A Parametric Approach for Maximum Flow Problems with an Additional Reverse Convex Constraint

Takahito Kuno*

November 28, 1995

ISE-TR-95-128

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.

A Parametric Approach for Maximum Flow Problems with an Additional Reverse Convex Constraint

Takahito Kuno*

Institute of Information Sciences and Electronics

University of Tsukuba

November 1995

Abstract. In this paper, we consider maximum integral flow problems with an additional reverse convex constraint involving one or two nonlinear variables. Based on a parametric approach, we propose a polynomial-time algorithm for computing an integral flow globally optimal to the problem with a single nonlinear variable. We extend this idea and solve the problem with two nonlinear variables. The algorithm solves a sequence of ordinary minimum cost flow problems by using a conventional method and yields a globally optimal solution in pseudo-polynomial time.

Key words: Global optimization, reverse convex constraint, maximum flow problem, minimum cost flow problem, parametric approach.

1. Introduction

In this paper, we consider a certain class of maximum flow problems with an additional reverse convex constraint.

Let G be a directed graph consisting of a set V of n nodes and a set E of m arcs. We suppose that each node in a subset F of V is a factory producing a common commodity and supplies it through G to a particular node, say $n \in V \setminus F$. Our purpose is to transport the commodity to the demand node n as many units as possible subject to a restriction that the total cost never exceeds an allotted budget. The total cost consists of two parts: *production costs* and *transportation costs.* While transportation costs can often be assumed to be linear, production costs generally exhibit economy of scale and are described by concave and nondecreasing functions of the production quantities. Therefore the problem has a reverse convex constraint expressing the budgetary limitation in its formula as well as the usual arc capacity and flow conservation constraints. Such a reverse convex program is a typical global optimization problem and has multiple locally optimal solutions, many of which fail to be globally optimal [9].

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

The problem is closely related to the production-transportation problem studied by some authors in [12, 13, 15, 16, 18]. The latter can be regarded as a dual problem of the former in the sense that the total cost is minimized subject to a fixed flow value, instead of restricted by a budget. Another related problem is the maximum flow problem with an additional linear constraint (see e.g. [1]), which ignores the production cost and hence is formulated as a linear program.

In this class of problems, we are concerned with the cases where $|F| \le 2$, i.e., the number of factories is one or two. We will develop an efficient algorithm based on a parametric approach in order to compute an integral flow globally optimal to either case. In Section 2, we will consider the case $|F| = 1$ and present a polynomial-time algorithm taking the flow value as a parameter. We will extend this algorithm and solve the problem with $|F| = 2$ in Section 3. The proposed algorithm solves a sequence of minimum cost flow problems by using the augmenting-path algorithm of Ford and Fulkerson [3] and yields a globally optimal solution in pseudo-polynomial time.

2. The Problem with Single Factory

We first consider the case where F is a singleton, i.e. factory $1 \in V$ produces y units of the commodity at a cost of $f(y)$ and supplies the demand node $n \in V$ through $G = (V, E)$. The production function $f : \mathbf{R}_+ \to \mathbf{R}$ is concave and nondecreasing, where $\mathbf{R}_+$ represents the set of nonnegative real numbers. The cost $c_{ij} \ge 0$ and capacity $u_{ij} \ge 0$ associated with each arc $(i, j) \in E$ are real and integral valued, respectively. When the budgetary limit is given by some real $b \ge 0$, the problem is formulated as follows:

$$
(\mathrm{P}_1)\quad
\left|
\begin{array}{ll}
\text{maximize} & y \\
\text{subject to} & \displaystyle\sum_{j|(i,j)\in E} x_{ij} - \sum_{j|(j,i)\in E} x_{ji} = \begin{cases} y & \text{for } i = 1, \\ 0 & \text{for all } i \in V \setminus \{1, n\}, \\ -y & \text{for } i = n, \end{cases} \\
& 0 \le x_{ij} \le u_{ij} \quad \text{for } (i, j) \in E, \quad y \in \mathbf{Z}_+, \\
& f(y) + \displaystyle\sum_{(i,j)\in E} c_{ij} x_{ij} \le b,
\end{array}
\right.
$$

where $\mathbf{Z}_+$ is the set of nonnegative integers. Due to the last constraint, (P_1) is neither linear nor convex programming but belongs to global optimization even though the constraint $y \in \mathbf{Z}_+$ is relaxed into $y \in \mathbf{R}_+$. Let

