数理解析研究所講究録 1349 巻 2004 年 83-93

D.C. Optimization Methods for Solving Minimum Maximal Network Flow Problem

Le Dung Muu*
(Hanoi Institute of Mathematics, Vietnam)

室蘭工業大学・情報工学科　施　建明† (Jianming SHI)
(Muroran Institute of Technology, Japan)

Abstract

We consider the minimum maximal flow problem, i.e., minimizing the flow value among maximal flow, which is an $\mathcal{NP}$-hard problem. After formulating the problem, we introduce some formulations of the problem and one of them is a minimization of a concave function over a convex set. The problem can also be cast into a d.c. programming. We propose a DCA algorithm for solving the problem.

Keywords: minimum maximal flow, d.c. optimization, smooth variational inequality, branch-and-bound, DCA algorithm

AMS Classification 2000: C90 C29

1 Introduction

The field of network flows has a rich and long history, tracing its roots back to the work of Gustav Kirchof who first systematically analyzed electrical circuits and other early pioneers of electrical engineering and mechanics. Such early work established the foundation of the key ideas of network flow theory. The key task of this field is to answer such questions as: which way to use a network is most cost-effective? Maximum flow problem and minimum cost flow problem are two typical problems of them. However, from the point of view of practical cases, we have another kind of problems which are inherently different from the typical ones. For instance, Figure 1 portrays a network with edge-flow-capacity 1 (unit) on all edges, where node s is the source and node t is the sink. The case (a) of the figure illustrates the maximum flow of the network, that is, the flow on all edges is 1 except the edge x_3, whose flow is 0. On the other hand, if the flow on x_3 is fixed at 1 and we cannot reduce it by some reasons such as emergency, then the network cannot be exploited at the most economical situation. In the case (a), we can send 2 (unit) of flow between nodes s and t. But in the case (b) where the flow on x_3 is fixed at 1, the possible flow value we can send between s and t is 1 (unit). The flow value we can send between s and t reduces from 2 (case (a)) to 1 (case (b)) due to the

*This work was completed during a visit of this author at the Department of Mathematics, University of Namur, Belgium. email: muu@math.ac.vn

†This author was partially supported by a Grant-in-Aid (C-14550405) from the Ministry of Education, Science, Sports and Culture of Japan. email: shi@csse.muroran-it.ac.jp

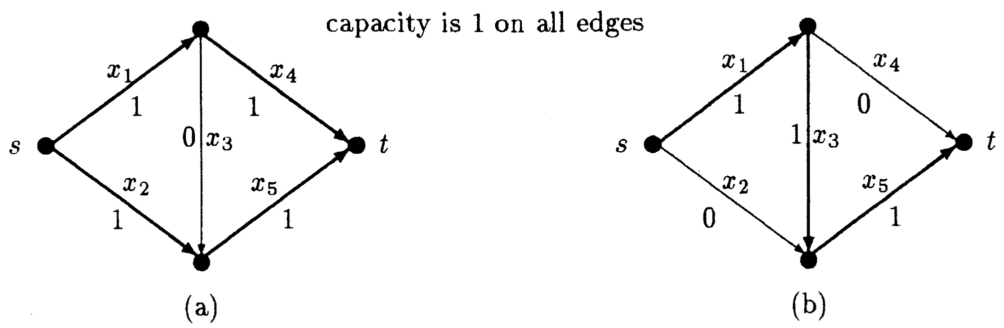

Figure 1: *Minimum maximal flow problem.*

fact that the flow value of x_3 is not controllable. It means that the maximum flow value is not attainable if the users on the network are disobedient.

From the point of view of modeling, the two cases are essentially different though they bear some resemblance. Assuming that the flow is controllable, the case (a) aims at an optimal value of flow. The case (b) also searches for an optimal value of flow, without the controllability of a network flow. The standard network-flow with the controllability has been well studied for several decades. Without the controllablitily, many problems in network-flow, e.g., the maximum flow problem, become more difficulty. Compared to the standard network-flow theory, uncontrollable network-flow theory is a new field, hence is still in its infancy.

