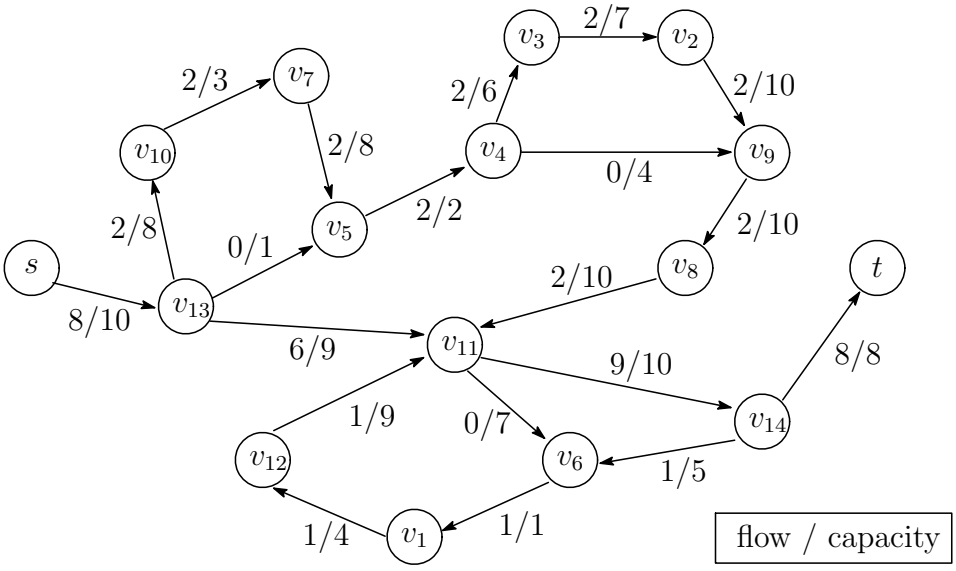


(a) minimum maximal flow with value 0

Figure 5.8: maximum flow vs. minimum maximal flow for p20_2



(a) maximum flow with value 8



(a) minimum maximal flow with value 8

Figure 5.9: maximum flow vs. minimum maximal flow for p20_5

Chapter 6

Conclusion and Further Works

In this chapter we sum up our study and list some possible further works.

6.1 Conclusion

We considered the minimum maximal flow problem which we think is one of typical and most difficult optimization problems over the efficient set. We used the integrality property of network flow problems and developed two algorithms: the cut-and-split method and the outer approximation method based on the D.C. optimization algorithms.

Exploiting the flow conservation equations of the problem, we further proposed an improvement on the algorithms by reducing the number of variables. Concerning the convergence of the algorithms, we showed that the outer approximation method terminates after finitely many iterations with the optimal value. We also extended the gap function to the whole space. Finally we carried out the computational experiment. We observed from the numerical results that the improved OA method works efficiently for many problems and surpasses both an application of vertex enumeration method and the method in Section 4.2 in computational time, especially as the number of variables grows. We also observed that our heuristics using the modified local search procedure provides a pretty good initial incumbent.

6.2 Further Works

More computational experiments should be carried out to verify the efficiency of the algorithms we proposed in this thesis. In many problems formulated as (P_E) the criterion matrix C has quite a small number p of rows. Some sophisticated algorithms for (P_E) take advantage of this property. However, in (mmF) the number p is equal to the number of arcs, i.e., $p = n$. Therefore it is likely that a primitive method surpasses some sophisticated algorithms. A well-organized comparative study of algorithms for (mmF) is necessary.

6.2.1 The way of finding an optimal solution

As we have shown in Section 5.4, the OA method provides the optimal value but may fail to provide an optimal solution of (mmF). Finding an optimal solution is still a hard task even when its value is at hand, however, the following lemma affords a clue to the way of finding an optimal solution.

Lemma 6.1 *Let $\varepsilon \in (0,1)$, $\boldsymbol{x}^*_\varepsilon$ be an ε-optimal solution of (mmF) and*

$$\Delta_\varepsilon = \{\, \boldsymbol{\xi} \in \mathbb{R}^n \mid A\boldsymbol{\xi} = \boldsymbol{0},\ \boldsymbol{\xi} \geqq \boldsymbol{0},\ \boldsymbol{e\xi} \leqq \varepsilon \,\}. \tag{6.1}$$

If $\boldsymbol{x}^*_\varepsilon + \bar{\boldsymbol{\xi}}$ is an integer vector for some $\bar{\boldsymbol{\xi}} \in \Delta_\varepsilon$ then $\boldsymbol{x}^*_\varepsilon + \bar{\boldsymbol{\xi}}$ is an optimal solution of (mmF).

