East-West J. of Mathematics: Vol. 9, No 2 (2007) pp. 125-138

D.C. OPTIMIZATION METHODS FOR SOLVING MINIMUM MAXIMAL NETWORK FLOW PROBLEM

Le Thanh Hue and **Nguyen Hoan Vu**

Faculty of Informatic Technology
University of Mining and Geology, Hanoi, Vietnam.
E-mail: huelt63@yahoo.com

Faculty of Mathematics-Mechanics-Informatics, University of Science
Vietnam National University, Hanoi, Vietnam.
E-mail: nghoanvu060753@yahoo.com

Abstract

We consider the minimum maximal flow problem, i.e., minimizing the flow value among maximal flow, which is an $\mathcal{NP}$-hard problem. This problem is formulated as an optimization problem over the Pareto set of a linear vector program. We use a d.c. optimization formulation of the problem to obtain solution methods for the problem. Two solution approaches are described. The first one is a global optimization method based upon a branch-and-bound strategy. The second is a local search technique based upon a d.c. optimization algorithm.

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

Key words: minimum maximal flow, d.c. optimization, smooth variational inequality, branch-and-bound, DCA algorithm
(2000) AMS Mathematics Subject Classification: C90 C29

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

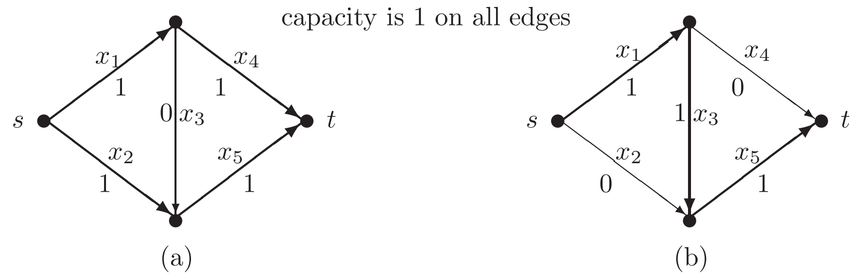

Figure 1: *Minimum maximal flow problem.*

case (a) of the figure illustrates the controllable flow, i.e., we freely increase and decrease each arc flow as long as the conservation equations and capacity constraints are kept satisfied. On the other hand, if the given arc flow is fixed and we cannot reduce it by some reasons, then the network cannot be exploited at the most economical situation. In the case (a), if we reduce flow on the edge x_3 to 0, 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 fact that the flow value on x_3 is not controllable. It means that the maximum flow value is not attainable in some cases. For example, if the users on a highway network are disobedient, such a case might occur.

From the point of view of modeling, the two cases are essentially different even though there is a little resemblance between the two cases. If the flow is controllable, we want to find an optimal value of flow, that is the case (a). In the case (b), the flow value is also optimal in the sense that one can not reduce the value on x_3. The standard network-flow with the controllability has been well studied for several decades. Without the controllability, many problems in network-flow, e.g., the maximum flow problem, become more difficult. By contrast with the standard network-flow theory, uncontrollable network-flow theory is a new field, hence is still in its infancy.

In this paper we consider minimum maximal flow (MMF) problem which finds out the minimum-value among the *maximal flow* of the network $\mathcal{N}$. Iri [6] 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 assumptions. In Iri's profound essay, several fundamental theorems and new research topics are described, but no algorithms for the corresponding problems are proposed there. To the authors' knowledge, no algorithms for MMF were known until Shi-Yamamoto [12]. As pointed out in [14], Shi-Yamamoto's algorithm is not efficient enough. After that, some algorithms for solving the problem were proposed by Shigeno-Takahashi-Yamamoto in [13] and others. In this paper we focus on the development of algorithm for MMF in virtue of d.c. optimization methods.

In next section 2, we present the problem and its equivalent formulations. In section 3, we discuss some related properties of d.c. programming and d.c. algorithm. Section 4 is devoted to describe two solution methods for the problem. The first one is based upon a branch-and-bound strategy that can find a global optimal solution to the problem. The second one is based upon a d.c. optimization local search technique, called DCA. The latter can find only a local solution but it works well for practical problems. Finally, a brief concluding remark is given in Section 5.