The natural question in this new field is: given a network $\mathcal{N}$, how to calculate the attainable maximum-value of flow of $\mathcal{N}$ when the flow is not controllable. To answer the question, in this paper we consider minimum maximal flow (MMF) problem which finds out the minimum-value among the *maximal flows* of the network $\mathcal{N}$. Iri [2] gave the definition of *uncontrollable flow* (*u-flow*) and presented fundamental problems related to u-flow. Although the concept of u-flow is quite different from maximal flow and their relationship is not known yet so much, the optimal value of minimum maximal u-flow of $\mathcal{N}$ is equal to the optimal value of MMF under some assumption. In Iri's profound essay, several fundamental theorems and new research topics are described, but no algorithms for the corresponding problems are proposed. To the authors' knowledge, no algorithms for MMF were known until Shi-Yamamoto [6]. As pointed out in [8], Shi-Yamamoto's algorithm is not efficient enough. After that, some algorithms for solving the problem were proposed in such as Shigeno-Takahashi-Yamamoto [7] and others. Since the theory dealing with the network-flow problems without assuming the amenability of flows is still in its infancy, in this paper we focus on the development of algorithm for MMF in virtue of d.c. optimization.

In next section 2, we give the problem mathematically and its equivalent formulations. In section 3, we discuss some related properties of d.c. programming and d.c. algorithm. Section 4 gives solution methods and a branch-and-bound algorithm that is based on local search technique for d.c. algorithm. Finally, a brief concluding remark is give in section 5. Unfortunately, at this stage no any numerical experiment can be reported in this paper.

2 Equivalent Formulations

Consider a directed network $\mathcal{N}(V, E, s, t, c)$, where V is the set of $m+2$ nodes, E is the set of n arcs, s is the single source node, t is the single sink node, and c is the vector of arc capacities. Let X denote the set of feasible flows, i.e.,

$$X := \{x \mid x \in R^n, Ax = 0, 0 \le x \le c\},$$

where the matrix A stands for a node-arc incident relationship in the network. Obviously, X is a compact convex set. A vector $z \in X$ is said to be *maximal flow* if there does not exists $x \in X$ such that $x \ge z$ and $x \ne z$. Let denote by f the flow value function. Usually f is assumed to be linear on X. For instance, it usually defined by

$$f(x) = \sum_{i \in \Delta^+(s)} x_i - \sum_{i \in \Delta^-(s)} x_i,$$

where $\Delta^+(s)$ and $\Delta^-(s)$ are the sets of arcs which leaves and enters the source node s, respectively. Then, f is a linear function. Let $d \in R^n$ with

$$d_i = \begin{cases} 1 & \text{if} \quad i \in \Delta^+(s), \\ -1 & \text{if} \quad i \in \Delta^-(s), \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

We see that $f(x) = dx$.

Throughout this paper R^k denotes the set of k-dimensional real column vectors, $R^k_+ = \{x \mid x \in R^k; x \ge 0\}$ and $R^k_{++} = \{x \mid x \in R^k, x > 0\}$. R_k denotes the set of k-dimensional real row vectors, and R_k^+ and R_k^{++} are defined in the similar way. We use e to denote both a row vector and a column vector of ones, and e_i to denote the ith unit row vector of an appropriate dimension. For a set S, $V(S)$ is the set of extreme points of S.

Let denote by X_M be the set of all maximal flows, i.e.,

$$X_M := \{z \in X \mid \text{ there does not exist } x \in X \text{ such that } x \ge z \text{ and } x \ne z\}.$$

The problem to be considered is given as

$$(P) \quad \min\{dx \mid x \in X_M\}. \tag{2.2}$$

Denote X_E the efficient set of the vector optimization problem

vmax x subject to $x \in X$.

Then (P) is equivalent to the problem

$$\min\{dx \mid x \in X_E\}. \tag{2.3}$$

2.1 Primal Formulation

We define a function r as

$$r(x) := \max\{e(y-x) \mid y \ge x, y \in X\}. \tag{2.4}$$

Clearly, $r(x) \ge 0$ for all $x \in X$. It is easy to see that r is a concave function on X. In fact, $\forall \beta \in (0,1)$ and $x', x'' \in X$ we have

$$\begin{aligned} & \beta r(x') + (1-\beta) r(x'') \\ = \; & \beta \max\{e(y-x') \mid y \ge x', y \in X\} + (1-\beta)\max\{e(y-x'') \mid y \ge x'', y \in X\} \\ = \; & \beta e(y_{r(x')} - x') + (1-\beta) e(y_{r(x'')} - x'') \\ = \; & e(\beta y' + (1-\beta) y'' - (\beta x' + (1-\beta) x'')) \\ \le \; & \max\{e(y - (\beta x' + (1-\beta) x'')) \mid y \ge \beta x' + (1-\beta) x'', y \in X\} \\ = \; & r(\beta x' + (1-\beta) x''), \end{aligned}$$

where $y_{r(x)} \in \arg\max\{e(y-x) \mid y \geq x, y \in X\}$. Moreover, $r(x)$ is piecewise-linear on X. In fact, adding a slack z such that

$$\begin{pmatrix} A & 0 & 0 \\ I & I & 0 \\ I & 0 & -I \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ c \\ x \end{pmatrix}, z \in R_+^{2n}, \iff Ay = 0, x \leq y \leq c.$$

