Vietnam Journal of Mathematics **39**:1(2011) 31-48

Vietnam Journal of MATHEMATICS

Smooth Optimization Algorithms for Optimizing Over the Pareto Efficient Set and Their Applications to Minmax Flow Problems *

Le Dung Muu[1] and Le Quang Thuy [2]

[1] *Institute of Mathematics, Hanoi*

[2] *Faculty of Applied Mathematics and Informatics, Hanoi University of Technology*

Received January 12, 2010
Revised January 24, 2011

Abstract. We propose smooth optimization algorithms for optimizing a real valued function over the efficient Pareto set of a linear multicriteria optimization problem. Application to minmax flow problems is discussed. Some computational results are reported.

2000 Mathematics Subject Classification: 90C29, 90C26.

Key words: Optimization over the efficient set, smooth optimization, regularization, algorithms, minmax flow problem.

1. Introduction

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a real valued continuous function on $\mathbb{R}^n, C$ be a $p \times n$ real matrix and X be a bounded convex polyhedral set which is defined by a system of linear equalities and/or inequalities. We consider the optimization problem over the efficient Pareto set

* This paper is supported by the Vietnam National Foundation for Science and Technology Development (NAFOSTED).

$$\min f(x), \quad \text{subject to } x \in X_E, \tag{1}$$

where X_E denotes the efficient Pareto solution set of the multiple objective linear programming problem

$$\text{Vmax } Cx, \quad \text{subject to } x \in X. \tag{2}$$

The problem (1) has some applications in decision making and has been considered in some research papers (see e.g., [1], [2], [5], [6], [9], [11], [14] and the references therein).

We recall that a vector $x^0 \in X$ is said to be an *efficient solution* of problem (2) if there is no $x \in X$ such that $Cx \geq Cx^0$ and $Cx \neq Cx^0$. It is well known that X_E consists of a union of faces of X, and therefore, in general, X_E is a nonconvex set. That is the reason why the problem (1), even with f linear, is a difficult global optimization one.

Some global optimization algorithms are proposed for solving the problem (1) (see e.g., [1], [5], [6], and the references quoted there). These algorithms can solve the problem (1) when the number of the criteria is relatively small (≤ 10). However, in some practical applications such as minmax flow problems, the number of the criteria is large. In this case, local optimization approaches should be used.

In [1] and [2] the problem (1) is formulated as a nonsmooth DC optimization program and the resulting problem is solved by a primal-dual algorithm.

In this paper, we propose a smooth optimization approach to the problem (1). Namely, we use a regularization technique widely used in variational inequality to obtain smooth and DC formulations for the problem. These formulations allow that the well developed smooth and DC optimization can be applied efficiently for solving the problem (1).

The paper is organized as follows. In Section 1 we show how to use the Yoshida regularization technique to obtain smooth optimization formulations for the problem (1). Sections 2 and 3 are devoted to description of the quadratic penalty and DC optimization algorithms for solving the problem. In Section 4 we apply the proposed algorithm to a minmax flow problem. Some computational results which show efficiency of the proposed algorithms are reported in the last section.

2. Smooth Optimization Formulations

From Philip [11] it is known that one can find a simplex $\Lambda \subset \mathbb{R}^n_{++}$, such that a point $x \in X_E$ if and only if there exists $\lambda \in \Lambda$ such that

$$x \in \text{argmax}\{\lambda^T Cx : \ x \in X\}.$$

Consequently,

$$X_E = \{x \in X : \lambda^T Cx \geq \phi(\lambda) \ \text{ with } \lambda \in \Lambda\}, \tag{3}$$

where

$$\phi(\lambda) := \max\{\lambda^T Cy : \ y \in X\}. \tag{4}$$

The simplex Λ can be defined as

$$\Lambda = \{\lambda \in \mathbb{R}^p_{++} : \lambda \geq e; \sum_{k=1}^{n} \lambda_k = M\},$$

for some $M \in \mathbb{R}$. From (3) it follows that $x \in X_E$ if and only if there exists $\lambda \in \Lambda$ such that $\langle -C^T\lambda, y - x\rangle \geq 0$ for all $y \in X$.

Let $c > 0$ play as a regularization parameter, and $K = \Lambda \times X$. For each $(\lambda, x) \in K$, we define

$$\gamma_c(\lambda, x) := \max_{y \in K}\{\langle -C^T\lambda, x - y\rangle - \frac{c}{2}\|x - y\|^2\}. \tag{5}$$

Since the objective function $\langle -C^T\lambda, x - y\rangle - \dfrac{c}{2}\|x - y\|^2$ is strongly concave on K with respect to y, the problem defining $\gamma_c(\lambda, x)$ has a unique solution that we denote by $y_c(\lambda, x) \in K$.

Proposition 2.1. *The following statements are true*

(i) $\forall\, (\lambda, x) \in \mathbb{R}^{p+n}$ *one has* $y_c(\lambda, x) = P_K\left(x + \dfrac{1}{c}C^T\lambda\right)$, *where* $P_K(x + \frac{1}{c}C^T\lambda)$ *is the projection of* $(x + \dfrac{1}{c}C^T\lambda)$ *on the set* K*;*

(ii) $\gamma_c(.,.)$ *is nonnegative and continuously differentiable on* K*;*

(iii) $(\lambda, x) \in K, \gamma_c(\lambda, x) = 0$ *if and only if* $x \in X_E$*;*

(iv) $\nabla\gamma_c(\lambda, x) = -[C(x - y_c(\lambda, x)), C^T\lambda + c(x - y_c(\lambda, x))]^T$.