2 Equivalent Formulations

Consider a directed network $\mathcal{N}(V, E, s, t, c)$, where V is the set of m nodes with two designated nodes source s and sink t, E is the set of n arcs, and c is the vector of capacities on arcs. A vector $x = (\ldots, x_h, \ldots) \in R^n$ is said to be a *feasible flow* if it satisfies the conservation equations and capacity constraints:

$$\sum_{h\in\Delta^+(i)} x_h = \sum_{h\in\Delta^-(i)} x_h \quad \forall i \in V \setminus \{s,t\}$$
$$0 \le x \le c < \infty, \text{ and } c > 0.$$

where $\Delta^+(i)$ and $\Delta^-(i)$ are the sets of arcs which leaves and enters the node i, respectively. We define the $(m \times n)$ matrix $A = [a_{ih}]_{h\in E}^{i\in V\setminus\{s,t\}}$, called the node-arc incidence matrix, by

$$a_{ih} = \begin{cases} 1 & \text{if } h \in \Delta^+(i) \\ -1 & \text{if } h \in \Delta^-(i) \\ 0 & \text{otherwise.} \end{cases}$$

The conservation equation is then simply written as $Ax = 0$. Let X denote the set of feasible flow, i.e.,

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

Obviously, X is a compact convex set. A vector $z \in X$ is said to be *maximal flow* if there does not exist $x \in X$ such that $x \ge z$ and $x \ne z$. Denote by f the flow value function which is defined for the source node by

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

Then, f is a linear function. Let $d^\top \in R^n$ with

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

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

Throughout this paper R^k denotes the set of k-dimensional real column vectors, $R^k_+ = \{x \mid x \in R^k; x \geq 0\}$ 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 similarly. 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 with appropriate dimension. For a set S, $V(S)$ is the set of extreme points of S.

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

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

The problem to be considered is given as

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

Denote by X_E the efficient set of the vector optimization problem

$$\text{vmax } \{x \mid x \in X\}.$$

We recall that a point x^* is an efficient point of X if there does not exist $\tilde{x} \in X$ such that $\tilde{x}_i \geq x^*_i \ \ \forall i = 1, \ldots, n$ and $\tilde{x} \neq x^*$. In this case X_M coincides with the efficient set of X. Then (P) is equivalent to the problem

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

2.1 Primal Formulation

Following Benson [3] we define a function r as

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

It follows from [8] that

$$\text{dom}(r) = X + R^n_-.$$

Clearly, $r(x) \geq 0$ 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

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

where $y_{r(x)} \in \arg\max\{\, e(y - x) \mid y \geq x, y \in X\}$. Moreover, by taking the dual problem of the linear program defining $r(x)$, we can see that r is piecewise-linear on X. In fact, adding a slacks u and v such that

$$\begin{pmatrix} A & 0 & 0 \\ I & I & 0 \\ I & 0 & -I \end{pmatrix} \begin{pmatrix} y \\ u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ c \\ x \end{pmatrix}, u, v \in R^n_+, \iff Ay = 0, x \leq y \leq c.$$

Then for a given $x \in R^n$, $r(x)$ is the optimal value of the following linear program

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

where I is an $(n \times n)$ unit matrix.

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

Proof This is trivial from (2.4). $\square$ 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

$$\min\{dx \mid x \in X, r(x) \leq 0\}. \tag{2.6}$$

Consider the following penalized problem for a fixed number t.

$$P(t) \quad \min\{dx + tr(x) \mid x \in X\}. \tag{2.7}$$

If $V(X) \subset \{x \in X \mid r(x) \leq 0\}$ we set $t_* = 0$, otherwise we set

$$\begin{aligned}
t_* &= \frac{\max\{dx \mid x \in X\} - \min\{dx \mid x \in X\}}{\min\{r(x) \mid x \in V(X) \setminus X_E\}} \\
&= \frac{\max\{dx \mid x \in X\} - \min\{dx \mid x \in X\}}{\min\{r(x) \mid x \in V(X), r(x) > 0\}} \geq 0.
\end{aligned} \tag{2.8}$$

We have (see [9])

Lemma 2. *For every $t > t_*$, the solution-sets of problem (2.6) and (2.7) coincide.*

Note that if $r(x)$ is integral for any integer x and feasible flow, then we can take

$$t_* = \max\{dx \mid x \in X\} - \min\{dx \mid x \in X\} \tag{2.9}$$

Denote by δ_X the indicator of X, i.e.,

$$\delta_X(x) = \begin{cases} 0 & \text{if } \ x \in X \\ +\infty & \text{if } \ x \notin X, \end{cases}$$

and $g(x) := dx + \delta_X(x)$,

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

Then $g(x)$ and $h(x)$ are 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. optimization problem. Hereafter, we use this primal d.c. formation for local search in DCA.