Then for a given $x \in R_+^n$, $r(x)$ is a solution of the following linear programming:

$$\left|\begin{array}{ll} \max & ey - ex \\ s.t. & \begin{pmatrix} A & 0 & 0 \\ E & E & 0 \\ E & 0 & -E \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ c \\ x \end{pmatrix}, \\ & y \geq 0, z \geq 0, \end{array}\right. \tag{2.5}$$

where E is an $(n \times n)$ unit matrix. As $r(x)$ is a solution of a linear maximization, we assume that

$$r(x) = c_B B^{-1} \begin{pmatrix} 0 \\ c \\ x \end{pmatrix} - ex, \tag{2.6}$$

where c_B is a corresponding coefficient vector of objective, and B is a basic matrix of problem (2.5).

Lemma 1 *If the capacity c is integral, then so is $r(x)$ for any integer x.*

Proof: This is trivial from (2.6). □

Because of concavecity, we assume that

$$r(x) = \min\{\ell_i(x) \mid i \in I\}, \tag{2.7}$$

where $\ell_i(x) := \langle l_i, x\rangle + \alpha_i$ are linear functions on R^n. It is easy to see that $x \in X, r(x) = 0$ if and only if $x \in X_E$. Hence Problem (2.2) can be rewritten equivalently as

$$\min\{dx \mid x \in X, r(x) \leq 0\}. \tag{2.8}$$

Consider the following penalized problem for a fixed number t.

$$P(t) \quad \min\{dx + t \cdot r(x) \mid x \in X\}. \tag{2.9}$$

Let $t_* := \max\{dx \mid x \in X\} - \min\{dx \mid x \in X\}$. Then

Lemma 2 *There exists a finite number $t_* \geq 0$, such that for every $t > t_*$, problem (2.8) is equivalent to (2.9).*

Proof: Therefore for any $t > t_*$ and $r(x) \neq 0$ we must have $r(x) \geq 1$ and

$$\begin{aligned} & \min\{dx + t \cdot r(x) \mid x \in X\} \\ \geq\ & \min\{dx + t_* \cdot r(x) \mid x \in X\} \\ \geq\ & \min\{dx \mid x \in X\} + \max\{dx \mid x \in X\} - \min\{dx \mid x \in X\} \\ \geq\ & \min\{dx \mid x \in X, r(x) = 0\}. \end{aligned}$$

And when $r(x_0) = 0$ then we see $dx_0 \geq \min\{dx \mid x \in X, r(x) = 0\}$. Hence,

$$\min\{dx + t \cdot r(x) \mid x \in X\} \geq \min\{dx \mid x \in X, r(x) = 0\}.$$

On the other hand, a feasible solution of $\min\{dx, \mid x \in X, r(x) = 0\}$ is also a feasible solution of (2.9). We have

$$\min\{dx + t \cdot r(x) \mid x \in X\} \leq \min\{dx \mid x \in X, r(x) = 0\}.$$

It implies that

$$\min\{dx + t \cdot r(x) \mid x \in X\} = \min\{dx \mid x \in X, r(x) = 0\}.$$

Notice that $r(\cdot) \geq 0$. Then, any $t > t_* \geq 0$ we have

$$\min\{dx + t \cdot r(x) \mid x \in X\} = \min\{dx \mid x \in X, r(x) = 0\} = \min\{dx \mid x \in X, r(x) \leq 0\}.$$

It is the assertion. □

Let denote by δ_X the indicator of X and $g(x) := dx + \delta_X(x)$

$$h(x) := \begin{cases} -t_* \cdot r(x) & \text{if } x \in X \\ +\infty & \text{if } x \notin X. \end{cases}$$

Then $g(x)$ and $h(x)$ are s convex and problem (P) is rewritten as

$$(P) \quad \min\{dx + \delta_X(x) - h(x)\} = \min\{g(x) - h(x)\}. \tag{2.10}$$

This is a d.c. programming. Hereafter, we use the formation for local search in DCA.