$$
D(y) = \left\{ x \in \mathbf{R}^m \;\middle|\; \begin{array}{l} \displaystyle\sum_{j|(i,j)\in E} x_{ij} - \sum_{j|(j,i)\in E} x_{ji} = \begin{cases} y, & i = 1, \\ 0, & i \in V \setminus \{1, n\}, \\ -y, & i = n, \end{cases} \\ 0 \le x_{ij} \le u_{ij}, \quad (i, j) \in E \end{array} \right\}, \tag{2.1}
$$

$$
D = \{(x, y) \in \mathbf{R}^m \times \mathbf{R} \mid x \in D(y), \; y \ge 0\}, \tag{2.2}
$$

$$C = \{(\boldsymbol{x}, y) \in \mathbf{R}^m \times \mathbf{R} \mid f(y) + \sum_{(i,j)\in E} c_{ij}x_{ij} > b\}, \tag{2.3}$$

where $m = |E|$ and vector $\boldsymbol{x}$ consists of x_{ij}, $(i, j) \in E$. Then the relaxed feasible region is expressed by the difference $D \setminus C$ of two convex sets D and C. This implies that (P_1) is a kind of reverse convex program (see [9]).

Let $\mathbf{c}$ and $\boldsymbol{u}$ denote the vectors consisting c_{ij} and u_{ij}, $(i, j) \in E$, respectively. To solve (P_1), we consider a minimum cost flow problem defined below in network $(G, 1, n, \mathbf{c}, \boldsymbol{u})$:

$$(\mathrm{P}_1(y)) \quad \left| \begin{array}{ll} \text{minimize} & \sum_{(i,j)\in E} c_{ij}x_{ij} \\ \text{subject to} & \boldsymbol{x} \in D(y), \end{array} \right.$$

where y is a constant. Problem (P_1) can be solved if for every integer $y \in [0, y_{\max}]$ we check whether the optimal value of $(\mathrm{P}_1(y))$ is greater than b or not, where $y_{\max}$ is the optimal value of a maximum flow problem associated with (P_1):

$$\left| \begin{array}{ll} \text{maximize} & y \\ \text{subject to} & (\boldsymbol{x}, y) \in D. \end{array} \right. \tag{2.4}$$

Let $\boldsymbol{x}^*(y)$ be an optimal solution of $(\mathrm{P}_1(y))$ and let

$$g(y) = \sum_{(i,j)\in E} c_{ij}x^*_{ij}(y). \tag{2.5}$$

Also let

$$h(y) = f(y) + g(y). \tag{2.6}$$

Then $h(y)$ represents the minimum of $g(y) + \sum_{(i,j)\in E} c_{ij}x_{ij}$ when the value of y is fixed.

Lemma 2.1. *Let*

$$y^* = \max\{y \mid h(y) \le b, \;\; y \in [0, y_{\max}] \cap \mathbf{Z}\}. \tag{2.7}$$

Then any optimal solution $(\boldsymbol{x}^*(y^*), y^*)$ *of* $(P_1(y^*))$ *is a globally optimal solution of* (P_1).

Proof: Obvious by definition. □

In this way, (P_1) is reduced to the problem (2.7) with only one variable y. Although the constraint function h is neither convex nor concave over the interval $[0, y_{\max}]$, it possesses some favorable properties.

Lemma 2.2. *Function* $h : [0, y_{\max}] \to \mathbf{R}$ *is continuous, nondecreasing and piecewise concave.*

Proof: It is well known (see e.g. [2, 4]) that the optimal value $g(y)$ of $(\mathrm{P}_1(y))$, a parametric right-hand-side linear program, depends convexly and piecewise affinely on $y \in [0, y_{\max}]$. Also g is nondecreasing on the assumption that $c_{ij} \geq 0$ for each $(i, j) \in E$. Hence h is continuous, nondecreasing on $[0, y_{\max}]$ and also concave on each affine piece of g, since h is the sum of g and the concave nondecreasing function f. □

Lemma 2.2 guarantees that $h(y) > b$ for all $y \in [y', y_{\max}]$ if $h(y') > b$. Exploiting this monotonic property, we can find an integer y^* satisfying (2.7) by applying a binary search procedure to the interval $[0, y_{\max}]$.

Algorithm A.

