Minimizing a Linear Multiplicative-Type Function under Network Flow Constraints

Takahito Kuno*
Takahiro Utsunomiya

September 12, 1995

ISE-TR-95-124

Institute of Information Sciences and Electronics
University of Tsukuba
Tsukuba, Ibaraki 305, Japan
Phone: +81-298-53-5540, Fax: +81-298-53-5206, E-mail: takahito@is.tsukuba.ac.jp

* The author was partly supported by Grant-in-Aid for Scientific Research of the Ministry of Education, Science and Culture, Grant No. (C)07680447.

Minimizing a Linear Multiplicative-Type Function under Network Flow Constraints

Takahito Kuno* and Takahiro Utsunomiya
Institute of Information Sciences and Electronics
University of Tsukuba

September 1995

Abstract. In this paper, we consider a special class of nonconvex network flow problems, whose objective function is a product of two affine functions. This problem arises when one tries to send as much flow as possible at minimum possible cost in an ordinary two-terminal network. We will show that a primal-dual algorithm can generate a globally optimal solution in pseudo-polynomial time and a globally ϵ-optimal solution in polynomial time.

Key words: Global optimization, linear multiplicative programming, nonconvex cost network flow, bicriteria decision making, primal-dual algorithm.

1. Introduction

Suppose $G = (V, E)$ is a directed graph consisting of a set V of n nodes and a set E of m arcs. Each arc $(i, j) \in E$ has an associated unit cost c_{ij} and nonnegative capacity u_{ij}. Given two distinct nodes s and t in V, we wish to find a flow $\boldsymbol{x}$ in network $N = (G, s, t, \boldsymbol{c}, \boldsymbol{u})$ which maximizes the total amount v of flow from s to t and simultaneously minimizes the total cost $\boldsymbol{cx}$ subject to

$$\begin{aligned} &\sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} v & \text{for } i = s, \\ 0 & \text{for all } i \in V \setminus \{s, t\}, \\ -v & \text{for } i = t, \end{cases} \\ &0 \leq x_{ij} \leq u_{ij} \quad \text{for each } (i, j) \in E. \end{aligned} \tag{1.1}$$

When such two objectives without a common scale need optimizing simultaneously, a handy approach is to optimize their product [12]. In our problem, minimizing $f(\boldsymbol{x}, v) = (\boldsymbol{cx} + c_0) \cdot (V - v)$ will provide a satisfactory solution for us, where c_0 and V are nonnegative constants expressing a setup cost and an ideal value of v respectively. This approach, however, seems to have some difficulty, because the product of two affine functions can be a nonconvex function [11]. Hence our objective function f may have

*The author was partially supported by Grand-in-Aid for Scientific Research of the Ministry of Education, Science and Culture, Grant No. (C)07680447.

multiple local minima in the feasible set defined by (1.1), many of which fail to be globally optimal.

An alternative approach proposed in [10] is to minimize $\max\{\alpha_1(\boldsymbol{cx}+c_0),\ \alpha_2(V-v)\}$ for some positive weights α_1 and α_2. Also the max flow problem with a side constraint $\boldsymbol{cx} \leq D$ can be considered to have the same purpose as our problem (see e.g. [1]). These are linear programming problems and can be solved in polynomial time.

In this paper, we will show that a global minimum of $f(\boldsymbol{x},\ v) = (\boldsymbol{cx} + c_0) \cdot (V - v)$ under the constraints (1.1) can be found in time $O((m + n \log n)v_{\max})$, where $v_{\max}$ is the maximal value of v satisfying (1.1). Moreover, if we give up accuracy, a globally ϵ-optimal solution can be obtained in time $O(m^2(m + n \log n)/\epsilon)$, i.e., the proposed algorithm is a fully polynomial-time approximation scheme [5].