Proof. (i) Since

$$\begin{aligned}
&\frac{1}{2c}\|C^T\lambda\|^2 - \frac{c}{2}\|x + \frac{1}{c}C^T\lambda - y\|^2 \\
=&\frac{1}{2c}\|C^T\lambda\|^2 - \frac{c}{2}(\|x - y\|^2 + \frac{2}{c}\langle C^T\lambda, x - y\rangle + \frac{1}{c^2}\|C^T\lambda\|^2) \\
=&\langle -C^T\lambda, x - y\rangle - \frac{c}{2}\|x - y\|^2,
\end{aligned}$$

we have

$$\langle -C^T\lambda, x - y\rangle - \frac{c}{2}\|x - y\|^2 = \frac{1}{2c}\|C^T\lambda\|^2 - \frac{c}{2}\|x + \frac{1}{c}C^T\lambda - y\|^2.$$

Then

$$\gamma_c(\lambda, x) = \max_{y \in K}\{\langle -C^T\lambda, x - y\rangle - \frac{c}{2}\|x - y\|^2, \ (\lambda, x) \in \mathbb{R}^{p+n}\}$$

$$= \max_{y\in K}\{\frac{1}{2c}\|C^T\lambda\|^2 - \frac{c}{2}\|x + \frac{1}{c}C^T\lambda - y\|,\ (\lambda, x) \in \mathbb{R}^{p+n}\}$$
$$= \frac{1}{2c}\|C^T\lambda\|^2 + \max_{y\in K}\{-\frac{c}{2}\|x + \frac{1}{c}C^T\lambda - y\|,\ (\lambda, x) \in \mathbb{R}^{p+n}\}$$
$$= \frac{1}{2c}\|C^T\lambda\|^2 - \frac{c}{2}\min_{y\in K}\{\|x + \frac{1}{c}C^T\lambda - y\|,\ (\lambda, x) \in \mathbb{R}^{p+n}\},$$

which implies that $y_c(\lambda, x) = P_K(x + \frac{1}{c}C^T\lambda)$.

(ii) It is easy to see that $y_c(\lambda, x) = P_K(x + \frac{1}{c}C^T\lambda)$. Since the projection operator is continuous, $y_c(.,.)$ is continuous on K. Then, from

$$\gamma_c(\lambda, x) = \langle -C^T\lambda, x - y_c(\lambda, x)\rangle - \frac{c}{2}\|x - y_c(\lambda, x)\|^2,$$

we can see that $\gamma_c(\lambda, x)$ is continuous on K.

(iii) Let $(\lambda^*, x^*) \in K$ be such that $\gamma_c(\lambda^*, x^*) = 0$. Then

$$\langle -C^T\lambda^*, x^* - y_c(\lambda^*, x^*)\rangle - \frac{c}{2}\|x^* - y_c(\lambda^*, x^*)\|^2 = 0,$$

which implies

$$\langle -C^T\lambda^*, x^* - y_c(\lambda^*, x^*)\rangle = \frac{c}{2}\|x^* - y_c(\lambda^*, x^*)\|^2 \geq 0. \tag{6}$$

On the other hand, by the well known optimality condition for convex programming applying to the problem defining $\gamma_c(\lambda^*, x^*)$ we obtain

$$\langle -C^T\lambda^* + c[y_c(\lambda^*, x^*) - x^*], x - y_c(\lambda^*, x^*)\rangle \geq 0\ \forall\ (\lambda, x) \in K.$$

With $x = x^*$ we have

$$-C^T\lambda^* + c[y_c(\lambda^*, x^*) - x^*], x^* - y_c(\lambda^*, x^*)\rangle \geq 0.$$

Thus

$$\langle -C^T\lambda^*, x^* - y_c(\lambda^*, x^*)\rangle - c\|x^* - y_c(\lambda^*, x^*)\|^2 \geq 0.$$

Combine this inequality with (5) we can deduce that $\|x^* - y_c(\lambda^*, x^*)\| \leq 0$. Hence $x^* = y_c(\lambda^*, x^*)$. By (i) we have $x^* = y_c(\lambda^*, x^*) = P_K(x^* + \frac{1}{c}C^T\lambda^*)$.

On the other hand, according to properties of the projection $P_K(.)$ we know that $x^* = P_K(x)$ if and only if $\langle y - x^*, x^* - x\rangle \geq 0$ for all $y \in K$. From the last inequality, by replacing $x = x^* + \frac{1}{c}C^T\lambda^*$ we obtain $\langle -C^T\lambda^*, y - x^*\rangle \geq 0$ for all $y \in K$, which implies $x^* \in X_E$.

Conversely, assume that $x^* \in X_E$. Then $\langle -C^T\lambda^*, y - x^*\rangle \geq 0$ for all $y \in K$. Note that

$$\gamma_c(\lambda^*, x^*) = \max_{y\in K}\{\langle -C^T\lambda^*, x^* - y\rangle - \frac{c}{2}\|x^* - y\|^2\}$$
$$= \max_{y\in K}\{-\langle -C^T\lambda^*, y - x^*\rangle - \frac{c}{2}\|x^* - y\|^2\} \leq 0.$$

Thus $\gamma_c(\lambda^*, x^*) = 0$.

(iv) Let $u = (\lambda, x)$ and $\varphi(u, y) = \langle C^T\lambda, x - y\rangle + \frac{c}{2}\|x - y\|^2$. Since the function $\varphi(u, .)$ is strongly convex and continuously differentiable with respect to the variable y, there exists a unique solution $y_c(\lambda, x)$ of the problem

$$\min_{(0,y)\in K} \varphi(u, y)$$

and the function $\gamma_c(\lambda, x)$ is continuously differentiable. A simple computation yields

$$\begin{aligned}\nabla\gamma_c(\lambda, x) &= -\nabla_u\varphi(u, y) = -\nabla_u\varphi(u, y_c(\lambda, x)) = -\nabla_{(\lambda,x)}\varphi((\lambda, x), y_c(\lambda, x)) \\ &= -[C(x - y_c(\lambda, x)), C^T\lambda + c(x - y_c(\lambda, x))]^T.\end{aligned}$$

■

By virtue of this proposition the problem (1) can be written equivalently as

$$\begin{aligned}&\min f(x), \\ &\text{subject to} \begin{cases} (\lambda, x) \in K, \\ \gamma_c(\lambda, x) = 0. \end{cases}\end{aligned} \tag{7}$$

Let $u = (\lambda, x)$, $g(u) = g(\lambda, x) = f(x)$. Then the problem (7) becomes

$$\begin{aligned}&\min\ g(u), \\ &\text{subject to } \begin{cases} u \in K, \\ \gamma_c(u) = 0. \end{cases}\end{aligned} \tag{8}$$

Since γ_c, by Proposition 2.1, is differentiable, this is a smooth optimization problem if the objective function g is smooth. Now we are going to discuss several methods for this smooth optimization problem.