2.2 Dual formulation

From a result of Philip [5] it follows that there exists a simplex $\Lambda \subseteq R_n$ such that a vector x is maximal flow if and only if there exists $\lambda \in \Lambda$ such that

$$\lambda x \geq \lambda y, \ \forall\, y \in X. \tag{2.11}$$

Thus the minimum-maximal flow problem to be considered can also be formulated as

$$\left| \begin{array}{ll} \min & dx \\ s.t. & \lambda \in \Lambda, \\ & x \in X, \\ & -\lambda(y - x) \geq 0, \ \forall\, y \in X. \end{array} \right. \tag{2.12}$$

This is a special case of mathematical programming with variational inequality constraints.

Denote

$$g(x, \lambda) = \frac{1}{2}||x||^2 + \frac{1}{2}||\lambda||^2 + \max_{v \in X}\left\{vx + v\lambda - \frac{1}{2}||v||^2\right\} \tag{2.13}$$

and

$$h(x, \lambda) = \frac{1}{2}||x + \lambda||^2 + \frac{1}{2}||x||^2. \tag{2.14}$$

Then

Lemma 3 *The constraints in (2.12) can be cast into the form*

$$\lambda \in \Lambda, x \in X, \; g(x,\lambda) - h(x,\lambda) = 0. \tag{2.15}$$

Proof: It is easy to see that

$$g(x,\lambda) - h(x,\lambda) = \max_{v \in X} \left\{ \lambda(v - x) - \frac{1}{2}\|v - x\|^2 \right\}.$$

Note that X is a convex set. Suppose that (2.11) holds for some $x \in X$ and some $\lambda \in \Lambda$. We have that

$$0 \leq \max_{v \in X} \left\{ \lambda v - \lambda x - \frac{1}{2}\|v - x\|^2 \right\} \leq \max\{\lambda v - \lambda x \mid v \in X\} = 0$$

which yields $g(x,\lambda) - h(x,\lambda) = 0$.

Suppose that $g(x,\lambda) - h(x,\lambda) = 0$ for some $x \in X$ and $\lambda \in \Lambda$. Then we have that

$$\max_{v \in X} \left\{ \lambda(v - x) - \frac{1}{2}\|v - x\|^2 \right\} = 0 \tag{2.16}$$

which implies that $\lambda(v - x) \leq 0$ for all $v \in X$. In fact, if we have some $v_0 \in X$ such that $\lambda(v_0 - x) > 0$ then we can take a point $\bar{v}$ on line segment $[v_0, x]$ satisfying $\|\bar{v} - x\| < \|\lambda\| \cos\theta$, where θ is the acute angel between λ and $v_0 - x$. Note that X is convex then $\bar{v} \in X$ but $\lambda(\bar{v} - x) - \frac{1}{2}\|\bar{v} - x\|^2 > 0$. It contradicts (2.16). □

Note that the both functions g and h are convex and differentiable.

From this lemma, it follows that the problem can be formulated by the following d.c. differentiable programming:

$$(DP) \quad \left| \begin{array}{ll} \min & f(x) \\ s.t. & \lambda \in \Lambda, \\ & x \in X, \\ & g(x,\lambda) - h(x,\lambda) = 0. \end{array} \right. \tag{2.17}$$

From Shigeno-Takahashi-Yamamoto [7], we see that the Λ in (2.17) could be replaced by

$$\{\lambda \mid \lambda \in R_n^{++}, \lambda \geq e, \lambda e = n^2\}.$$

Then we take the above set as Λ to design algorithms.

3 Properties of d.c programming and DCA

DCA (D.C algorithm) [4] is a prime-dual approach for finding local optimum in d.c. programming. More detailed results on DCA can be found in scuh as [3]. Some numerical experiments are reported that it finds a global minimizer often if one chose a 'good' start point.

Consider the following general problem:

$$(DC_p) \quad v_p := \inf\{g(x) - h(x) \mid x \in R^n\} \tag{3.18}$$

where $g(\cdot), h(\cdot) : R^n \to R \cup \{-\infty, +\infty\}$ are low semicontinuous (lsc) convex functions on R^n. It is easy to see that problem (P) is a special case of (DC_p) as shown in (2.10) under the convention $+\infty = +\infty - (+\infty)$. We also suppose that $g(x) - h(x)$ is bounded below on R^n. The ε-subgradient of g at point x_0 are defined by

$$\partial_\varepsilon g(x_0) := \{y \in R^n \mid g(x) \geq g(x_0) + \langle x - x_0, y\rangle - \varepsilon, \forall x \in X\}$$

and $\partial g(x_0) := \partial_0 g(x_0)$. The conjugate function of g is given by