2. Reduction to Minimization of a Univariate Function

The problem stated above can be formulated as follows:

$$
\text{(P)}\quad
\begin{array}{|ll}
\text{minimize} & f(\boldsymbol{x},\ v) = \Big(\displaystyle\sum_{(i,j)\in E} c_{ij}x_{ij} + c_0\Big) \cdot (V - v) \\
\text{subject to} & \displaystyle\sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} v, & i = s, \\ 0, & i \in V \setminus \{s,\ t\}, \\ -v, & i = t, \end{cases} \\
& 0 \leq x_{ij} \leq u_{ij}, \quad (i,\ j) \in E,
\end{array}
$$

where c_{ij}'s and u_{ij}'s are nonnegative integers. It is reasonable to assume that $c_0 \geq 0$ and $V \geq v_{\max}$, where $v_{\max}$ is the maximal value of v satisfying all constraints of (P). Then both the values of $\sum_{(i,j)\in E} c_{ij}x_{ij} + c_0$ and $V - v$ are always nonnegative. However, in case either $c_0 = 0$ or $V = v_{\max}$, the minimal value of f is equal to zero and the problem becomes an ordinary network flow problem. To avoid such a trivial case, we assume that

$$
c_0 > 0,\quad V > v_{\max}. \tag{2.1}
$$

The objective function f is then the product of two positive affine functions and hence quasiconcave on the feasible set of (P) [11].

If we fixed the value of v in (P), then we have a minimum cost flow problem:

$$
\text{(P}(v)\text{)}\quad
\begin{array}{|ll}
\text{minimize} & \displaystyle\sum_{(i,j)\in E} c_{ij}x_{ij} \\
\text{subject to} & \displaystyle\sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} v, & i = s, \\ 0, & i \in V \setminus \{s,\ t\}, \\ -v, & i = t, \end{cases} \\
& 0 \leq x_{ij} \leq u_{ij}, \quad (i,\ j) \in E.
\end{array}
$$

As well known, we can solve (P(v)) in strongly polynomial time and obtain an optimal flow $\boldsymbol{x}^*(v)$ if $0 \leq v \leq v_{\max}$. Let

$$F(v) \equiv f(\boldsymbol{x}^*(v), v). \tag{2.2}$$

Then we can see that solving the original problem (P) amounts to locating a global minimizer of F in the interval $[0, v_{\max}]$.

Lemma 2.1. *If $v^* \in [0, v_{\max}]$ is a global minimizer of F, then $(\boldsymbol{x}^*(v^*), v^*)$ solves (P), where $\boldsymbol{x}^*(v^*)$ is an optimal flow of (P(v^*)).* □

Since capacity u_{ij} of each arc $(i, j) \in E$ is integral, (P(v)) has an integer optimal flow if v is an integer in $[0, v_{\max}]$ (see e.g., [1]). Moreover, since f is quasiconcave on the feasible set of (P), there is a global minimum point of f among integral $(\boldsymbol{x}^*(v), v)$'s. These facts imply that one can obtain a globally optimal solution of (P) by solving (P(v)) for all integers in $[0, v_{\max}]$. Although such a primitive algorithm can run in pseudo-polynomial time, i.e., in time $O(mM(m, n)U)$ where $M(m, n)$ is the running time of a minimum cost flow algorithm and $U = \max\{u_{ij} : (i,j) \in E\}$, it would be far from efficient in practice.

3. Pseudo-Polynomial Algorithm for Finding a Globally Optimal Solution

To improve the efficiency of the algorithm for solving (P), let us observe some characteristics of function F.

For $v \in [0, v_{\max}]$, let us denote by $g(v)$ the optimal value of (P(v)). Then we have

$$F(v) = (g(v) - c_0) \cdot (V - v). \tag{3.1}$$

The following proposition is well known in the literature about parametric linear programming (see e.g., [2]):

Proposition 3.1. *Function $g : [0, v_{\max}] \to \mathbb{R}$ is convex and piecewise affine.* □

Furthermore, since c_{ij}'s are assumed to be nonnegative, g must be nondecreasing. Let p be the number of affine pieces of g and let

$$g(v) = \alpha_k v - \beta_k, \quad v \in [v_{k-1}, v_k], \quad k = 1, \ldots, p, \tag{3.2}$$

where $v_0 = 0$, $v_p = v_{\max}$, and α_k's and β_k's are nonnegative constants. Then

$$F(v) = (\alpha_k v - \beta_k + c_0) \cdot (V - v), \quad v \in [v_{k-1}, v_k], \quad k = 1, \ldots, p. \tag{3.3}$$

We see that F is a concave quadratic function on each $[v_{k-1}, v_k]$ and hence achieves the minimum over the subinterval at either v_{k-1} or v_k. Among such end points v_k, $k = 0, 1, \ldots, p$, exists a global minimizer v^* of F over the whole interval $[0, v_{\max}]$. Hence, to solve (P), we need only to specify all affine pieces of g. In the rest of this section, we will show that one can successively generate all affine pieces of g in the course of solving a minimum cost flow problem ($P(v_{\max})$) using the primal-dual algorithm of Ford and Fulkerson [3].