Proof: (feasibility) Let $\boldsymbol{x}^* = \boldsymbol{x}^*_\varepsilon + \bar{\boldsymbol{\xi}}$ and $\boldsymbol{y}^*$ be an optimal solution of $\max\{\, \boldsymbol{ey} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} \geqq \boldsymbol{x}^* \,\}$. Note that

$$\boldsymbol{ex}^* \text{ is integer,} \tag{6.2}$$

$$\boldsymbol{ex}^*_\varepsilon \leqq \boldsymbol{ex}^* \leqq \boldsymbol{ey}^*, \tag{6.3}$$

and also

$$\boldsymbol{ey}^* \text{ is integer,} \tag{6.4}$$

since $X \cap \{\, \boldsymbol{y} \mid \boldsymbol{y} \geqq \boldsymbol{x}^* \,\}$ inherits the integrality property of X.

Suppose we have the inequality

$$\boldsymbol{ey}^* < \boldsymbol{ex}^*_\varepsilon + 1. \tag{6.5}$$

Then by (6.3) and (6.5) together with the integrality of $\boldsymbol{ex}^*$ and $\boldsymbol{ey}^*$ we see that $\boldsymbol{ex}^* = \boldsymbol{ey}^*$. Hence $\bar{g}(\boldsymbol{x}^*) = \boldsymbol{ey}^* - \boldsymbol{ex}^* = 0$, meaning that $\boldsymbol{x}^* \in X_M$.

The inequality (6.5) is seen as follows. Let $\boldsymbol{y}^*_\varepsilon$ be an optimal solution of $\max\{\, \boldsymbol{ey} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} \geqq \boldsymbol{x}^*_\varepsilon \,\}$, and let $\boldsymbol{\xi}^* = \boldsymbol{y}^*_\varepsilon - \boldsymbol{x}^*_\varepsilon$. We see that $A\boldsymbol{\xi}^* = A\boldsymbol{y}^*_\varepsilon - A\boldsymbol{x}^*_\varepsilon = \boldsymbol{0}$, $\boldsymbol{\xi}^* \geqq \boldsymbol{0}$ and $\boldsymbol{e\xi}^* = \boldsymbol{e}(\boldsymbol{y}^*_\varepsilon - \boldsymbol{x}^*_\varepsilon) = g(\boldsymbol{x}^*_\varepsilon) \leqq \varepsilon$, and hence $\boldsymbol{\xi}^* \in \Delta_\varepsilon$. Then $\boldsymbol{ey}^*_\varepsilon = \boldsymbol{e}(\boldsymbol{x}^*_\varepsilon + \boldsymbol{\xi}^*) \leqq \boldsymbol{ex}^*_\varepsilon + \varepsilon < \boldsymbol{ex}^*_\varepsilon + 1$. The point $\boldsymbol{y}^*$ is a feasible solution of $\max\{\, \boldsymbol{ey} \mid \boldsymbol{y} \in X,\ \boldsymbol{y} \geqq \boldsymbol{x}^*_\varepsilon \,\}$, since $\boldsymbol{y}^* \in X$ and $\boldsymbol{y}^* \geqq \boldsymbol{x}^* = \boldsymbol{x}^*_\varepsilon + \bar{\boldsymbol{\xi}} \geqq \boldsymbol{x}^*_\varepsilon$. Then we see that $\boldsymbol{e}(\boldsymbol{y}^*_\varepsilon - \boldsymbol{y}^*) \geqq 0$, and hence $\boldsymbol{ey}^* \leqq \boldsymbol{ey}^*_\varepsilon < \boldsymbol{ex}^*_\varepsilon + 1$.