$$g^*(y) := \sup\{\langle x, y\rangle - g(x) \mid x \in R^n\}.$$

From lsc of and g and h, we see that $g = g^{**}$ and $h = h^{**}$ hold. Consider a dual problem of (DC_p):

$$(DC_d) \quad v_d := \inf\{h^*(y) - g^*(y) \mid y \in R^n\}. \tag{3.19}$$

We have that

$$\begin{aligned} v_p &= \inf\{g(x) - h(x) \mid x \in R^n\} \\ &= \inf\{g(x) - \sup\{\langle x, y\rangle - h^*(y) \mid y \in R^n\} \mid x \in R^n\} \\ &= \inf\{g(x) + \inf\{h^*(y) - \langle x, y\rangle \mid y \in R^n\} \mid x \in R^n\} \\ &= \inf\{h^*(y) + \inf\{g(x) - \langle x, y\rangle \mid x \in X\} \mid y \in R^n\} \\ &= \inf\{h^*(y) - \sup\{\langle x, y\rangle - g(x) \mid x \in X\} \mid y \in R^n\} \\ &= \inf\{h^*(y) - g^*(y) \mid y \in R^n\} \\ &= v_d. \end{aligned} \tag{3.20}$$

For a pair (x, y), Fenchel's inequality $g(x) + g^*(y) \geq \langle x, y\rangle$ holds for any proper convex function g and its conjugate g^*. If $y \in \partial g(x)$ then $g(x) + g^*(y) = \langle x, y\rangle$.

Theorem 4 *(Hiriart-Urruty '88) A point x^* is a globally optimal solution of DC_p if and only if $\partial_\varepsilon h(x^*) \subseteq \partial_\varepsilon g(x^*)$ holds for every $\varepsilon > 0$.*

Proof: Suppose x^* is a globally optimal solution of (DC_p) and $y_0 \in \partial_\varepsilon h(x^*)$. Then $x^* \in \mathrm{dom}(h)$ and by (3.20) $h^*(y) - g^*(y) \geq v_d = v_p = g(x^*) - h(x^*)$ that implies

$$h(x^*) + h^*(y) \geq g(x^*) + g^*(y) \tag{3.21}$$

holds for every $y \in \mathrm{dom}(h^*)$, and $\langle x^*, y_0\rangle + \varepsilon \geq h(x^*) + \langle y, y_0\rangle - h(y)$ which yields

$$\langle x^*, y_0\rangle + \varepsilon \geq h(x^*) + h^*(y_0). \tag{3.22}$$

Ultimately, it follows from (3.21) and (3.22) that for every $\varepsilon > 0$,

$$\varepsilon + \langle x^*, y\rangle \geq h(x^*) + h^*(y_0) \geq g(x^*) + g^*(y_0). \tag{3.23}$$

It is nothing but $y_0 \in \partial_\varepsilon g(x^*)$.

Suppose $\partial_\varepsilon h(x^*) \subseteq \partial_\varepsilon g(x^*)$ holds for every $\varepsilon > 0$. It yields

$$h(x^*) - h(y) \geq g(x^*) - g(y)$$

for all $y \in R^n$. In fact, if $h(x^*) - h(y_0) < g(x^*) - g(y_0)$ for some y_0 we take $z \in \partial h(y_0)$ and

$$\varepsilon_0 := \frac{1}{2}\{g(x^*) - g(y_0) - h(x^*) + h(y_0)\} + h(x^*) - h(y_0) + \langle y_0 - x^*, z\rangle.$$

Then $\varepsilon_0 > 0$. From $z \in \partial h(y_0)$ we have $h(x) \geq h(y_0) + \langle x - y_0, z\rangle$. Hence

$$h(x^*) - h(y_0) + \langle y_0 - x^*, z\rangle + h(x) \geq h(x^*) + \langle x - x^*, z\rangle.$$

It implies $\varepsilon_0 + h(x) \geq h(x^*) + \langle x - x^*, z\rangle$ for all $x \in R^n$, i.e., $z \in \partial_{\varepsilon_0} h(x^*)$. On the other hand, by the definition of ε_0 we have

$$\varepsilon_0 < g(x^*) - g(y_0) + \langle y_0 - x^*, z\rangle.$$

It means $z \notin \partial_{\varepsilon_0} g(x^*)$. It contradicts $\partial_\varepsilon h(x^*) \subseteq \partial_\varepsilon g(x^*)$ for every $\varepsilon > 0$. □

For a given point, it is still very difficult to check its globally optimality by Theorem 4. Let us consider some local properties of $g - h$. A point $x^\star$ is said to be *local minimal* of $g - h$ if there exists a neighborhood N of $x^\star$ such that

$$(g - h)(x) \geq (g - h)(x^\star), \forall x \in N.$$

It is easy to see the following result.