Recall that the primal-dual algorithm in [3] builds up an optimal flow of ($P(v_{\max})$) step by step, by adding flows along augmenting paths with the least cost in an auxiliary network N'. At each stage, N' is constructed from $N = (G = (V, E), s, t, \mathbf{c}, \boldsymbol{u})$ and the present flow $\boldsymbol{x}'$ according to the rules below: For each $(i, j) \in E$,

Rule 1: if $x'_{ij} < u_{ij}$, then let $(i, j) \in E_1$, $u'_{ij} = u_{ij} - x'_{ij}$ and $c'_{ij} = c_{ij}$,

Rule 2: if $x'_{ij} > 0$, then let $(j, i) \in E_2$, $u'_{ij} = x'_{ij}$ and $c'_{ij} = -c_{ij}$.

The resulting graph $G' = (V, E_1 \cup E_2)$ together with capacity vector $\boldsymbol{u}'$ and cost vector $\mathbf{c}'$ consists the auxiliary network $N' = (G', s, t, \mathbf{c}', \boldsymbol{u}')$ with respect to flow $\boldsymbol{x}'$. Unless the present flow value v' reaches $v_{\max}$, we can find an augmenting path $\pi \subset E_1 \cup E_2$ with the least cost in N', by solving a shortest path problem from s to t in G' with arc length $\mathbf{c}'$. Let

$$\bar{\delta} = \min\{u'_{ij} : (i, j) \in \pi\}. \tag{3.4}$$

Then we have a well-known result.

Lemma 3.2. *Let $0 \le \delta \le \bar{\delta}$. Also for each $(i, j) \in E$, let*

$$x'_{ij}(\delta) = \begin{cases} x'_{ij} + \delta & \text{if } (i, j) \in \pi \cap E_1, \\ x'_{ij} - \delta & \text{if } (j, i) \in \pi \cap E_2, \\ x'_{ij} & \text{otherwise.} \end{cases} \tag{3.5}$$

Then $\boldsymbol{x}'(\delta)$ is an optimal solution of ($P(v' + \delta)$).

Proof: See e.g. [1, 3]. □

According to (3.5) we update flow $\boldsymbol{x}'$, and then proceed to the next stage. Here we should note that

$$\begin{aligned} g(v' + \delta) &= \sum_{(i,j)\in E} c_{ij} x'_{ij}(\delta) \\ &= \sum_{(i,j)\in E} c_{ij} x'_{ij} + \delta\Big(\sum_{(i,j)\in\pi\cap E_1} c_{ij} - \sum_{(j,i)\in\pi\cap E_2} c_{ij}\Big). \end{aligned} \tag{3.6}$$

This implies that g is an affine function on $[v', v' + \bar{\delta}]$. Hence all points in $(v', v' + \bar{\delta})$ can be discarded when we locate a global minimizer v^* of F in $[0, v_{\max}]$.

From the above observation, we can summarize an algorithm for solving (P).

Algorithm PD.

Step 0. Let $(\boldsymbol{x}', v') = (\mathbf{0}, 0)$, $(\boldsymbol{x}^*, v^*) = (\mathbf{0}, 0)$ and $F^* = c_0 V$.

Step 1. Construct the auxiliary network $N' = (G', s, t, \boldsymbol{c}', \boldsymbol{u}')$ with respect to $\boldsymbol{x}'$.

Step 2. If there is no path from s to t in G', then terminate. Otherwise, compute a shortest path π and let $\bar{\delta} = \min\{u'_{ij} : (i, j) \in \pi\}$. According to (3.5), compute $\boldsymbol{x}'(\bar{\delta})$ and let $(\boldsymbol{x}', v') = (\boldsymbol{x}'(\bar{\delta}), v' + \bar{\delta})$.