3. Quadratic Penalty and Smooth DC Optimization Algorithms

Note that the problem (8), even with linear g and convex polyhedral K, is a nonconvex program, because of nonconvexity of the constraint $\gamma_c(u) = 0$. In this section we describe two algorithms, which seem suitable for the problem (8).

3.1. Quadratic Penalty Algorithm

A well known tool to handle the difficult constraint $\gamma_c(u) = 0$ is the penalty function (see e.g., [7], [12]). More precisely, we consider the quadratic penalized problem corresponding to the problem (8)

$$\alpha(t) := \min\{G_t(u) = g(u) + \tfrac{t}{2}\gamma_c^2(u)\}, \quad \text{subject to } u \in K. \tag{9}$$

The following theorem shows a relationship between the penalized problem (9) and the problem (8).

Theorem 3.1. ([7]) *Suppose that g is continuous on K. Let u^k be the global optimal solution of the penalized problem*

$$\min\{G_{t_k}(u) = g(u) + \tfrac{t_k}{2}\gamma_c^2(u)\}, \quad \textit{subject to } u \in K. \tag{10}$$

Suppose that $\{t_k\}$ is an increasing sequence of positive numbers such that $t_k \to +\infty$ as $k \to +\infty$. Then every limit point u^* of the sequence $\{u^k\}$ is a global optimal solution to the problem (8).

Since (9) is a nonconvex program, finding its global optimal solutions is difficult. Instead, we propose an algorithm that can find only a stationary point. It is well known from the Lagrange optimality condition that, under a certain regularity condition, if u^* is an optimal solution to the problem (8), then u^* is a stationary point. Here we call u^* a stationary point for (8) if there exists a Lagrange multiplier μ_* such that the relation

$$0 \in \nabla g(u^*) + \mu^* \nabla\gamma_c(u^*) + N_K(u^*)$$

holds true.

Algorithm(Quadratic penalty algorithm)

- *Initialization*

  Choose two sequences of positive numbers $\{t_k\}$, $\{\tau_k\}$ such that $t_{k+1} > t_k$, $t_k \to +\infty$ and $\tau_{k+1} < \tau_k$, $\tau_k \to 0$ and a tolerance $\epsilon \geq 0$.

  Find a point $u^0 \in K$.
- *Iteration k* ($k = 1, 2, 3, \cdots$)

  $*$ *Step k_1*

  Let $u^{k_0} = u^{k-1}$ and go to iteration j.

  $*$ *Iteration j* ($j = 0, 1, 2, \cdots, j_k$)

  – *Step j_1*. If $\|\nabla G_{t_k}(u^{k_j}) + d^{k_j}\| \leq \tau_k$, for some $d^{k_j} \in N_K(u^{k_j})$, then let $u^k = u^{k_j}$ and go to *Step k_2*.

    Otherwise, go to *Step j_2*.

  – *Step j_2*. Find $v^{k_j} \in \operatorname{argmin}\{\langle \nabla G_{t_k}(u^{k_j}), u - u^{k_j}\rangle,\ u \in K\}$

    If $\langle \nabla G_{t_k}(u^{k_j}), v^{k_j} - u^{k_j}\rangle \geq 0$, then terminate: $u^k = u^{k_j}$ is an stationary point.

    Else, let $h^{k_j} = v^{k_j} - u^{k_j}$, $u^{k(j+1)} = u^{k_j} + \alpha_{k_j} h^{k_j}$, where

$$\alpha_{k_j} \in \text{argmin}\{\Phi(\alpha) = G_{t_k}(u^{k_j} + \alpha h^{k_j}),\ \alpha \in [0,1]\}.$$

Let $j := j+1$, and return to step j.

$*$ *Step* k_2

If $\|\nabla G_{t_k}(u^k) + d^k\| \leq \epsilon$, for some $d^k \in N_K(u^k)$, then terminate: u^k is an ϵ - stationary point.

Else let $u^{(k+1)_0} = u^k$ and go to iteration k with $k := k+1$.

Notice for performance of the algorithm:
* To find the starting point $u^0 = (\lambda^0, x^0) \in K$ we can use a method of the linear programming.
* At each iteration j, we need to calculate $\nabla G(u^{k_j})$. Since

$$G_{t_k}(u) = g(u) + \frac{t_k}{2}\gamma_c^2(u),$$

we have

$$\nabla G(u^{k_j}) = \nabla g(u^{k_j}) + t_{k_j}\nabla\gamma_c(u^{k_j})\gamma_c(u^{k_j}).$$

To calculate $\nabla\gamma_c(u^{k_j})$ we have to solve the strongly convex quadratic program

$$\min_{y \in K} \|y - (x^k + \frac{1}{c}\lambda^k)\|^2.$$

Assuming that f is continuously differentiable on an open convex set containing K we have the following theorem for convergence of the proposed algorithm.