(optimality) We show that $\boldsymbol{x}^*$ solves (mmF). Clearly, $\boldsymbol{d\bar{\xi}} \leqq \boldsymbol{e\bar{\xi}}$ since $\boldsymbol{d} \leqq \boldsymbol{e}$ and $\bar{\boldsymbol{\xi}} \geqq \boldsymbol{0}$. For any $\boldsymbol{v} \in X_M \cap X_V$, we see that $g(\boldsymbol{v}) \leqq \varepsilon$, and $\boldsymbol{v}$ is an integer vector by the integrality property of X. Since $\boldsymbol{x}^*_\varepsilon = \boldsymbol{x}^* - \bar{\boldsymbol{\xi}}$ is an optimal solution of (mmF_ε), we have $\boldsymbol{dx}^*_\varepsilon \leqq \boldsymbol{dx}$ for all $\boldsymbol{x} \in X$ such that $g(\boldsymbol{x}) \leqq \varepsilon$, and hence $\boldsymbol{dx}^*_\varepsilon \leqq \boldsymbol{dv}$ for all $\boldsymbol{v} \in X_M \cap X_V$. Then we see that $\boldsymbol{dx}^* = \boldsymbol{dx}^*_\varepsilon + \boldsymbol{d\bar{\xi}} \leqq \boldsymbol{dv} + \boldsymbol{e\bar{\xi}} < \boldsymbol{dv} + 1$. Since both $\boldsymbol{x}^*$ and $\boldsymbol{v}$ are integer vectors, we have $\boldsymbol{dx}^* \leqq \boldsymbol{dv}$ for all $\boldsymbol{v} \in X_M \cap X_V$. □

6.2.2 Improvement of the outer approximation method of the parameter set

The fact that T of (5.19) has n rows, which corresponds the criteria, is also a drawback of the variable reduction technique in Section 5.5. When we apply the outer approximation algorithm stated in Section 4.2 to (mmF^R), we consider

$$\left|\begin{array}{ll} \min\limits_{\boldsymbol{x} \in \mathbb{R}^{n-m}} & \boldsymbol{dTx} \\ \text{s.t.} & (\boldsymbol{x}, \boldsymbol{\lambda}) \in X^R \times \Lambda \\ & \boldsymbol{\lambda Tx} \geqq \boldsymbol{\lambda Tx'} \text{ for all } \boldsymbol{x'} \in X^R. \end{array}\right.$$

Note that Λ has only n vertices, which we will denote by $\boldsymbol{\lambda}^1, \cdots, \boldsymbol{\lambda}^n$. This problem is equivalent to

$$\left|\begin{array}{ll} \min\limits_{\boldsymbol{x} \in \mathbb{R}^{n-m}} & \boldsymbol{dTx} \\ \text{s.t.} & (\boldsymbol{x}, \boldsymbol{t}) \in X^R \times \Lambda(T) \\ & \boldsymbol{tx} \geqq \boldsymbol{tx'} \text{ for all } \boldsymbol{x'} \in X^R, \end{array}\right.$$

where

$$\Lambda(T) = \{\, \boldsymbol{\lambda} T \in \mathbb{R}_{n-m} \mid \boldsymbol{\lambda} \in \Lambda \,\} = \mathrm{conv}\{\boldsymbol{\lambda}^1 T, \cdots, \boldsymbol{\lambda}^n T\}.$$

By the above observation, if we find the minimal set of inequality constraints defining $\Lambda(T)$, we can reduce the dimension of the parameter set in Section 4.2. However, it is difficult to find such set of inequality constraints.

6.2.3 The way of constructing the submatrix T of (5.19)

We do not know at present which submatrix is theoretically best as the basic matrix of the decomposition. The matrix T of (5.19) is

$$T = \begin{bmatrix} -B^{-1}N \\ I \end{bmatrix},$$

and its size is $n \times (n-m)$. If a row of $-B^{-1}N$ is nonnegative, it is obviously a nonnegative linear combination of rows of I. Hence, it can be deleted from the beginning. Let T' be the matrix after deleting all such rows from T. Then the downsized problem

$$\left|\begin{array}{ll} \min\limits_{\boldsymbol{x} \in \mathbb{R}^{n-m}} & \boldsymbol{d}T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{x} \in X^R \\ & \nexists \boldsymbol{y}^N \in X^R : T'\boldsymbol{y}^N \geqq T'\boldsymbol{x}^N \text{ and } T'\boldsymbol{y}^N \neq T'\boldsymbol{x}^N \end{array}\right.$$

is equivalent to (mmF^R). Therefore, it is desirable to find the basic matrix B such that $-B^{-1}N$ has as many nonnegative rows as possible. The efficient algorithm for finding such a basic matrix, however, awaits future studies.

Appendix

A Uncontrollable Flow

The concept of *uncontrollable flow*, raised by Iri [29–31], is closely related to but slightly different from the maximal flow.