Step 3. If $(\sum_{(i,j)\in E} c_{ij}x'_{ij} + c_0) \cdot (V - v') < F^*$, then revise the incumbent:

$$(\boldsymbol{x}^*, v^*) = (\boldsymbol{x}', v'), \quad F^* = (\sum_{(i,j)\in E} c_{ij}x'_{ij} + c_0) \cdot (V - v').$$

Step 4. Return to Step 1. □

Theorem 3.3. *Algorithm PD yields a globally optimal solution $(\boldsymbol{x}^*, v^*)$ of (P) in $O(m(m + n\log n)U)$ arithmetic operations, where $U = \max\{u_{ij} : (i, j) \in E\}$.*

Proof: The main parts of this algorithm are the construction of N' in Step 1 and the computation of π and $\boldsymbol{x}'(\bar{\delta})$ in Step 2. It is obvious that both the construction of N' and the computation of $\boldsymbol{x}'(\bar{\delta})$ can be done in time $O(m)$. On computing π, we can transform $\boldsymbol{c}'$ into a nonnegative vector by introducing node potentials, because all c_{ij}'s are nonnegative. (see e.g. [1] for details). We can therefore obtain π using Dijkstra's algorithm in time $O(m + n\log n)$ [4]. Since $\bar{\delta} \geq 1$ on the assumption that all u_{ij}'s are integral, Steps 1 and 2 are repeated at most $v_{\max}$ times. Hence the total number of arithmetic operations is bounded by $O((m + n\log n)v_{\max}) \leq O(m(m + n\log n)U)$. □

Note that if Algorithm PD lacks Step 3, it is nothing but the primal-dual algorithm of Ford and Fulkerson. Although the worst-case time complexity of the algorithm is not polynomial in the input length, its practical efficiency is guaranteed by many experiments performed so far.

4. Polynomial Algorithm for Finding a Globally ϵ-Optimal Solution

Since the worst-case number of affine pieces of g is exponential in the input length [16], it would be hard to design polynomial-time algorithms for finding a globally optimal solution of (P). However, if we give up accuracy, it is possible to find a globally ϵ-optimal solution in polynomial time. In the sequel, we impose the following assumption on the ideal value V of v:

$$V - v_{\max} \geq U \equiv \max\{u_{ij} : (i, j) \in E\}. \tag{4.1}$$

Given a tolerance $\epsilon \in (0, 1]$, we say that a feasible solution $(\boldsymbol{x}^\epsilon, v^\epsilon)$ of (P) is *globally ϵ-optimal* if it satisfies

$$0 \leq \frac{f(\boldsymbol{x}^\epsilon, v^\epsilon) - f(\boldsymbol{x}^*, v^*)}{f(\boldsymbol{x}^*, v^*)} \leq \epsilon, \tag{4.2}$$

where $(\boldsymbol{x}^*, v^*)$ is a globally optimal solution of (P). To obtain such an approximate solution, we consider a problem with truncated arc capacities:

$$(\overline{\mathrm{P}}) \quad \left| \begin{array}{ll} \text{minimize} & f(\boldsymbol{x}, v) = (\sum_{(i,j)\in E} c_{ij}x_{ij} + c_0)\cdot(V - v) \\ \text{subject to} & \sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} v, & i = s, \\ 0, & i \in V \setminus \{s, t\}, \\ -v, & i = t, \end{cases} \\ & 0 \leq x_{ij} \leq \bar{u}_{ij}, \quad (i, j) \in E, \end{array} \right.$$

where

$$\bar{u}_{ij} = M\lfloor u_{ij}/M \rfloor, \quad (i, j) \in E, \tag{4.3}$$

for some positive constant M, and the other notations are the same as (P). We can of course apply Algorithm PD to $(\overline{\mathrm{P}})$. Then an optimal solution $(\bar{\boldsymbol{x}}, \bar{v})$ will be obtained in time $O(m(m + n\log n)U/M)$ because the flow augmentation $\bar{\delta}$ in Step 2 cannot be less than M. We also note that $(\bar{\boldsymbol{x}}, \bar{v})$ is feasible to the original problem (P) and satisfies

$$f(\boldsymbol{x}^*, v^*) \leq f(\bar{\boldsymbol{x}}, \bar{v}). \tag{4.4}$$

On the other hand, we can obtain an feasible solution of $(\overline{\mathrm{P}})$ by rounding $(\boldsymbol{x}^*, v^*)$.

Let $\tilde{E}$ be the set of arcs (i, j)'s such that $x^*_{ij} > 0$. Then we can decompose flow $\boldsymbol{x}^*$ into a number of flows along directed paths from s to t in $\tilde{G} = (V, \tilde{E})$, using the following procedure:

0° Let $E' = \tilde{E}$, $x'_{ij} = x^*_{ij}$ for each $(i, j) \in E'$ and $k = 1$.