Theorem 3.2. *Suppose that $\tau_k \to 0, t_k \to +\infty$ and that the sequence $\{\|d^k\|\}$ is bounded. Then, for every limit point u^* of $\{u^k\}$ satisfying the regular condition $\nabla\gamma_c(u^*) \neq 0$, there exists $\mu^* \in \mathbb{R}$ such that*

$$0 \in \nabla g(u^*) + \mu^*\nabla\gamma_c(u^*) + N_K(u^*).$$

Proof. Let u^* be a limit point of the sequence $\{u^k\}$ for which the regular condition is satisfied. Without loss of generality, by taking subsequences if necessary, we may assume that $u^k \to u^*$.

First, we show that u^* is a feasible point, which means that $\gamma_c(u^*) = 0$ and $u^* \in K$.
In fact, for $d^k \in N_K(u^k)$, one has

$$t_k\|\gamma_c(u^k)\nabla\gamma_c(u^k)\| \leq \|t_k\gamma_c(u^k)\nabla\gamma_c(u^k) + \nabla g(u^k) + d^k\| + \|\nabla g(u^k) + d^k\|.$$

Since

$$\nabla G_{t_k}(u^k) = \nabla g(u^k) + t_k\gamma_c(u^k)\nabla\gamma_c(u^k),$$

it follows that

$$t_k\|\gamma_c(u^k)\nabla\gamma_c(u^k)\| \leq \|\nabla G_{t_k}(u^k) + d^k\| + \|\nabla g(u^k) + d^k\| \leq \tau_k + \|\nabla g(u^k) + d^k\|.$$

The last inequality implies

$$\|\gamma_c(u^k)\nabla\gamma_c(u^k)\| \le \frac{1}{t_k}(\tau_k + \|\nabla g(u^k) + d^k\|). \tag{11}$$

Taking the limit in both sides of the inequality (11), by the boundedness of $\|d^k\|$, we obtain $\gamma_c(u^*)\nabla\gamma_c(u^*) = 0$, from which, since $\nabla\gamma_c(u^*) \neq 0$, it follows that $\gamma_c(u^*) = 0$. In addition, since $u^k \in K$ and K is closed, $u^* \in K$.

Next, we show that there exists $\mu^* \in \mathbb{R}$ such that

$$0 \in \nabla g(u^*) + \mu^*\nabla\gamma_c(u^*) + N_K(u^*).$$

Indeed, from the step j_1 of the algorithm, we have $\|\nabla G_{t_k}(u^k) + d^k\| \le \tau_k$. Hence $\nabla G_{t_k}(u^k) + d^k \in B(o, \tau_k)$, where $B(o, \tau_k)$ stands for the ball centered at o with radius τ_k.

Let $\mu_k = t_k\gamma_c(u^k)$ and

$$w^k := \nabla g(u^k) + t_k\gamma_c(u^k)\nabla\gamma_c(u^k) + d^k.$$

Then

$$\mu_k\nabla\gamma_c(u^k) = w^k - \nabla g(u^k) - d^k.$$

From the last equation it follows that

$$\mu_k\nabla\gamma_c(u^k)^T\nabla\gamma_c(u^k) = \nabla\gamma_c(u^k)^T(w^k - \nabla g(u^k) - d^k).$$

Since $\nabla\gamma_c(u^k) \to \nabla\gamma_c(u^*) \neq 0$, we have $\nabla\gamma_c(u^k)^T\nabla\gamma_c(u^k) \neq 0$ for every k large enough. Then

$$\mu_k = [\nabla\gamma_c(u^k)^T\nabla\gamma_c(u^k)]^{-1}\nabla\gamma_c(u^k)^T(w^k - \nabla g(u^k) - d^k).$$

Letting $k \to \infty$, by the boundedness of $\|d^k\|$ and $w^k \in B(o, \tau_k)$ with $\tau_k \to 0$, we have

$$\mu_k = t_k\gamma_c(u^k) \to [\nabla\gamma_c(u^*)^T\nabla\gamma_c(u^*)]^{-1}\nabla\gamma_c(u^*)^T(-\nabla g(u^*) - d^*) := \mu^*.$$

On the other hand, from $\|\nabla G_{t_k}(u^k) + d^k\| \le \tau_k$, by replacing

$$\nabla G_{t_k}(u^k) = \nabla g(u^k) + t_k\gamma_c(u^k)\nabla\gamma_c(u^k)$$

and taking the limit in both sides, we obtain

$$0 \in \nabla g(u^*) + \mu^*\nabla\gamma_c(u^*) + N_K(u^*).$$

■

3.2. Smooth DC Optimization Approach

In [1 - 3] several DC formulations have been presented for the problem of optimizing a real valued function over the efficient Pareto set of a linear vector programming problem. The DC programs formulated there are nonsmooth. In this section we present a smooth DC formulation for the problem. By definition we have

$$\gamma_c(\lambda, x) = \max_{y\in X}\{\langle -C^T\lambda, x-y\rangle - \frac{c}{2}\|x-y\|^2, (\lambda, x) \in \mathbb{R}^{p+n}\}.$$

A simple computation shows that $\gamma_c(\lambda, x) = g(\lambda, x) - h(\lambda, x)$, where

$$g(\lambda, x) = \frac{1}{2}\|x\|^2 + \frac{1}{2}\|C^T\lambda\|^2 + \max_{y\in X}\{\langle C^T\lambda + cx, y\rangle - \frac{c}{2}\|y\|^2\}, \tag{12}$$

$$h(\lambda, x) = \frac{1}{2}\|C^T\lambda + x\|^2 + \frac{c}{2}\|x\|^2. \tag{13}$$

It is easy to see that both g and h are convex. Since the objective function of the maximization problem

$$\max_{y\in X}\{\langle C^T\lambda + cx, y\rangle - \frac{c}{2}\|y\|^2, (\lambda, x) \in \mathbb{R}^{p+n}\}$$

is strongly concave, the function g is differentiable (see e.g., [10], [15]). Clearly, h is differentiable.