Given the directed network $\mathcal{N} = (V, E, \boldsymbol{c})$ in Section 4.1, let

$$\mathrm{supp}(\boldsymbol{x}) = \{\, h \in E \mid x_h \neq 0 \,\}, \tag{6.6}$$

and

$$\mathcal{F} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \begin{array}{l} A\boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{x} \geqq \boldsymbol{0},\ f_v(\boldsymbol{x}) \geqq 0,\ \text{and} \\ \mathrm{supp}(\boldsymbol{x}) \text{ is an elementary } s\text{-}t\text{-path} \end{array} \right\}. \tag{6.7}$$

A flow $\boldsymbol{x}$ is said to be an *uncontrollable flow*, abbreviated to *u-flow*, if $\boldsymbol{x}$ is a non-negative combination of flows in $\mathcal{F}$. Note that $\boldsymbol{x} = \boldsymbol{0}$ is a *trivial* u-flow. An u-flow $\boldsymbol{x}$ is said to be *feasible* if $\boldsymbol{x} \leqq \boldsymbol{c}$. And a feasible u-flow $\boldsymbol{x}$ is said to be *maximal* if there is no u-flow $\boldsymbol{x}'$ such that $\boldsymbol{x}' \neq \boldsymbol{0}$ and $\boldsymbol{x} + \boldsymbol{x}'$ is feasible. Figure 6.1 shows an example of a maximal u-flow. Note that the flow in this figure is not maximal flow.

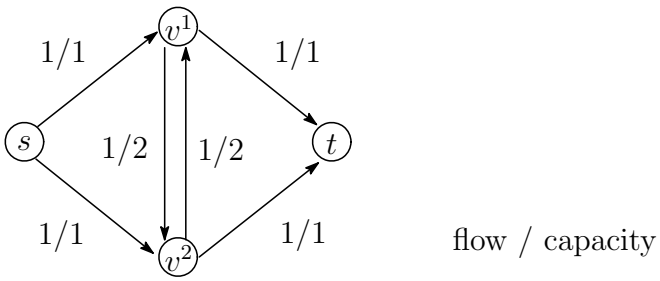


Figure 6.1: An example of maximal u-flow

B Dual Simplex Method

We review the dual simplex method. The aims of Appendix B are to solve

$$\left|\begin{array}{ll} \min\limits_{\boldsymbol{x}\in\mathbb{R}^n} & \boldsymbol{dx} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geqq \boldsymbol{0} \\ & \boldsymbol{e}^j\boldsymbol{x} \leqq 0 \qquad \text{for } j \in I^z(\boldsymbol{w}), \end{array}\right.$$

where $I^z(\boldsymbol{w}) = \{\, j \mid w_j = 0 \,\}$ for a given $\boldsymbol{w} \in \mathbb{R}^n$, and to obtain the corresponding basic matrix of the system $A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geqq \boldsymbol{0}$.

/** dual simplex method **/

⟨0⟩ (initialization) We first solve

$$\left|\begin{array}{ll} \min\limits_{\boldsymbol{x}\in\mathbb{R}^n} & \boldsymbol{dx} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geqq \boldsymbol{0}. \end{array}\right.$$

Let I_{B^0} and I_N be the index set of the optimal basic variables and nonbasic variables, respectively. Introducing slack variables s_j for $j \in I^z(\boldsymbol{w})$, we extend the problem to

$$\left|\begin{array}{ll} \min\limits_{(\boldsymbol{x},\boldsymbol{s})} & \boldsymbol{dx} \\ \text{s.t.} & \bar{A}\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{s}\end{bmatrix} = \bar{\boldsymbol{b}} \\ & (\boldsymbol{x},\boldsymbol{s}) \geqq \boldsymbol{0}, \end{array}\right.$$

where

$$\bar{A} = \begin{bmatrix} A & O \\ A' & I \end{bmatrix}, \quad A' = \left[\boldsymbol{e}^j\right]_{j\in I^z(\boldsymbol{w})}, \quad \text{and} \quad \bar{\boldsymbol{b}} = \begin{bmatrix}\boldsymbol{b}\\ \boldsymbol{0}\end{bmatrix}.$$

The current tableau is

A	O	$\boldsymbol{b}$
A'	I	$\mathbf{0}$
$\boldsymbol{d}$	$\mathbf{0}$	min

(6.8)

Defining $I_B = I_{B^0} \cup \{n+1, \ldots, n+|I^z(\boldsymbol{w})|\}$, we obtain matrices $B = \left[\boldsymbol{a}^j\right]_{j \in I_B}$ and $N = \left[\boldsymbol{a}^j\right]_{j \in I_N}$, where $\boldsymbol{a}^j$ is the jth column of $\bar{A}$.