Step 0. Compute $y_{\max}$ by solving problem (2.4). Let $\ell = 0$ and $r = y_{\max}$.

Step 1. Let $y' = \lfloor(\ell + r)/2\rfloor$. Compute an optimal solution $x^*(y')$ and the optimal value $g(y')$ of $(\mathrm{P}_1(y'))$ and let $h(y') = f(y') + g(y')$.

Step 2. If $y' = \ell$, then terminate. (If $h(y') > 0$, then (P_1) is infeasible. Otherwise, $(x^*(y'), y')$ is optimal to (P_1).)

Step 3. If $h(y') > 0$, then $r = y'$. Otherwise, let $\ell = y' + 1$. Return to Step 1. □

Theorem 2.3. *Algorithm A solves (P_1) in $O(M(m, n)\log mU)$ arithmetic operations and $O(\log mU)$ evaluations of f, where $M(m, n)$ is the running time of a minimum cost flow algorithm and $U = \max\{u_{ij} \mid (i, j) \in E\}$.*

Proof: Step 0 solves a maximum flow problem (2.4) and hence requires less than $M(m, n)$ arithmetic operations. The binary search for y^* is carried out in Steps 1 – 3. After $\log y_{\max}$ iterations, it yields the most right integer $y' \in [0, y_{\max}]$ among those satisfying $h(y') \leq b$ if such a y' exists. In each iteration, Step 1 evaluates f at y' and Step 2 solves a minimum cost flow problem $(\mathrm{P}_1(y'))$. Therefore, the total number of arithmetic operations is $O(M(m, n)\log y_{\max})$ and that of evaluations of f is $O(\log y_{\max})$. □

Note that $M(m, n)$ is strongly polynomial, e.g. $O(nm^2 \log^2 n)$ [5]. In Algorithm A, we have only to use the monotonicity but not the concavity of f. As a result, the running time can be bounded by a polynomial function of the problem input length, which will be almost the same complexity needed to solve a maximum flow problem with an additional linear constraint if the value of f is provided by an oracle. In other words, (P_1) is not essentially a class of global optimization problems, though it is in appearance. In the next section, however, we will show that the concavity of the production function is substantial in the problem with two factories.

3. The Problem with Two Factories

The second case supposes $F = \{1, 2\}$. Factories 1 and 2 supply y_1 and y_2 units, respectively, to terminal $n \in V$ through $G = (V, E)$. The cost of producing $\boldsymbol{y} = (y_1, y_2)$ is given as $f(\boldsymbol{y})$ by a function $f : \mathbf{R}^2_+ \to \mathbf{R}$, which is concave and coordinatewise nondecreasing, i.e.

$$f(\boldsymbol{y}') \leq f(\boldsymbol{y}'') \text{ if } \mathbf{0} \leq \boldsymbol{y}' \leq \boldsymbol{y}''. \tag{3.1}$$

It is often assumed that f is separable, i.e. $f(\boldsymbol{y}) = f_1(y_1) + f_2(y_2)$ for some concave nondecreasing functions $f_i : \mathbf{R}_+ \to \mathbf{R}$, $i = 1, 2$. However, we need not impose such an assumption upon our algorithm. As before, the cost c_{ij} and capacity u_{ij} of each arc $(i, j) \in E$ are nonnegative real and integral valued, respectively. Then the problem with a budgetary limit b, which is a nonnegative real, is as follows:

$$(\mathrm{P}_2) \quad \left| \begin{array}{ll} \text{maximize} & y_1 + y_2 \\ \text{subject to} & \displaystyle\sum_{j|(i,j)\in E} x_{ij} - \sum_{j|(j,i)\in E} x_{ji} = \begin{cases} y_1 & \text{for } i = 1, \\ y_2 & \text{for } i = 2, \\ 0 & \text{for all } i \in V \setminus \{1, 2, n\}, \\ -y_1 - y_2 & \text{for } i = n, \end{cases} \\ & 0 \leq x_{ij} \leq u_{ij} \;\text{ for } (i, j) \in E, \;\; \boldsymbol{y} \in \mathbf{Z}^2_+, \\ & f(\boldsymbol{y}) + \displaystyle\sum_{(i,j)\in E} c_{ij}x_{ij} \leq b. \end{array} \right.$$