Thus, the problem (1) can be converted into the DC constrained problem

$$\min\{f(x) : g(\lambda, x) - h(\lambda, x) \le 0, (\lambda, x) \in K\}. \tag{14}$$

For $t > 0$, we consider the penalized problem

$$\min\{f_t(\lambda, x) := f(x) + tg(\lambda, x) - th(\lambda, x) : (\lambda, x) \in K\}. \tag{15}$$

When f is a differentiable DC function, we can solve this penalized problem using the differentiability version of the DCA algorithm developed in [1], [4] as follows. For simplicity of notation we write $u = (\lambda, x)$.

Start from a point $u^0 = (\lambda^0, x^0) \in K$. At each iteration $k = 0, 1, ...$, having u^k we construct v^k and u^{k+1} by setting

$$v^k = \nabla th(u^k) = t[C(C^T\lambda^k + x^k), C^T\lambda^k + (1+c)x^k]^T,$$
$$u^{k+1} = \arg\min\{f(x) + tg(u) - \langle u - u^k, v^k\rangle :\ u \in K\}.$$

It is proved in [1], among others, that $f_t(u^{k+1}) \le f_t(u^k)$ for every k and that any cluster point of the sequence $\{u^k\}$ is a stationary point to the problem (15). We note that DCA can be used for solving nonsmooth DC optimization problems of the form

$$\min\{F(u) := G(u) - H(u) : u \in \mathbb{R}^n\},$$

where G and H are lower semicontinuous proper convex functions. For this nonsmooth optimization problem, the sequence of iterates is constructed by taking

$$v^k \in \partial H(u^k), \;\; u^{k+1} = \arg\min\{G(u) - \langle v^k, u\rangle\}.$$

Thus, the algorithm in this nonsmooth case crucially depends on the choice of $v^k \in \partial H(u^k)$. In our case, since H is differentiable, this difficulty can be avoided.

4. Application to a Minmax Flow Problem

In this section we apply the results presented in the preceding sections to solve the minimum maximal flow problem.

In the minmax flow problem to be considered, we are given a directed network flow $N(V, E, s, t, p)$, 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 $p \in \mathbb{R}^n$ is the vector of arc capacities.
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 $x \in \mathbb{R}^n$ is said to be a *feasible flow* if it satisfies the capacity constraints:

$$0 \le x_h \le p_h \text{ for each arc } h \in E \tag{16}$$

and conservation equations:

$$\sum_{\partial^+ h = v} x_h = \sum_{\partial^- h = v} x_h \text{ for each node } v \in V \backslash \{s, t\}. \tag{17}$$

Note that conservation equations can be simply written as

$$Ax = 0,$$

where the $m \times n$ matrix $A = (a_{vh})$ is the well-known node-arc incidence matrix, whose entry a_{vh} for each $(v, h) \in V \backslash \{s, t\} \times E$ is defined as

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

Then, the constraint (16), (17) becomes

$$Ax = 0, \quad 0 \le x \le p.$$

Let X denote the set of feasible flows, i.e.,

$$X = \{x \in \mathbb{R}^n : Ax = 0,\ 0 \le x \le p\}.$$

For each $x \in X$, the value of the flow x is given by

$$d^T x = \sum_{\partial^+ h = s} x_h - \sum_{\partial^- h = s} x_h, \tag{18}$$

where d is a n - dimensional row vector defined as

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

A feasible flow $x^0 \in X$ is said to be a *maximal flow* if there is no feasible flow $x \in X$ such that $x \ge x^0$ and $x \ne x^0$. We use X_E to denote the set of maximal flows. Then the minimal value flow problem on the set of maximal flows (*Minmax flow Problem*) is given as

$$\begin{aligned} &\min\ d^T x, \\ &\text{subject to } x \in X_E. \end{aligned} \tag{19}$$

Obviously, X_E is the set of all efficient Pareto solutions of the multiple objective linear program

$$\begin{aligned} &\text{Vmax } x, \\ &\text{subject to } x \in X. \end{aligned} \tag{20}$$

For this particular case, the number of the decision variables is just equal to the number of the criteria. As we have mentioned, one can find a simplex Λ such that a point $x \in X_E$ if and only if there is $\lambda \in \Lambda \subset \mathbb{R}^n_{++}$ such that

$$x \in \text{argmax}\{\lambda^T y\ : y \in X\}.$$

Thus,

$$X_E = \{x \in X : \lambda^T x \ge \phi(\lambda) \quad \text{for all } \lambda \in \Lambda\}, \tag{21}$$

where

$$\phi(\lambda) = \max\{\lambda^T y : y \in X\}. \tag{22}$$

For the minmax flow problem we have the following result [13].

Theorem 4.1. ([13]) *Let Λ be one of the following simplices*

$$\Lambda := \{\lambda \in \mathbb{Z}^n : e \le \lambda \le ne\}$$

or

$$\Lambda = \{\lambda \in \mathbb{R}^n : \lambda \ge e; \sum_{k=1}^n \lambda_k = n^2\},$$

where e is the vector whose every entry is one. Then x is a maximal folow if and only if there is $\lambda \in \Lambda$ such that

$$x \in \text{argmax}\{\lambda^T y : y \in X\}.$$

In this case the gap function $\gamma_c(\lambda, x)$ defined by (5) takes the form

$$\gamma_c(\lambda, x) = \max_{y \in K}\{\langle -\lambda, x - y\rangle - \frac{c}{2}\|x - y\|^2,\ (\lambda, x) \in \mathbb{R}^{2n}\}.$$

By a simple computation we have

$$\nabla\gamma_c(\lambda, x) = -[x - y_c(\lambda, x), \lambda + c(x - y_c(\lambda, x))]^T.$$

Let $K = \Lambda \times X$. Then the problem (19) can be formulated as

$$\begin{aligned} &\min\ d^T x, \\ &\text{subject to } \begin{cases} (\lambda, x) \in K, \\ g(\lambda, x) - h(\lambda, x) = 0, \end{cases} \end{aligned} \tag{23}$$

where, since C is the identity matrix, by (12) and (13), we have

$$g(\lambda, x) = \frac{1}{2}\|x\|^2 + \frac{1}{2}\|\lambda\|^2 + \max_{y \in X}\{\langle \lambda + cx, y\rangle - \frac{c}{2}\|y\|^2\}, \tag{24}$$

$$h(\lambda, x) = \frac{1}{2}\|\lambda + x\|^2 + \frac{c}{2}\|x\|^2. \tag{25}$$

Thus the algorithms described in the preceding section can be applied to the minmax flow problem (19).