$\langle k1 \rangle$ (dual simplex phase) Let $(\boldsymbol{d}^B, \boldsymbol{d}^N)$ be the partition of $(\boldsymbol{d},\ \mathbf{0})$ corresponding to B and N. Set $\bar{\boldsymbol{b}} := B^{-1}\bar{\boldsymbol{b}}$. If $\bar{\boldsymbol{b}} \geqq \mathbf{0}$, meaning the current basic solution is feasible, then go to $\langle k2 \rangle$ (Since $\bar{\boldsymbol{d}} \geqq \mathbf{0}$ is always met in the dual simplex phase, if $\bar{\boldsymbol{b}} \geqq \mathbf{0}$ then we obtain an optimal solution). Otherwise, choose the rth row of $\bar{\boldsymbol{b}}$ such that $\bar{b}_r < 0$. Set $\bar{\boldsymbol{d}} := \boldsymbol{d}^N - \boldsymbol{d}^B B^{-1} N$ and $\boldsymbol{\beta} := \boldsymbol{\beta}^r N$, where $\boldsymbol{\beta}^r$ is the rth row of B^{-1}. If $\boldsymbol{\beta} \geqq \mathbf{0}$, stop (there is no feasible solution). Otherwise, solve

$$s \in \arg\min \left\{ \frac{\bar{d}_j}{\beta_j^r} \;\middle|\; \bar{d}_j < 0 \right\}. \tag{6.9}$$

Update B and N by pivotting at (s, r), and go to $\langle k1 \rangle$.

$\langle k2 \rangle$ (deleting phase) Let I_B be the index set of the current basic variables. If $\{n+1, \ldots, n+|I^z(\boldsymbol{w})|\} \subseteqq I_B$ then set $I_B := I_B \setminus \{n+1, \ldots, n+|I^z(\boldsymbol{w})|\}$ and stop (the desired index set I_B is obtained). Otherwise, choose $j \in \{n+1, \ldots, n+|I^z(\boldsymbol{w})|\} \setminus I_B$ and add it into I_B by pivotting.

C An idea to reduce the number of variables

After explaining a possible way to reduce the number of variables of (mmF) we demonstrate that it does not work well due to the high degeneracy of the problem.

An idea to reduce the variables is a combination of primal and dual representations of the feasible region $X = \{\, \boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \mathbf{0},\ \mathbf{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c} \,\}$. First choose a vertex $\boldsymbol{v}^0 \in X_V \cap \operatorname{int} \bar{G}$ and enumerate all, say q adjacent vertices $\boldsymbol{v}^1, \ldots, \boldsymbol{v}^q \in X_V$

linked to $\boldsymbol{v}^0$. Then X can be rewritten as

$$X = \left\{ \boldsymbol{x} \in \mathbb{R}^n \,\middle|\, \boldsymbol{x} = \boldsymbol{v}^0 + \sum_{i=1}^{q} \mu_i(\boldsymbol{v}^i - \boldsymbol{v}^0),\ \boldsymbol{0} \leqq \boldsymbol{x} \leqq \boldsymbol{c},\ \boldsymbol{\mu} \geqq \boldsymbol{0} \right\}.$$

Letting $V = \begin{bmatrix} \boldsymbol{v}^1 - \boldsymbol{v}^0 & \cdots & \boldsymbol{v}^q - \boldsymbol{v}^0 \end{bmatrix} \in \mathbb{R}^{n \times q}$, we define

$$\begin{aligned} X^R &= \{\, \boldsymbol{\mu} \in \mathbb{R}^q \mid \boldsymbol{0} \leqq \boldsymbol{v}^0 + V\boldsymbol{\mu} \leqq \boldsymbol{c},\ \boldsymbol{\mu} \geqq \boldsymbol{0} \,\}, \text{ and} \\ X_M^R &= \{\, \boldsymbol{\mu} \in X^R \mid \bar{g}^R(\boldsymbol{\mu}) \leqq 0 \,\}, \end{aligned}$$

where

$$\bar{g}^R(\boldsymbol{\mu}) = \max\{\, \boldsymbol{e}V\boldsymbol{\nu} - n\boldsymbol{e}\boldsymbol{t} \mid \boldsymbol{\nu} \in X^R,\ V\boldsymbol{\nu} + \boldsymbol{t} \geqq V\boldsymbol{\mu},\ \boldsymbol{t} \geqq \boldsymbol{0} \,\} - \boldsymbol{e}V\boldsymbol{\mu}.$$

Then the problem (mmF) is equivalent to

$$(mmF_R) \qquad \left| \begin{array}{ll} \min\limits_{\boldsymbol{\mu}} & \boldsymbol{d}(\boldsymbol{v}^0 + V\boldsymbol{\mu}) \\ \text{s.t.} & \boldsymbol{\mu} \in X_M^R. \end{array} \right.$$