Let

$$D(\boldsymbol{y}) = \left\{ \boldsymbol{x} \in \mathbf{R}^m \;\middle|\; \begin{array}{l} \displaystyle\sum_{j|(i,j)\in E} x_{ij} - \sum_{j|(j,i)\in E} x_{ji} = \begin{cases} y_1, & i = 1, \\ y_2, & i = 2, \\ 0, & i \in V \setminus \{1, 2, n\}, \\ -y_1 - y_2, & i = n, \end{cases} \\ 0 \leq x_{ij} \leq u_{ij}, \;\; (i, j) \in E \end{array} \right\}, \tag{3.2}$$

$$D = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbf{R}^m \times \mathbf{R}^2 \mid \boldsymbol{x} \in D(\boldsymbol{y}), \;\; \boldsymbol{y} \geq \mathbf{0}\}, \tag{3.3}$$

$$C = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbf{R}^m \times \mathbf{R}^2 \mid f(\boldsymbol{y}) + \sum_{(i,j)\in E} c_{ij}x_{ij} > b\}. \tag{3.4}$$

If the constraint $\boldsymbol{y} \in \mathbf{Z}^2_+$ is relaxed to $\boldsymbol{y} \geq \mathbf{0}$, the feasible region is again expressed by the difference $D \setminus C$ of two convex sets D and C.

Let us consider a minimum cost flow problem in network $(G, F, n, \boldsymbol{c}, \boldsymbol{u})$:

$$(\mathrm{P}_2(\boldsymbol{y})) \quad \left| \begin{array}{ll} \text{minimize} & \displaystyle\sum_{(i,j)\in E} c_{ij}x_{ij} \\ \text{subject to} & \boldsymbol{x} \in D(\boldsymbol{y}), \end{array} \right.$$

where $\boldsymbol{y}$ is a constant vector. We denote by $\boldsymbol{x}^*(\boldsymbol{y})$ an optimal solution of $(\mathrm{P}_2(\boldsymbol{y}))$ and by $g(\boldsymbol{y})$ the optimal value if $D(\boldsymbol{y}) \neq \emptyset$. If we try to extend Algorithm A to (P_2), we have to check the value $g(\boldsymbol{y})$ for each integral point $\boldsymbol{y}$ in the projection of D onto the $\boldsymbol{y}$-space, denoted by

$$\Omega = \{\boldsymbol{y} \in \mathbf{R}^2 \mid D(\boldsymbol{y}) \neq \emptyset, \ \ \boldsymbol{y} \geq \mathbf{0}\}. \tag{3.5}$$

To make it systematically, let us observe the minimum value of $f(\boldsymbol{y}) + \sum_{(i,j)\in E} c_{ij}x_{ij}$ when $\boldsymbol{y}$ is fixed, i.e. the value given by

$$h(\boldsymbol{y}) = f(\boldsymbol{y}) + g(\boldsymbol{y}). \tag{3.6}$$

Lemma 3.1. *Let*

$$\boldsymbol{y}^* \in \operatorname{argmax}\{y_1 + y_2 \mid h(\boldsymbol{y}) \leq b, \ \ \boldsymbol{y} \in \Omega \cap \mathbf{Z}^2\}. \tag{3.7}$$

Then any optimal solution $(\boldsymbol{x}^*(\boldsymbol{y}^*), \boldsymbol{y}^*)$ of $(P_2(\boldsymbol{y}^*))$ is a globally optimal solution of (P_2).

Proof: Obvious. □

Lemma 3.2. *If $h(\boldsymbol{y}') > b$ for some $\boldsymbol{y}' \in \Omega$, then*

$$h(\boldsymbol{y}) > b, \ \ \forall \boldsymbol{y} \in \{\boldsymbol{y} \in \Omega \mid \boldsymbol{y} \geq \boldsymbol{y}'\}. \tag{3.8}$$

Proof: Since $c_{ij} \geq 0$ for each $(i, j) \in E$, the optimal value $g(\boldsymbol{y})$ of $(\mathrm{P}_2(\boldsymbol{y}))$ does not fall below $g(\boldsymbol{y}')$ when $\boldsymbol{y} \geq \boldsymbol{y}'$. Hence (3.8) follows from (3.1) and (3.6). □

Corollary 3.3. *If $\min\{h(\boldsymbol{y}) \mid y_1 + y_2 = v, \ \ \boldsymbol{y} \in \Omega\} > b$ for a number v, then*

$$h(\boldsymbol{y}) > b, \ \ \forall \boldsymbol{y} \in \{\boldsymbol{y} \in \Omega \mid y_1 + y_2 \geq v\}. \tag{3.9}$$