Lemma 5 *A point x^* is local minimal for $g - h$, then $\partial h(x^\star) \subseteq \partial g(x^\star)$.*

Proof: Suppose that $(g-h)(x) \geq (g-h)(x^\star), \forall x \in N$. Then $g(x) - g(x^\star) \geq h(x) - h(x^\star)$ Taking $z \in \partial h(x^\star)$, we have $h(x) \geq h(x^\star) + \langle x - x^\star, z\rangle$ for all $x \in R^n$. Therefore we see that $g(x) \geq g(x^\star) + \langle x - x^\star, z\rangle$ for $x \in N$. Note that g is convex, then $g(x) \geq g(x^\star) + \langle x - x^\star, z\rangle$ holds for $x \in R^n$. □

Lemma 6 *If h is a piecewise-linear convex function on* dom(h) *and $\partial h(x^\star) \subseteq \partial g(x^\star)$, then $x^\star$ is local minimal for $g - h$.*

Proof: It is enough to consider $x \in \mathrm{dom}(g)$. Suppose that h is piecewise-linear convex. Then there exists a neighborhood $N(x^\star)$ such that for any $x \in N(x^\star)$ we can choose $z \in \partial h(x^\star)$ such that $h(x) - h(x^\star) = \langle x - x^\star, z\rangle$. From $\partial h(x^\star) \subseteq \partial g(x^\star)$ we have $g(x) \geq g(x^\star) + \langle x - x^\star, z\rangle$ holds for $x \in N(x^\star)$. It implies that $g(x) - h(x) \geq g(x^\star) - h(x^\star)$ for $x \in N(x^\star)$. Then $x^\star$ is local minimal for $g - h$. □

Next we describe the framework of DCA.

algorithm BASIDCA

step 0: pick up a point $x^0 \in \mathrm{dom}(h)$, calculate $y^0 \in \partial h(x^0)$; $k = 1$;
step 1: calculate $x^k \in \arg\min\{g(x) - (h(x^{k-1}) + \langle x - x^{k-1}, y^{k-1}\rangle) \mid x \in R^n\}$;
calculate $y^k \in \arg\min\{h^*(y) - (g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle) \mid y \in R^n\}$;
step 2: If $\partial h(x^k) \cap \partial g(x^k) \neq \emptyset$, stop; otherwise, $k := k + 1$ goto step 1.

Lemma 7 *Suppose that the points x^k and y^k are generated in* BASIDCA, *then $x^k \in \partial h^*(y^k)$ and $y^{k-1} \in \partial g(x^k)$.*

Proof: Assume that x^{k-1} and y^{k-1} are in hand. We have

$$\begin{aligned} & \min\{g(x) - (h(x^{k-1}) + \langle x - x^{k-1}, y^{k-1}\rangle) \mid x \in R^n\} \\ = \; & \min\{g(x) - \langle x, y^{k-1}\rangle \mid x \in R^n\} - h(x^{k-1}) + \langle x^{k-1}, y^{k-1}\rangle \end{aligned} \tag{3.24}$$

and

$$\begin{aligned} & \min\{h^*(y) - (g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle) \mid y \in R^n\} \\ = \; & \min\{h^*(y) - \langle x^k, y\rangle \mid y \in R^n\} - g^*(y^{k-1}) + \langle x^k, y^{k-1}\rangle. \end{aligned} \tag{3.25}$$

Thus, from Step 1 of BASIDCA $g(x) - \langle x, y^{k-1}\rangle \geq g(x^k) - \langle x^k, y^{k-1}\rangle$ for all x, and $h^*(y) - \langle x^k, y\rangle \geq h^*(y^k) - \langle x^k, y^k\rangle$ for all y. It yields $y^{k-1} \in \partial g(x^k)$ and $x^k \in \partial h^*(y^k)$. □

Lemma 8 *$g(x^k) - h(x^k)$ and $h^*(y^k) - g^*(y^k)$ in* BASIDCA *are decreasing as iteration k increasing. If $g - h$ is bounded below and* BASIDCA *does not terminate within finitely many iterations, then $g(x^k) - h(x^k)$ and $h^*(y^k) - g^*(y^k)$ in* BASIDCA *are decreasing as iteration k increasing, and the sequence $\{x^k\}_{k=1,2,\cdots}$ converges to an accumulate point $x^\star$ such that $\partial h(x^\star) \cap \partial g(x^\star) \neq \emptyset$.*