If $q < n$, (mmF_R) is worth considering, however, q can be much larger than n in spite of the low dimensionality of X. Take the network in Figure 6.2 with unit capacity for all arcs, i.e., $\boldsymbol{c} = \boldsymbol{e}$, and take the origin $\boldsymbol{0}$ as $\boldsymbol{v}^0 \in X_V \cap \operatorname{int} \bar{G}$. Then we see that the unit flow conveyed along a simple path from source to sink is a vertex of X linked to the origin. This means that q is as many as the simple paths from source to sink, which amounts to $3^3 = 27$ in this example while $n = 15$ and $\dim X = 13$.

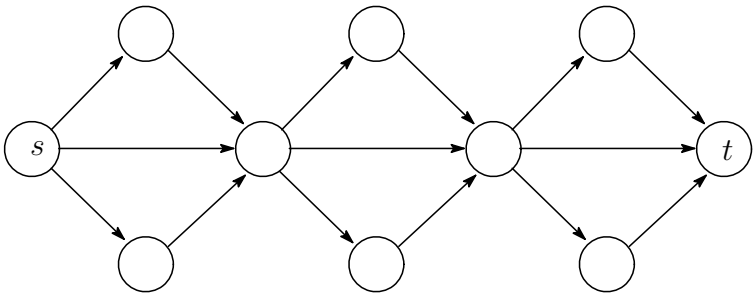


Figure 6.2: The case where q is larger than n

Acknowledgements

With great appreciation, I would like to express my gratitude to my supervisor, Professor Yoshitsugu Yamamoto for his valuable advice and encouragement. Completion of this thesis would be impossible without his help.

I am also grateful to Professors Ushio Sumita, Seizo Ikuta, Ryo Sato, Akiko Yoshise, Takahito Kuno, Maiko Shigeno, Kazutoshi Ando, Junya Gotoh and Masahiro Hachimori, for their valuable suggestions and criticisms that have improved this thesis.

My thanks also go to the faculty of the National Defense Academy: Professors Takeo Yamada, Seiji Kataoka and other members for their help and encouragement.

Also I benefited greatly from the comments, suggestions and criticisms provided by the anonymous referees.

I wish to thank my pleasant friends Hidetomo Hanatsuka, Ayami Suzuka, Hidetoshi Nagai, Hiroaki Sorimachi and other members of the E410 laboratory for their kindness and friendly cooperation.

I would like to thank all members of Japan Self-Defence Force for their aid. Last but not least, a special word of appreciation goes to my parents.

Bibliography

[1] F. A. Al-Khayyal and J. E. Falk. Jointly constrained biconvex programming. *Math. Oper. Res.*, 8(2):273–286, 1983.

[2] L. T. H. An. Analyse numérique des algorithmes de l'Optimisation d.c. Approaches locales et globales. Code et simulations numéiques en grande dimension. Applications. Thèse de Doctrat de l'Université de Rouen, France, 1994.

[3] L. T. H. An, P. D. Tao, and L. D. Muu. Numerical solution for optimization over the efficient set by d.c. optimization algorithms. *Oper. Res. Lett.*, 19:117–128, 1996.

[4] L. T. H. An, P. D. Tao, and L. D. Muu. Simplicially-constrained dc optimization over the efficient and weakly sets. *J. Optim. Theory Appl.*, 117:503–531, 2003.

[5] L. T. H. An, P. D. Tao, and N. V. Thoai. Combination between global and local methods for solving an optimization problem over the efficient set. *European J. Oper. Res.*, 142:258–270, 2002.

[6] M. Atteia and A. E. Qortobi. Quasi-convex duality. Lecture Notes, *Optimization and Optimizal Control* 30, Control and Information Science, 1981.

[7] H. P. Benson. Optimization over the efficient set. *J. Math. Anal. Appl.*, 98:562–580, 1984.

[8] H. P. Benson. An all-linear programming relaxation algorithm for optimizing over the efficient set. *J. Global Optim.*, 1:83–104, 1991.

[9] H. P. Benson. A finite, nonadjacent extreme-point search algorithm for optimization over the efficient set. *J. Optim. Theory Appl.*, 73(1):47–64, 1992.

[10] H. P. Benson. A geometrical analysis of the efficient outcome set in multiple objective convex programs with linear criterion functions. *J. Global Optim.*, 6:231–251, 1995.