2.2 A Dual formulation

From a result of Philip [11] 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 under consideration 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 equilibrium variational constraints (see e.g. [7])

Let

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

and

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

Then we have the following result;

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

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

Proof It is easy to see that

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

Note that X is a convex set. Suppose that (2.11) holds for some $x \in X$ and some $\lambda \in \Lambda$. Then 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 $\widehat{g}(x,\lambda) - \widehat{h}(x,\lambda) = 0$.

Suppose that $\widehat{g}(x,\lambda) - \widehat{h}(x,\lambda) = 0$ 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 $\widehat{g}$ and $\widehat{h}$ are convex and differentiable.

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

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

From Shigeno-Takahashi-Yamamoto [13], we see that if $\mathcal{N}$ is acyclic then Λ in (2.17) could be replaced by

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

Then one can take the above set as Λ to design algorithms.

Another dual d.c. formulation can be found in [1].

3 Properties of d.c programming and DCA

DCA (D.C algorithm) [10] is a prime-dual approach for finding local optimum in d.c. programming. More detailed results on DCA can be found in such as [9]. 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 lower 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 $+\infty = +\infty - (+\infty)$. The set of 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

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

Notice that g and h are lsc, 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 $v_p = v_d$ (see [9]).

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 $\varepsilon > 0$.*

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 results (see also [9]).

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

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

From the above Lemmas, we have

Theorem 7. *Suppose that h is a piecewise-linear convex function on* $\mathrm{dom}(h)$*. Then $x^\star$ is local minimal for $g-h$ if and only if $\partial h(x^\star) \subseteq \partial g(x^\star)$.*

Note that in problem (2.7) $g(x) = dx$ and $h(x) = -t_* r(x)$. Since X is polyhedral, the function $r(x)$ defined by (2.4) is piecewise linear concave. Now we can describe the framework of DCA. This algorithm in general converges to a stationary point which may be not a local minimum.

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

Lemma 8. *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 Suppose that x^{k-1} and y^{k-1} are in hand. We have

$$\begin{aligned} & \min\{g(x) - (h(x^{k-1}) + \langle x - x^{k-1}, y^{k-1}\rangle) \mid x \in R^n\} \\ = \; & \min\{g(x) - \langle x, y^{k-1}\rangle \mid x \in R^n\} - h(x^{k-1}) + \langle x^{k-1}, y^{k-1}\rangle \end{aligned} \tag{3.20}$$

and

$$\begin{aligned} & \min\{h^*(y) - (g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle) \mid y \in R^n\} \\ = \; & \min\{h^*(y) - \langle x^k, y\rangle \mid y \in R^n\} - g^*(y^{k-1}) + \langle x^k, y^{k-1}\rangle. \end{aligned} \tag{3.21}$$

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 9. *The values $g(x^k) - h(x^k)$ 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 the sequence $\{x^k\}_{k=1,2,\cdots}$ converges to a 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

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

The statement that any cluster point $x^\star$ of the sequence $\{x^k\}$ is a stationary point, i.e., $\partial h(x^\star) \cap \partial g(x^\star) \neq \emptyset$, follows from (iii) of Theorem 3 in [9]. $\square$

We recall that a point $x^\star$ for which $\partial h(x^\star) \cap \partial g(x^\star) \neq \emptyset$ is called a stationary point of $f = g - h$. Clearly, every local minimum of f is a stationary point. Generally, when we have $y^0 \in \partial h(x^\star)$ and $y^0 \notin \partial g(x^\star)$, then

$$h(x) \geq h(x^\star) + \langle x - x^\star, y^0 \rangle \tag{3.22}$$

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 (3.22) 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 $\partial h(x^0) \nsubseteq \partial g(x^0)$ then we can find a smaller value at point x^1.

4 Description of the Algorithm

Now we go back to problem (P). In this section, we propose two algorithms for solving the problem. The first algorithm is based upon the following general framework of branch-and-bound strategy.