1° If there is no path from s to t in (V, E'), then terminate. Otherwise, compute a path $\pi_k \subset E'$ with the least number of arcs, and let $\delta_k = \min\{x'_{ij} : (i, j) \in \pi_k\}$.

2° For each $(i, j) \in \pi_k$, let $x'_{ij} = x'_{ij} - \delta_k$. Also let $E' = E' \setminus \{(i, j) \in \pi_k : x'_{ij} = 0\}$.

3° Let $k = k + 1$ and go to 1°. □

Since at least one arc is deleted from E' every iteration, this procedure terminates and yields at most $|\tilde{E}|$ directed paths π_k's such that

$$x^*_{ij} = \sum_{\{k:\pi_k\in\Pi_{ij}\}} \delta_k \text{ for each } (i, j) \in \tilde{E}; \;\; v^* = \sum_{k=1}^{q} \delta_k, \tag{4.5}$$

where

$$\Pi_{ij} = \{\pi_k : (i,\, j) \in \pi_k,\ k = 1,\, \ldots,\, q\} \tag{4.6}$$

for each $(i,\, j) \in \tilde{E}$, and $q \leq |\tilde{E}|$ is the total number of π_k's. For each k, let

$$\tilde{\delta}_k = \max\{0,\, \delta_k - M\}. \tag{4.7}$$

Proposition 4.1. *For each $(i,\, j) \in E$, let*

$$\tilde{x}_{ij} = \begin{cases} x^*_{ij} = \sum_{\{k:\pi_k \in \Pi_{ij}\}} \tilde{\delta}_k & \text{if } (i,\, j) \in \tilde{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

Also let

$$\tilde{v} = \sum_{k=1}^{q} \tilde{\delta}_k. \tag{4.9}$$

Then $(\tilde{\boldsymbol{x}},\, \tilde{v})$ is a feasible solution of $(\overline{P})$.

Proof: We obviously have $\sum_{\{j:(s,j)\in E\}} \tilde{x}_{sj} = \tilde{v}$ and $\sum_{\{j:(j,t)\in E\}} \tilde{x}_{jt} = -\tilde{v}$. At other nodes $i \in V \setminus \{s,\, t\}$, flow $\tilde{\boldsymbol{x}}$ is conserved because it consists of flows along paths π_k's from s to t. If $(i,\, j) \in E$ belongs to some path π_k with $\tilde{\delta}_k > 0$, then $0 < \tilde{x}_{ij} \leq x^*_{ij} - M \leq \bar{u}_{ij}$. Otherwise, the value of $\tilde{x}_{ij}$ vanishes. Hence, $\tilde{\boldsymbol{x}}$ satisfies all capacity constraints as well. □

Thus $(\boldsymbol{x}^*,\, v^*)$ provides an feasible solution $(\tilde{\boldsymbol{x}},\, \tilde{v})$ of $(\overline{\text{P}})$ satisfying

$$f(\bar{\boldsymbol{x}},\, \bar{v}) \leq f(\tilde{\boldsymbol{x}},\, \tilde{v}). \tag{4.10}$$

We also have

$$\sum_{(i,j)\in E} c_{ij}\tilde{x}_{ij} \leq \sum_{(i,j)\in E} c_{ij}x^*_{ij}, \tag{4.11}$$

since all c_{ij}'s are nonnegative. This together with $\tilde{v} \geq v^* - qM$ implies

$$\begin{aligned} f(\tilde{\boldsymbol{x}},\, \tilde{v}) &\leq \Big(\sum_{(i,j)\in E} c_{ij}x^*_{ij} + c_0\Big) \cdot (V - v^* + qM) \\ &= f(\boldsymbol{x}^*,\, v^*) + \Big(\sum_{(i,j)\in E} c_{ij}x^*_{ij} + c_0\Big)mM. \end{aligned} \tag{4.12}$$

Hence, from (4.10) and (4.12) we have

$$\frac{f(\bar{\boldsymbol{x}},\, \bar{v}) - f(\boldsymbol{x}^*,\, v^*)}{f(\boldsymbol{x}^*,\, v^*)} \leq \frac{mM}{V - v^*} \leq \frac{mM}{U}, \tag{4.13}$$

by noting $V - v^* \geq U$ on assumption (4.1). Therefore an optimal solution $(\bar{\boldsymbol{x}},\, \bar{v})$ of $(\overline{\text{P}})$ can satisfy the ϵ-optimality condition (4.2) if we let

$$M = \epsilon U/m. \tag{4.14}$$

In this case, the time complexity of Algorithm PD is bounded by $O(m^2(m + n\log n)/\epsilon)$.