Proof: By $y^{k-1} \in \partial g(x^k)$ in Lemma 8, we have $g(x^k) + g^*(y^{k-1}) = \langle x^k, y^{k-1}\rangle$. Then

$$\begin{aligned} & g(x^k) - h(x^k) = \langle x^k, y^{k-1}\rangle - g^*(y^{k-1}) - h(x^k) \leq h^*(y^{k-1}) - g^*(y^{k-1}) \\ \leq \; & h^*(y^{k-1}) + g(x^{k-1}) - \langle x^{k-1}, y^{k-1}\rangle \leq g(x^{k-1}) - h^{**}(x^{k-1}) \leq g(x^{k-1}) - h(x^{k-1}). \end{aligned}$$

Notice that $g(x) - h(x)$ is bounded below, therefore there exists a number $\alpha^\star$ such that

$$\lim_{k\to\infty} g(x^k) - h(x^k) = \alpha^\star$$

Assume that $x^\star, y^\star$ are two accumulated points of x^k and y^k, respectively. Then $g(x^\star) + g^*(x^\star) = h(x^\star) + h^*(x^\star)$. On the other hand, from $y^{k-1} \in \partial g(x^k)$ we obtain $\langle x^\star, y^\star\rangle = g(x^\star) + g^*(x^\star)$. It implies that $x^\star \in \partial g(x^k) \cap \partial h(x^\star)$. The assertion is proved. □

Generally, when we have $y^0 \in \partial h(x^\star)$ and $y^0 \notin \partial g(x^\star)$, then

$$h(x) \geq h(x^\star) + \langle x - x^\star, y^0\rangle \tag{3.26}$$

and $\langle x^\star, y^0\rangle < g(x^\star) + g^*(y^0)$. Let $x^0 := x^\star$, by Step 1 of BASIDCA we obtain x^1 such that $y^0 \in \partial g(x^1)$. Then $\langle x^1, y^0\rangle = g(x^1) + g^*(y^0)$. From (3.26) we have $\langle x^0, y^0\rangle \geq h(x^0) - h(x^1) + \langle x^1, y^0\rangle$. From the above inequalities and expressions we see that

$$\begin{aligned} & h(x^0) - h(x^1) + g(x^1) + g^*(y^0) \\ = \; & h(x^0) - h(x^1) + \langle x^1, y^0\rangle \\ \leq \; & \langle x^0, y^0\rangle \\ < \; & g(x^0) + g^*(y^0). \end{aligned}$$

It yields, ultimately, $g(x^1) - h(x^1) < g(x^0) - g(x^0)$. It means that when $\partial h(x^0) \not\subseteq \partial g(x^0)$ then we can find a smaller value at point x^1.

4 Solution Methods and Algorithms

Now we go back to problem (P). In this section, we give an algorithms to solve the problem. A general framework of branch-and-bound algorithm can be stated follows.

algorithm GF

step 0: initial setting and calculating,
step 1: branching operation,
step 2: local search for a smaller upper bound,
step 3: find a larger lower bound,
step 4: remove some regions, go to Step 1.

We describe the Step 1-3 following in detail.

branching operation (Step 1) A simplex-based division is usually exploited in branch-and-bound method. At some step, a contemporary simplex S is divided into two smaller ones S_1 and S_2. Taking into account the convergence of the algorithms we need the division to be *exhaustive*, i.e., a nested sequence of simplices $\{S_k\}_{k=1,2,\cdots}$ has the following properties: $\mathrm{int}(S_i) \cap \mathrm{int}(S_j) = \emptyset$ if $i \neq j$, $S_{k+1} \subseteq S_k$ for all k, $\lim_{k\to\infty} \cap_{k=1}^{\infty} S_k = x^0$ for some x^0. At each step, we chose divide a simplex S_k into two smaller ones S_{2k} and S_{2k+1} by bisecting the longest edge of S_k. The sequence $\{S_k\}_{k=1,2,\cdots}$ in such process is exhaustive.

local search for a smaller upper bound (Step 2) There are many methods to do local search. Here we exploit BASIDCA in this step. Even BASIDCA is not going to find a global optimum theoretically,, but in many numerical experiments, it finds a global optimum practically.