4.1. DC Algorithm for the Minmax Flow Problem

Thanks to the particular property of the minmax flow problem under consideration, one can find a DC decomposition for the gap function γ_c such that the subproblems are strongly convex quadratic. We need the following result.

Theorem 4.2. *For any $c > 0$ fixed, there exists $\rho > 0$ such that the function*

$$f_c(u) = \frac{1}{2}\rho||u||^2 - \gamma_c(u)$$

is convex on K.

Proof. For the above minmax flow problem, the function $\gamma_c(u)$ can be written as

$$\gamma_c(u) = \frac{1}{2c}||\lambda||^2 - \frac{c}{2}\min_{y \in K}\{||x + \frac{1}{c}\lambda - y||^2\}.$$

Thus, we have

$$f_c(u) = \frac{1}{2}\rho||u||^2 - \frac{1}{2c}||\lambda||^2 + \frac{c}{2}\min_{y \in K}\{||x + \frac{1}{c}\lambda - y||^2\},$$

or

$$f_c(u) = (\frac{1}{2}\rho - \frac{1}{2c})||\lambda||^2 + \frac{1}{2}\rho||x||^2 + \frac{c}{2}\min_{y\in K}\{||x + \frac{1}{c}\lambda - y||^2\}.$$

Clearly, if $(\frac{1}{2}\rho - \frac{1}{2c}) > 0$, then

$$q(u) = (\frac{1}{2}\rho - \frac{1}{2c})||\lambda||^2 + \frac{1}{2}\rho||x||^2$$

is a convex function.
Now we show that

$$p(u) := p(\lambda, x) = \frac{c}{2}\min_{y\in K}\{||x + \frac{1}{c}\lambda - y||^2\}$$

is convex on K. Indeed, let $y^u, y^v \in K$ such that

$$p(u) = \frac{c}{2}||x^u + \frac{1}{c}\lambda^u - y^u||^2,$$

and

$$p(v) = \frac{c}{2}||x^v + \frac{1}{c}\lambda^v - y^v||^2.$$

Since K is convex, $\theta y^u + (1-\theta)y^v \in K$ whenever $0 \le \theta \le 1$. On the other hand,

$$p(\theta u + (1-\theta)v) = \frac{c}{2}\min_{y\in K}\{||\theta x^u + (1-\theta)x^v + \frac{1}{c}(\theta\lambda^u + (1-\theta)\lambda^v) - y||^2\}.$$

Hence

$$p(\theta u + (1-\theta)v) \le \frac{c}{2}||\theta x^u + (1-\theta)x^v + \frac{1}{c}(\theta\lambda^u + (1-\theta)\lambda^v) - (\theta y^u + (1-\theta)y^v)||^2.$$

We observe that the inequality

$$\theta||x^u + \frac{1}{c}\lambda^u - y^u||^2 + (1-\theta)||x^v + \frac{1}{c}\lambda^v - y^v||^2 \ge$$
$$||\theta x^u + (1-\theta)x^v + \frac{1}{c}(\theta\lambda^u + (1-\theta)\lambda^v) - (\theta y^u + (1-\theta)y^v)||^2$$

is equivalent to the one

$$\theta(1-\theta)\{||x^u + \frac{1}{c}\lambda^u - y^u||^2 + ||x^v + \frac{1}{c}\lambda^v - y^v||^2 - 2\langle x^u + \frac{1}{c}\lambda^u - y^u, x^v + \frac{1}{c}\lambda^v - y^v\rangle\} \ge 0.$$

It is easy to see that the latter inequality is true. Hence

$$\theta(1-\theta)||x^u + \frac{1}{c}\lambda^u - y^u) - (x^v + \frac{1}{c}\lambda^v - y^v)||^2 \ge 0,$$

which shows the convexity of f_c. ■

From the result of Theorem 4.2, the problem (23) can be converted to the DC problem

$$\alpha(t) := \min_{u\in K}\{G_t(u) = b^T u + \tfrac{1}{2}\rho||u||^2 - [\tfrac{1}{2}\rho||u||^2 - t\gamma_c(u)]\}, \qquad (26)$$

where $u = (\lambda, x), b = (0, d)$.

For simplicity of notation, we take

$$G(u) = b^T u + \frac{1}{2}\rho||u||^2,\ H(u) = \frac{1}{2}\rho||u||^2 - t\gamma_c(u).$$

Then the problem has the form

$$\min\{G(u) - H(u) : u \in K\}. \qquad (27)$$

The sequences $\{u^k\}$ and $\{v^k\}$ constructed by DCA for the problem (27) now look as

$$v^k \in\ \nabla H(u^k) = \rho u^k - t\nabla\gamma_c(u^k),$$

$$u^{k+1} = \arg\min\{b^T u + \frac{1}{2}\rho||u||^2 - \langle u - u^k, v^k\rangle :\ u \in K\}.$$

Thus, at each iteration k we have to solve the two strongly convex quadratic programs

$$\min_{y\in K} \|y - (x^k + \frac{1}{c}\lambda^k)\|^2 \qquad (28)$$

and

$$\min_{u\in K}\{b^T u + \frac{1}{2}\rho||u||^2 - \langle u - u^k, v^k\rangle\}. \qquad (29)$$

Let $y_c(\lambda^k, x^k)$ be the unique solution of (28). Then

$$v^k = \rho u^k - t\nabla\gamma_c(u^k) = \rho(\lambda^k, x^k) + t[x^k - y_c(\lambda^k, x^k), \lambda + c(x^k - y_c(\lambda^k, x^k))]^T.$$

It is proved in [1], among others, that $G_t(u^{k+1}) \le G_t(u^k)$ for every k, that if $u^{k+1} = u^k$, then u^k is a stationary point, and that any cluster point of the sequence $\{u^k\}$ is a stationary point to the problem (26). In the sequel we call u^k an ϵ - stationary point if $||u^{k+1} - u^k|| \le \epsilon$.