Proof: For an arbitrary $\boldsymbol{y}' \in \{\boldsymbol{y} \in \Omega \mid y_1 + y_2 = v\}$,

$$h(\boldsymbol{y}') \geq \min\{h(\boldsymbol{y}) \mid y_1 + y_2 = v, \ \ \boldsymbol{y} \in \Omega\} > b.$$

Hence the assertion follows Lemma 3.2. □

3.1. Parametrization of problem (P_2)

Lemma 3.1 and Corollary 3.3 leads us to an outline of solution to (P_2):

Let us install an artificial node s in G and denote by $\bar{G} = (\bar{V}, \bar{E})$ the directed graph with $\bar{V} = V \cup \{s\}$ and $\bar{E} = E \cup \{(s, 1), (s, 2)\}$. Also let $u_{si} = +\infty$ for $i = 1, 2$. Then the maximum value $v_{\max}$ of $y_1 + y_2$ such that $\boldsymbol{y} \in \Omega$ is given by the optimal value of a maximum flow problem in network $(\bar{G}, s, n, \boldsymbol{u})$:

$$\left|\begin{array}{ll} \text{maximize} & v \\ \text{subject to} & (\boldsymbol{x}, \boldsymbol{y}) \in D, \\ & y_1 + y_2 = v, \ \ \boldsymbol{y} \geq \mathbf{0}. \end{array}\right. \tag{3.10}$$

For some $v \in [0, v_{\max}]$, if we can see that

$$(\mathrm{T}(v)) \ : \min\{h(\boldsymbol{y}) \mid y_1 + y_2 = v, \ \ \boldsymbol{y} \in \Omega\} > b$$

holds, we need not search $\Omega \cap \{\boldsymbol{y} \in \mathbf{R}^2 \mid y_1 + y_2 \geq v\}$ for $\boldsymbol{y}^*$ satisfying (3.7) any longer. Hence a binary search procedure will locate v^* such that $y_1^* + y_2^* = v^*$ in the interval $[0, v_{\max}]$ if only we can check $(\mathrm{T}(v))$ for each integer $v \in [0, v_{\max}]$. Note that any $\boldsymbol{y} \in \mathrm{argmin}\{h(\boldsymbol{y}) \mid y_1 + y_2 = v^*, \ \ \boldsymbol{y} \in \Omega \cap \mathbf{Z}^2\}$ provides a globally optimal solution $(\boldsymbol{x}^*(\boldsymbol{y}), \boldsymbol{y})$ of (P_2).

Suppose an arbitrary integer $v \in [0, v_{\max}]$ is given. Let $c_{si} = 0$ for $i = 1, 2$. To check if $(\mathrm{T}(v))$ is true or false, we first solve a minimum cost flow problem in network $(\bar{G}, s, n, \boldsymbol{c}, \boldsymbol{u})$:

$$(\mathrm{Q}(v)) \left| \begin{array}{ll} \text{minimize} & \sum_{(i,j) \in E} c_{ij} x_{ij} \\ \text{subject to} & (\boldsymbol{x}, \boldsymbol{y}) \in D, \\ & y_1 + y_2 = v, \quad \boldsymbol{y} \geq \mathbf{0}. \end{array} \right.$$

Let $(\bar{\boldsymbol{x}}(v), \bar{\boldsymbol{y}}(v))$ be an optimal solution of $(\mathrm{Q}(v))$. Then $\bar{\boldsymbol{x}}(v)$ is obviously an optimal solution of $(\mathrm{P}_2(\bar{\boldsymbol{y}}(v)))$.