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 such that

$$\mathrm{int}(S_1) \cap \mathrm{int}(S_2) = \emptyset,\; S_1 \cup S_2 = S.$$

Taking into account the convergence of the algorithms we need the division to be *exhaustive*, i.e., any nested sequence of simpleces generated by this bisection tends to a singleton, that is $\lim_{k\to\infty} \cap_{k=1}^{\infty} S_k = x^0$ for some x^0. To obtain an exhaustive simplex-bisection, at each step k, we can devide a simplex S_k into two smaller ones by bisecting, for example, the longest edge of S_k via its middpoint. The sequence $\{S_k\}_{k=1,2,\cdots}$ generated by this way is exhaustive (see eg. [5]).

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 approximate a stationary point that will be used to improve the upper bound in the branch and bound algorithm.

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 convexity of $h(x)$, we have $l_i(x) \geq h(x)$ for all $x \in S_i$. Then

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

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

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

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

algorithm MMFDCA

step 0: let $\varepsilon > 0$ and S_0 be a simplex such that $X \subseteq S_0$. let $x^0 := 0, y^0 := (-1, \cdots, -1)$, $b^U := \text{basiDCA}(X), b_L := \min\{dx \,|\, Ax = 0, 0 \leq x \leq c\}, M := \{S_0\}$

step 1: select $S_0 \in M$ such that $b_L = L(X \cap S_0)$ and divide S_0 into two simpleces S_1 and S_2,

step 2: $u^i := \text{basiDCA}(X \cap S_i)$ 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 $b^U - b^L \leq \varepsilon$ then Stop, b^U is an $\varepsilon-$global optimal value.

step 4: $M := \{S \in M \,|\, L(X \cap S) < b^U\}$, if $M = \emptyset$ then Stop, $b^U = u^i$ is an $\varepsilon-$global optimal value, and x^i, for which $u^i = \text{basiDCA}(X \cap S_i)$ is an $\varepsilon-$global optimal solution. Otherwise go to Step 1.

The convergence of the above algorithm follows from the exhaustive partition. Here we omit the detailed proof. A general proof under such exhaustiveness can be found in many books, such as [5].

DCA for mmf problem It is well known that (see, e.g. [5, 4] and the references therein) global optimization methods can solve only problems with moderate sizes. Unfortunately, practical minimum maximal flow problems often are large-scale. For this case, a local optimization method should be used.

In the rest part of this section we will use DCA for solving the minimum maximal flow problem by using its primal d.c. formulation described in Section 2. Note that, in contrast to general case, for the minimum maximal flow problem the exact penalty parameter t_* can be easy to compute by (2.9) for integral flow and by (2.8) for general case.

Now fixed $t > t_*$ and consider the unconstrained d.c. problem

$$\min\{dx + \delta_X(x) + tr(x) \,|\, x \in X\}$$

Let

$$g(x) := dx + \delta_X(x), \quad h(x) = -tr(x)$$

as before. Since $-r(x)$ is the optimal value of the linear program

$$\left| \begin{array}{ll} ex + \min & -ey \\ \quad s.t. & \begin{pmatrix} A & 0 & 0 \\ I & I & 0 \\ I & 0 & -I \end{pmatrix} \begin{pmatrix} y \\ u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ c \\ x \end{pmatrix}, \\ & y \geq 0, u \geq 0, v \geq 0, \end{array} \right. \tag{4.23}$$

the convex function $-r(x)$ is subdifferentiable at every point $x \in X$. From the duality of linear programming, by taking the dual problem of the linear program defining $-r(x)$, we have

$$(LP(x)) \qquad \left| \begin{array}{rl} -r(x) = ex + \max & \Big(c\eta + x\gamma\Big) \\ s.t. & A^\top \xi + \eta + \gamma \leq -e, \\ & \eta \leq 0, -\gamma \leq 0. \end{array} \right.$$

Let $s(x) = (\xi(x), \eta(x), \gamma(x))$ be an optimal solution of this linear program, and $s(z) = (\xi(z), \eta(z), \gamma(z))$ be an optimal solution to $LP(z)$. Then for any $x \in X$ and $z \in X$, we have $c\eta(z) + z\gamma(z) \geq c\eta(x) + z\gamma(x)$ and

$$\begin{array}{rcl} -r(z) + r(x) & = & \langle z - x, e\rangle + \langle \eta(z) - \eta(x), c\rangle - x\gamma(x) + z\gamma(z) \\ & \geq & \langle z - x, e\rangle + z\gamma(x) - z\gamma(z) - x\gamma(x) + z\gamma(z) \\ & = & \langle z - x, e + \gamma(x)\rangle. \end{array}$$