Algorithm

- *Initialization*: Choose a tolerance $\epsilon > 0$ and find a point $u^0 \in K$.
  Let $k := 0$.
- *Iteration* k ($k = 0, 1, 2, ...$)
  ∗ Calculate $v^k = \rho u^k - t\nabla\gamma_c(u^k)$.
  ∗ Compute

$$u^{k+1} = \arg\min\{b^T u + \frac{1}{2}\rho||u||^2 - \langle u - u^k, v^k\rangle :\ u \in K\}.$$

  ∗ If $||u^{k+1} - u^k|| \le \epsilon$, then terminate: u^k is an ϵ - stationary point.

Otherwise, go to iteration k with $k := k + 1$.

5. Illustrative Example and Computational Results

To illustrate the algorithm we take an example in [8]. In this example the network has $4 + 2$ nodes and 10 arcs as shown in Fig.1, where the number attached to each arc is the arc capacity
$p = (p_1, ..., p_{10})^T = (8, 3, 1, 4, 2, 1, 7, 1, 2, 8)^T$.

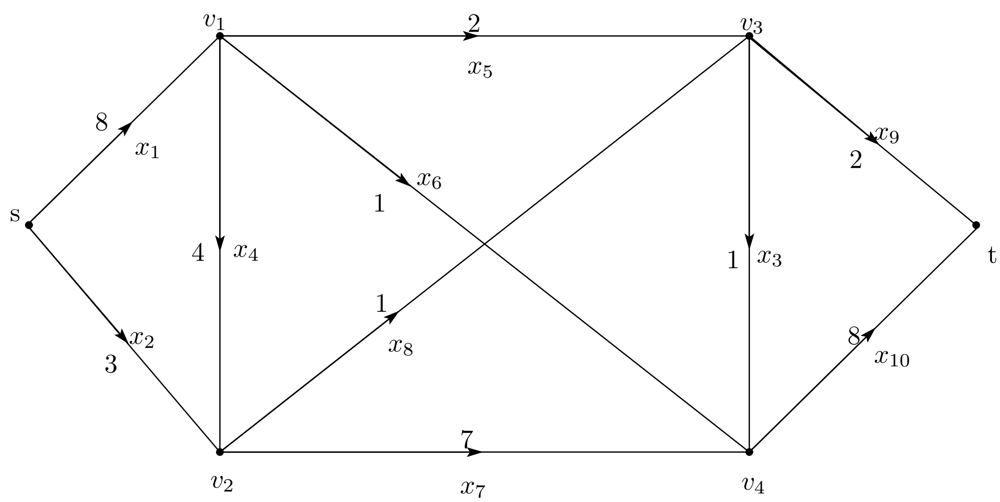

Fig.1. Network with 6 nodes and 10 arcs.

Initialization: Compute an extreme maximal flow $u^0 = (\lambda^0, x^0)$
where $\lambda^0 = (1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.400000,$
$1.000000, 1.000000, 90.600000)$;
$x^0 = (7.000000, 3.000000, 0.066667, 4.000000, 2.000000, 1.000000, 6.933333,$
$0.066667, 2.000000, 8.000000)$.

Iteration k:

At $k = 1$, the iterate is $u^1 = (\lambda^1, x^1)$, where
$\lambda^1 = (1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.720001,$
$1.000000, 1.000000, 90.279999)$,
$x^1 = (6.995152, 3.000000, 0.069899, 4.000000, 2.000000, 0.995152, 6.934949,$
$0.065051, 1.995152, 8.000000)$.

After 12 iterations, we obtain $u^* = (\lambda^*, x^*)$, where
$\lambda^* = (1.000000, 1.000000, 1.622026, 1.000000, 1.000000, 1.000000, 2.501089,$
$1.000000, 1.000000, 88.876885)$,
$x^* = (6.000000, 3.000000, 1.000000, 4.000000, 2.000000, 0.000000, 7.000000,$
$0.000000, 1.000000, 8.000000)$.

It is the same global optimal solution $x^* = (6, 3, 1, 4, 2, 0, 7, 0, 1, 8)$ that is obtained in [8]. In Fig.2 (x_i^*, p_i) is given on each arc.

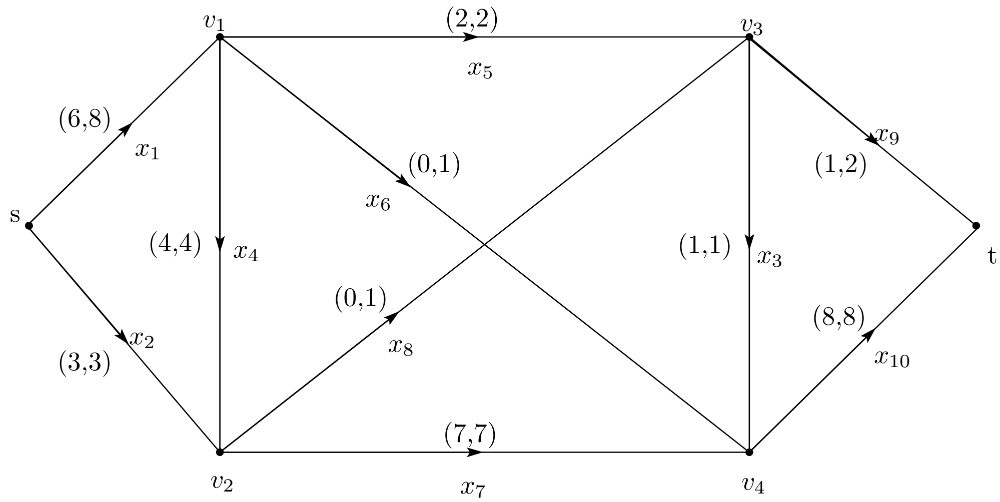

Fig.2. A minimum maximal flow (x_i^*, p_i).