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

[12] H. P. Benson and S. Sayin. Optimization over the efficient set: Four special cases. *J. Optim. Theory Appl.*, 80(1):3–18, 1994.

[13] S. Bolintineanu. Minimization of a quasi-concave function over an efficient set. *Math. Program.*, 61:89–110, 1993.

[14] P. C. Chen, P. Hansen, and B. Jaumard. On-line and off-line vertex enumeration by adjacency lists. *Oper. Res. Lett.*, 10:403–409, 1991.

[15] V. Chvátal. *Linear programming.* W.H. Freeman and Campany, New York, 1983.

[16] J. P. Dauer and T. A. Fosnaugh. Optimization over the efficient set. *J. Global Optim.*, 7:261–277, 1995.

[17] J. G. Ecker and I. A. Kouada. Finding efficient points for linear multiple objective programs. *Math. Program.*, 8:375–377, 1975.

[18] J. G. Ecker and I. A. Kouada. Finding all efficient extreme points for multiple objective linear programs. *Math. Program.*, 14:249–261, 1978.

[19] J. G. Ecker and J. H. Song. Optimizing a linear function over an efficient set. *J. Optim. Theory Appl.*, 83(3):541–563, 1994.

[20] R. M. Freund and J. B. Orlin. On the complexity of four polyhedral set containment problems. *Math. Program.*, 33:139–145, 1985.

[21] K. Fukuda and A. Prodon. Double description method revisited. In *Combinatorics and Computer Science*, pages 91–111, 1995.

[22] J. Fülöp. *A Cutting Plane Algorithm for Linear Optimization over the Efficient Set*. Lecture Notes 405, Economics and Mathematical Systems. Springer-Verlag, Heidelberg, 1994.

[23] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

[24] J. Gotoh, N. V. Thoai, and Y. Yamamoto. Global optimization method for solving the minimum maximal flow problem. *Optim. Methods Softw.*, 18:395–415, 2003.

[25] R. Horst, J. de Vries, and N. V. Thoai. On finding new vertices and redundant constraints in cutting plane algorithms for global optimization. *Oper. Res. Lett.*, 7(2):85–90, 1988.

[26] R. Horst and N. V. Thoai. Maximizing a concave function over the efficient or weakly-efficient set. *European J. Oper. Res.*, 117:239–252, 1999.

[27] R. Horst and H. Tuy. *Global Optimization*. Springer-Verlag, Berlin, third and revised and enlarged edition, 1995.

[28] Y. D. Hu and E. J. Sun. Connectedness of the efficient set in strictly quasiconcave vector maximization. *J. Optim. Theory Appl.*, 78(3):613–622, 1993.

[29] M. Iri. An essay in the theory of uncontrollable flows and congestion. Technical Report 94-03, Department of Information and System Engineering, Chuo University, TRISE, 1994.

[30] M. Iri. Network flow - theory and applications with practical impact. *System Modelling and Optimization*, pages 24–36, 1996.

[31] M. Iri. Theory of uncontrollable flows - a new type of network-flow theory as a model for the 21st century of multiple values. *Comput. Math. Appl.*, 35(10):107–123, 1998.

[32] H. Isermann and R. E. Steuer. Computational experience concerning payoff tables and minimum criterion values over the efficient set. *European J. Oper. Res.*, 33:91–97, 1987.

[33] H. Konno, P. T. Thach, and H. Tuy. *Optimization on Low Rank Nonconvex Structures*. Kluwer, Dordrecht, 1997.

[34] O. L. Mangasarian. *Nonlinear Programming*. MacGraw-Hill, New York, 1969.

[35] L. D. Muu. A convex-concave programming method for optimizing over the efficient set. *Acta Math. Vietnam.*, 25(1):67–85, 2000.

[36] L. D. Muu and L. T. Luc. *Global optimization approach to optimization over the efficient set*. Lecture Notes 452, Economics and Mathematical System. Springer-Verlag, Heidelberg, 1997.

[37] L. D. Muu and J. Shi. D.c. optimization methods for solving minimum maximal network flow problem. Lecture notes 1349, *captivation of convexity ; fascination of nonconvexity*, Research Institute for Mathematical Sciences, Kyoto University, 2004.

[38] P. H. Naccache. Connectedness of the set of nondominated outcomes in multicriteria optimization. *J. Optim. Theory Appl.*, 25(3):459–467, 1978.

[39] M. Padberg. *Linear programming*. Algorithms and Combinatorics 12. Springer-Verlag, Heidelberg, 1995.

[40] J. Philip. Algorithms for the vector maximization problem. *Math. Program.*, 2:207–229, 1972.