It implies that $e + \gamma(x) \in \partial(-r(x))$. Hence $t(e + \gamma(x)) \in \partial h(x)$. This vector $t(e+\gamma(x))$ can serve as y^k in Step 1 of BASIDCA. We rewrite Algorithm BASIDCA with employing the character of mmf problem as follows.

algorithm DCA4MMF

step 0: choose $t > t_*$, pick $x^0 \in X$ and solve Linear program (LP(x^0)) to obtain an optimal solution $s(x^0) = (\xi(x^0), \eta(x^0), \gamma(x^0)$. Take $y^0 = t(e + \gamma(x^0))$, $k = 1$;
step 1: solve $\min\{dx + tr(x^{k-1}) - \langle y^{k-1}, x - x^{k-1}\rangle \,|x \in X\}$ to obtain an optimal solution x^k.

solve ($LP(x^k)$) to obtain an optimal solution $s(x^k) = (\xi(x^k), \eta(x^k), \gamma(x^k))$;
take $y^k = t(e + \gamma(x^k))$.
step 2: if $\langle d - y^k, x^k\rangle = \min\{\langle d - y^k, x\rangle \mid x \in X\}$, then terminate;
x^k is a stationary point to the problem;
otherwise, $k := k + 1$ goto step 1.

Convergence

Since r is piecewise linear, by Theorem 7 and Lemma 9 we have the following convergence result.

Theorem 10. *If Algorithm* DCA4MMF *terminates at some iteration k, then x^k is a stationary point. If the algorithm does not terminate in finitely many iterations, it genetates an infinite sequence $\{x^k\}$ such that its every accumulation point is a stationary point of Problem (P).*

Proof If the algorithm terminate at some iteration k, we have $\langle d - y^k, x\rangle \geq \langle d - y^k, x^k\rangle$ for all $x \in X$. It yields $dx \geq dx^k + \langle x - x^k, y^k\rangle$ for all $x \in X$. Thus $y^k \in \partial g(x^k)$. Since $y^k = t(e + \gamma(x^k))$, we have $y^k \in \partial h(x^k)$. Hence $y^k \in \partial h(x^k) \cap \partial g(x^k)$ which means that x^k is a stationary point.

If the algorithm does not terminate in finitely many iterations, it generates an infinite sequence $\{x^k\}$. By Lemma 9 we see that every accumulation point of the sequence is a stationary point of Problem (P). □

5 Conclusion

We have formulated the minimum maximal flow problem as a piecewise d.c. optimization problem by using an exact penalty function technique. We have proposed two solution-approaches to the latter problem. The first one is a global optimization method based upon a branch-and-bound strategy. The second one is a local search using a d.c. optimization method. The both methods strongly employ special structure of the minimum maximal network flow.

References

[1] Le Thi H. A., Pham D. T. and L. D. Muu , *Numerical solution for optimization over the efficient set by d.c. optimization algorithms*, Operations Research Letters **19** (1996), 117-128.

[2] Le Thi H. A., Pham D. T. and L. D. Muu, *A unified d.c. optimization approach to optimization over the weakly efficient and efficient sets,* Journal of Optimization Theory and Applications **117**(2003), 503-531.

[3] R. Benson, *Optimization over the efficient set,* Journal of Mathematical Analysis and Its Applications, **98**(1984) 562-580.

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

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

[6] M. Iri, *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).

[7] Z. Q. Luo, J. S. Pang and D. Ralph, "Mathematical Program with Equilibrium Constraints", CAMBRIDGE Uninersity Press, 1996.

[8] L. D. Muu and L. T. Luc, *Global optimization approach to optimizing over the efficient set,* In Lectures Notes in Economics and Mathematical Systems **542** (Springer 1997), 182-195.

[9] Pham D. T. and Le Thi H. A., *Convex analysis approach to d.c. programming: theory, algorithms and applications,* Acta Mathematica Vietnamica **22**(1997), 289-355.

[10] D. T. Pham, *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.

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

[12] J. Shi and Y. Yamamoto, *A global optimization method for minimum maximal flow problem,* ACTA Mathematica Vietnamica **22**(1997), 271-287.

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

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