To test the DCA algorithm, for each pair (m, n) we run it on 5 randomly different sets of data. These sets of data are chosen as in [16]. Test problems are executed on CPU, chip Intel Core(2) 2.53 GHz, RAM 2 GB, $C^{++}(VC^{++}2005)$ programming language. Numerical results are summarized in Table 1.

Table 1. Computational Results

m	n	Data	Op-value	Iter.	Value-γ_c	Time
6	10	$data_1$	7	7	3.02E-07	0.77
6	10	$data_2$	2	19	7.92E-07	0.86
6	10	$data_3$	5	12	1.95E-06	0.53
6	10	$data_4$	3	3	0.00E+00	0.19
6	10	$data_5$	8	4	0.00E+00	0.23
16	20	$data_6$	6	8	0.00E+00	0.55
16	20	$data_7$	4	16	4.46E-07	0.67
16	20	$data_8$	1	26	2.40E-07	1.05
16	20	$data_9$	0	21	4.41E-07	0.88
16	20	$data_{10}$	6	43	6.82E-07	1.76
30	70	$data_{11}$	3	116	1.06E-05	7.05
30	70	$data_{12}$	4	500	4.00E-05	30.58
30	70	$data_{13}$	7	165	8.00E-06	10.14
30	70	$data_{14}$	2	234	3.94E-06	14.31
30	70	$data_{15}$	6	242	1.13E-06	14.73
100	200	$data_{16}$	4	199	2.42E-07	41.73
100	200	$data_{17}$	1	191	1.52E-05	40.52
100	200	$data_{18}$	2	500	7.70E-06	105.74
100	200	$data_{19}$	4	390	2.37E-05	80.19
100	200	$data_{20}$	5	500	3.60E-05	103.09,

where

Data: Sets of data

Op-value: Optimal value

Iter.: Number of iterations

Value-γ_c: Value of $\gamma_c(.)$

Time: Average CPU-Time in seconds.

Remark: We can see from Table 1 that the algorithm terminates at an approximate Pareto-solution, since the value of γ at the iterate point is near to zero.

6. Conclusions

We have to use a regularization technique to formulate the problem of optimizing a real valued function over the Pareto efficient set of a multiple objective linear program as a smooth DC constrained optimization problem. We have applied the quadratic penalty function and the DCA algorithms for solving the resulting penalized DC problem. We also have applied the smooth DCA algorithm to a minmax flow problem. The obtained computational results have shown that the proposed smooth DC algorithm can solve efficiently the minmax flow problem.

References

1. H.A. Le Thi, T. Pham Dinh and L. D. Muu, Numerical solution for optimization over the efficient set by DC optimization algorithms, *Operations Research Letters* **19** (1996), 117 - 128.
2. H.A. Le Thi, T. Pham Dinh and L. D. Muu, Simplicaly-constrained DC optimization over the efficient and weakly sets. *J. Optimization Theory and Applications* **117** (2003), 503 - 531.
3. H.A. Le Thi and T. Pham Dinh, Convex analysis approach to DC programming: Theory, algorithms and applications, *Acta Math. Vietnam.* **22** (1997), 289 - 355.
4. H.A. Le Thi and T. Pham Dinh, Solving a class of linearly constrained indefinite quadratic problems by DC algorithms, *J. Global Optimization* **11** (1997), 253 - 285.
5. H.P. Benson, Optimization over the efficient set, *J. Math. Anal. Appl.* **98** (1984), 560 - 580.
6. H.P. Benson, An all-linear programming relaxation algorithm for optimization over the efficient set, *J. Global Optimization* **1** (1991), 83 - 104.
7. A. Fiacco and G.Mc. Cormick, *Nonliear Programming Squential Unconstrained Minimization Techniques*, John Willey & Sons, New York, 1968.
8. J. Gotoh, N.V. Thoai and Y. Yamamoto, Global optimization method for solving the minimum maximal flow problem, *Optimization. Methods Software* **18** (2003), 395 - 415.
9. N.T.B. Kim and L.D. Muu, On the projection of the efficient set and potential applications, *Optimization* **51** (2002). 401 - 421.
10. H. Mine and M. Fukushima, A minimization method for the sum of a convex function and a continuously differentiable function, *J. Optimization Theory and Applications* **33** (1981), 9 - 23.
11. J. Philip, Algorithms for the vector maximization problem, *Math. Program.* **2** (1972), 207 - 229.
12. N. V. Quy and L. D. Muu, On penalty function method for class of nonconvex constrained optimization problems, *Vietnam J. Math.* **29** (2001), 235 - 256.
13. M. Shigeno, I. Takahashi and Y. Yamamoto, Minimum maximal flow problem - an optimization over the efficient set, *J. Global Optimization* **25** (2003), 425 - 443.
14. Y. Yamamoto, Optimization over the efficient set: overview, *J. Global Optimization* **22** (2002), 285 - 317.
15. D. L. Zhu and P. Marcotte, An extended descent framework for variational inequalities, *J. Optimization Theory and Applications* **80** (1994), 349 - 366.
16. Daisuke Zenke *Algorithms for the Minimum Maximal Flow Problem - Optimization over the Efficient Set*, Graduate School of Systems and Information Engineering, University of Tsukuba March, 2006.