[41] T. Q. Phong and H. Q. Tuyen. Bisection search algorithm for optimizing over the efficient set. *Vietnam J. Math.*, 28:217–226, 2000.

[42] G. R. Reeves and R. C. Reid. Minimum values over the efficient set in multiple objective decision making. *European J. Oper. Res.*, 36:334–338, 1988.

[43] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer-Varlag, Berlin, 1998.

[44] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Monographs and Textbooks 176. Academic Press, Orlando, 1985.

[45] S. Sayin. An algorithm based on facial decomposition for finding the efficient set in multiple objective linear programming. *Oper. Res. Lett.*, 19:87–94, 1996.

[46] S. Sayin. Optimizing over the efficient set using a top-down search of faces. *Oper. Res.*, 48(1):65–72, 2000.

[47] J. Shi and Y. Yamamoto. A global optimization method for minimum maximal flow problem. *Acta Math. Vietnam.*, 22:271–287, 1997.

[48] M. Shigeno, I. Takahashi, and Y. Yamamoto. Minimum maximal flow problem - an optimization over the efficient set -. *J. Global Optim.*, 25:425–443, 2003.

[49] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application.* John Wiley & Sons, New York, 1986.

[50] R. E. Steuer. Manual for the adbase multiple objective linear programming package. *Department of Management Science and Information Technology, University of Georgia*, 1995.

[51] J. Stoer and C. Witzgall. *Convexity and Optimization in Finite Dimensions I.* Springer-Verlag, Berlin, 1970.

[52] P. D. Tao and E. B. Souad. Duality in d.c. (difference of convex functions) optimization. subgradient methods. In *International Series of Numerical Mathematics, Trends in Mathematical Optimization*, volume 84. Birkhäuser Verlag, Basel, 1988.

[53] P. T. Thach. Quasiconjugates of functions, duality relationship between quasiconvex minimization under a reverse convex constraints and quasiconvex maximization under a convex constraint, and applications. *J. Math. Anal. Appl.*, 159(2):299–322, 1991.

[54] P. T. Thach. Global optimality criterion and a duality with a zero gap in nonconvex optimization. *SIAM J. Math. Anal.*, 24(6):1537–1556, 1993.

[55] P. T. Thach. A nonconvex duality with zero gap and applications. *SIAM J. Optim.*, 4(1):44–64, 1994.

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

[57] T. V. Thieu, B. T. Tam, and V. T. Ban. An outer-approximation method for globally minimizing a concave function over a compact convex set. *Acta Math. Vietnam.*, 8:21–40, 1983.

[58] N. V. Thoai. A class of optimization problems over the efficient set of a multiple criteria nonlinear programming problem. *European J. Oper. Res.*, 122:58–68, 2000.

[59] N. V. Thoai. Conical algorithm in global optimization for optimizing over efficient sets. *J. Global Optim.*, 18:321–336, 2000.

[60] J. F. Toland. On subdifferential calculus and duality in nonconvex optimization. *Bull. Soc. Math. France, Mémoire*, 60:177–183, 1979.

[61] H. Tuy. D.c. optimization: Theory, methods and algorithms. In R. Horst and P. M. Pardalos, editors, *Handbook of Global Optimization*. Kluwer Academic Publishers, Netherlands, 1995.

[62] H. Tuy. *Convex Analysis and Global Optimization*. Kluwer, Dordrecht, 1998.

[63] D. J. White. *Optimality and Efficiency*. John Wiley & Sons, Chichester, 1982.

[64] D. J. White. The maximization of a function over the efficient set via a penalty function approach. *European J. Oper. Res.*, 94:143–153, 1996.

[65] S. Yamada, T. Tanino, and M. Inuiguch. An inner approximation method for optimization over the weakly-efficient set. *J. Global Optim.*, 16:197–217, 2000.

[66] Y. Yamamoto. Optimization over the efficient set: Overview. *J. Global Optim.*, 22(1-4):285–317, 2002.

[67] Y. Yamamoto and D. Zenke. D.c. optimization methods for the minimum maximal flow problem. Discussion Paper 1094, Institute of Policy and Planning Sciences, Tsukuba University, 2004.

[68] Y. Yamamoto and D. Zenke. Cut and split method for the minimum maximal flow problem. *Pacific J. Optim.*, 1(2):387–404, 2005.

[69] Y. Yamamoto and D. Zenke. Outer approximation method for the minimum maximal flow problem. Discussion Paper 1115, Institute of Policy and Planning Sciences, Tsukuba University, 2005.