As shown in (2.10) problem (P) can be rewritten as a d.c. programming $\min\{g - h\}$, then we can use BASIDCA to obtain a locally optimal solution. Then we assume that

$$u^i := \mathrm{basiDCA}(X \cap S_i)$$

is a local optimal solution on $X \cap S_i$ by BASIDCA.

find a larger lower bound (Step 3) Assume that $l_i(x)$ is an affine function such that $l_i(v_j) = h(v_j)$ for all vertices $v_j \in V(S_i)$. From the convexcity of $h(x)$, we have $l_i(x) \geq h(x)$ for all $x \in S_i$. Then

$$L(X \cap S_i) := \min\{dx + \delta_{X\cap S_i}(x) - l_i(x) \mid x \in R^n\} \leq \min\{dx + \delta_{X\cap S_i}(x) - h(x) \mid x \in R^n\}.$$

Moreover, if $V(S_i) := \{v_1, \cdots, v_{p_i}\}$ is in hand, then it is easy to calculate $L(X \cap S_i)$ because

$$\begin{aligned} & \min\{dx + \delta_{X\cap S_i}(x) - l_i(x) \mid x \in R^n\} \\ = \; & \min\left\{ d\left(\sum_{j=1}^{p_i} \lambda_j v_j\right) + t_*\left(\sum_{j=1}^{p_i} \lambda_j r(v_j)\right) \;\middle|\; \begin{array}{l} \sum_j \lambda_j = 1, \lambda_j \geq 0, \\ A(\sum_{j=1}^{p_i} \lambda_j v_j) = b, 0 \leq \sum_{j=1}^{p_i} \lambda_j v_j) \leq c \end{array} \right\}. \end{aligned}$$

Based on the above discussion, we give a whole algorithm as follows.

algorithm MMFDCA

step 0: let ε and S_0 such that $X \subseteq S_0$. let $x^0 := 0, y^0 := (-1, \cdots, -1)$,
$b^U := \mathrm{basiDCA}(X), b_L := \min\{dx \mid x \in Ax = b, 0 \leq x \leq b\}, M := S_0$
step 1: select $S_0 \in M$ such that $b_L = L(X \cap S_0)$ and divid S_0 into S_1 and S_2,
step 2: $u^i := \mathrm{basiDCA}(X \cap S_i)$ for all $i = 1, 2$. if $u^i < b^U$ then $b^U = u^i$,
step 3: if $L(X \cap S_i) > b_L$ then $b_L = L(X \cap S_i)$, if $b^U - b^L < \varepsilon$ then Stop,
step 4: $M := \{S \in M \mid L(X \cap S) < b^U\}$, if $M = \emptyset$ then Stop, otherwise go to Step 1.

The convergence of the the above algorithm is from the exhaustive partition. Here we omit the detailed proof. A general proof under such exhaustiveness can be found in many books, scuh as [1].

5 Conclusion

A branch-and-bound algorithm via DCA subroutine for solving problem (P) is proposed in this paper. Apart from the algorithm, we also discussed a dual formulation for (P) and investigated some properties of a general d.c. programming. Though we have not proposed an algorithm for problem (DC_p), it can be solved by differential programming. Due to that problem $(P(t_*))$ is a concave minimization over a convex set, we can solve it by many existing methods directly or indirectly. Among these methods, it might be interesting to compare the behaviour of the different algorithms. We consider it as further research topics.

References

[1] Horst, R. and H. Tuy, *Global Optimization, Deterministic Approaches*, second edition, Springer-Verlag, 1993.

[2] Iri, M., "An essay in the theory of uncontrollable flows and congestion", *Technical Report*, Department of Information and System Engineering, Faculty of Science and Engineering, Chuo University, TRISE 94-03 (1994).

[3] PHAM DINH, T. and T. H. LE THI, "Convex analysis approach to d.c. programming: theory, algorithms and applications", Acta Mathematica Vietnamica **22** (1997) 289–355.

[4] PHAM DINH, T., "Duality in d.c. (difference of convex functions) optimization. Subgradient methods, Trends in Mathematical Optimization, International Series of Numerical Mathematics **84** Birkhauser (1988) 277-293.

[5] Phillips, J. , "Algorithms for the vector maximization problem", *Mathematical Programming*, **2** (1972), 207-229.

[6] SHI, J. and Y. YAMAMOTO, "A global optimization method for minimum maximal flow problem", *ACTA Mathematica Vietnamica* **22** (1997) 271-287.

[7] SHIGENO, M., I. TAKAHASHI and Y. YAMAMOTO, "MMinimum maximal flow problem: an optimization over the efficient set", *Journal of Global Optimization* **25** (2003) 425-443.

[8] Yamamoto, Y. "Optimization over the efficient set: Overview", *Journal of Global Optimization* **1-4** (2001) 285-317.