5. Conclusion

We showed in this paper that a parametric approach provides an efficient algorithm for solving a class of nonconvex cost network flow problems. The algorithm we proposed to solve (P) can generate a globally optimal solution in pseudo-polynomial time and a globally ϵ-optimal solution in polynomial time.

Minimization of a product of two affine functions like (P) is in general called linear multiplicative programming and has real world applications in abundance [12]. Also, among many global optimization problems [9], the linear multiplicative programming problem is one of a few problems which can be solved in a practical sense. In fact, the computational results reported in [11, 13] show that parametric simplex algorithms solve a general linear multiplicative programming problem in just a little more computational time than needed for solving a linear programming problem of the same size. However, it is still an open question whether a linear multiplicative programming problem is polynomially solvable or not. Besides linear multiplicative programming problems, parametric approaches are very effective for solution of certain concave cost network flow problems [8, 14, 15, 17, 18]. The readers are also referred to [6, 7] for the current state-of-the-art of nonconvex network optimization.

References

[1] Ahuja, R.K., T.L. Magnanti and J.B. Orlin, *Network flows: Theory, Algorithms and Applications,* Prentice Hall (N.J., 1993).

[2] Chvătal, V., *Linear Programming,* Freeman and Company (N.Y., 1971).

[3] Ford, L.R. and D.R. Fulkerson, *Flows in Networks,* Princeton University Press (N.J., 1962).

[4] Fredman, M.L. and R.E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of ACM* **34** (1988), 596 – 615.

[5] Garey, M.S. and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W.H. Freeman (N.Y., 1979).

[6] Guisewite, G.M., "Network problems," in R. Horst and P.M. Pardalos (eds.), *Handbook of Global Optimization,* Kluwer Academic Publishers (Dortrecht, 1995).

[7] Guisewite, G.M. and P.M. Pardalos, "Minimum concave-cost network flow problems: applications, complexity and algorithms," *Annals of Operations Research* **25** (1990), 75 – 100.

[8] Guisewite, G.M. and P.M. Pardalos, "A polynomial time solvable concave network flow problem," *Networks* **23** (1993), 143 – 149.

[9] Horst, R. and H. Tuy, *Global Optimization: Deterministic Approaches,* Springer-Verlag (Berlin, 1990).

[10] Katoh, N., "An efficient algorithm for bicriteria minimum-cost circulation problem," *Journal of the Operations Research Society of Japan* **32** (1989), 420 – 440.

[11] Konno, H. and T. Kuno, "Linear multiplicative programming," *Mathematical Programming* **56** (1992), 51 – 64.

[12] Konno, H. and T. Kuno, "Multiplicative programming problems," in R. Horst and P.M. Pardalos (eds.), *Handbook of Global Optimization,* Kluwer Academic Publishers (Dortrecht, 1995).

[13] Konno, H., Y. Yajima and T. Matsui, "Parametric simplex algorithms for solving a special class of nonconvex minimization problems," *Journal of Global Optimization* **1** (1991), 65 – 82.

[14] Kuno, T. and T. Utsunomiya, "A decomposition algorithm for solving certain classes of production-transportation problems with concave production cost," Technical Report ISE-TR-94-113, Institute of Information Sciences and Electronics, University of Tsukuba (Tsukuba, 1994) to appear in *Journal of Global optimization.*

[15] Kuno, T. and T. Utsunomiya, "A primal-dual algorithm for globally solving a production-transportation problem with concave production cost," Technical Report ISE-TR-95-123, Institute of Information Sciences and Electronics, University of Tsukuba (Tsukuba, 1995).

[16] Ruhe, G., "Complexity results for multicriterial and parametric network flows using pathological graph of Zadeh," *Zeitschrift für Operations Research* **32** (1988), 9 – 27.

[17] Tuy, H., N.D. Dan and S. Ghannadan, "Strongly polynomial time algorithms for certain concave minimization problems on networks," *Operations Research Letters* **14** (1993), 99 – 109.

[18] Tuy, H., S. Ghannadan, A. Migdalas and P. Värbrand, "Strongly polynomial algorithm for a production-transportation problem with concave production cost," *Optimization* **27** (1993), 205 – 227.