We next consider the following problem taking y_1 as a parameter:

$$(\mathrm{Q}(y_1; v)) \left| \begin{array}{ll} \text{minimize} & \sum_{(i,j) \in E} c_{ij} x_{ij} \\ \text{subject to} & \boldsymbol{x} \in D(y_1, v - y_1). \end{array} \right.$$

We solve this problem, (i) increasing the value of y_1 from $\bar{y}_1(v)$, and then (ii) decreasing the value of y_1 from $\bar{y}_1(v)$. Apparently, the minimum of the optimal values of $(\mathrm{Q}(y_1; v))$ provides the left-hand-side value of $(\mathrm{T}(v))$. Let $\bar{\boldsymbol{x}}(y_1; v)$ denote an optimal solution and let

$$f(y_1; v) = f(y_1, v - y_1), \tag{3.11}$$

$$g(y_1; v) = \sum_{(i,j) \in E} c_{ij} \bar{x}_{ij}(y_1; v) = g(y_1, v - y_1), \tag{3.12}$$

$$h(y_1; v) = f(y_1, v - y_1) + g(y_1, v - y_1) = h(y_1, v - y_1). \tag{3.13}$$

Note that $\bar{\boldsymbol{x}}(y_1; v)$ is optimal to $(\mathrm{P}_2(y_1, v - y_1))$ and that $g(\cdot\,; v)$ is convex and piecewise affine at any y_1 such that $(y_1, v - y_1) \in \Omega$ (see e.g. [2, 4]).

Lemma 3.4. *Function $h(\cdot\,; v)$ is continuous at any y_1 such that $(y_1, v - y_1) \in \Omega$, and concave on each affine piece of $g(\cdot\,; v)$.*

Proof: Although the monotonicity may fail, $f(\cdot\,; v)$ is still a concave function. Hence the sum $h(\cdot\,; v)$ of $f(\cdot\,; v)$ and $g(\cdot\,; v)$ has the above properties. □

We see from Lemma 3.4 that there is a global minimum of $h(\cdot\,; v)$ among break points of $g(\cdot\,; v)$. We will generate all the break points by using the augmenting-path algorithm of Ford and Fulkerson [3] (see also [1]).

3.2. Enumeration of break points of $g(\cdot\,; v)$

Suppose an optimal solution $\bar{\boldsymbol{x}}(y';\, v)$ of $(\mathrm{Q}(y';\, v))$ is given for some $y' \geq \bar{y}_1(v)$. Based on $\bar{\boldsymbol{x}}(y';\, v)$, we define an auxiliary network $\mathcal{N} = (G' = (V,\, E_1 \cup E_2),\, 1,\, 2,\, \boldsymbol{c}',\, \boldsymbol{u}')$ according to the rules below: For each $(i,\, j) \in E$,

rule 1: if $\bar{x}_{ij}(y';\, v) < u_{ij}$, then $(i,\, j) \in E_1$ and let

$$u'_{ij} = u_{ij} - \bar{x}_{ij}(y';\, v), \quad c'_{ij} = c_{ij}, \tag{3.14}$$

rule 2: if $\bar{x}_{ij}(y';\, v) > 0$, then $(i,\, j) \in E_2$ and let

$$u'_{ji} = \bar{x}_{ij}(y';\, v), \quad c'_{ji} = -c_{ij}. \tag{3.15}$$

If there is no path from node 1 to 2 in G', it can be shown that $y_1 \leq y'$ for all y_1 such that $(y_1,\, v - y_1) \in \Omega$. Otherwise, we can find a flow augmenting path $\pi \subset E_1 \cup E_2$ from node 1 to 2 with the least cost, by solving a shortest path problem in G' with arc length $\boldsymbol{c}'$. Let

$$\bar{\delta} = \min\{u'_{ij} \mid (i,\, j) \in \pi\}. \tag{3.16}$$

Lemma 3.5. *Let $\delta \in [0,\, \bar{\delta}]$. Also, for each $(i,\, j) \in E$, let*

$$x'_{ij}(\delta) = \begin{cases} \bar{x}_{ij}(y';\, v) + \delta & \text{if} \quad (i,\, j) \in \pi \cap E_1, \\ \bar{x}_{ij}(y';\, v) - \delta & \text{if} \quad (j,\, i) \in \pi \cap E_2, \\ \bar{x}_{ij}(y';\, v) & \text{otherwise}. \end{cases} \tag{3.17}$$

Then $\boldsymbol{x}'(\delta)$ is optimal to $(Q(y' + \delta,\, v))$.

Proof: Follows from a well-known result on the augmenting-path algorithm for minimum cost flow problems (see e.g. [10]). □

According to (3.17), we can compute the optimal value $g(y' + \delta;\, v)$ of $(\mathrm{Q}(y' + \delta;\, v)$ as follows:

$$\begin{aligned} g(y' + \delta;\, v) &= \sum_{(i,j)\in E} c_{ij} x'_{ij}(\delta) \\ &= g(y';\, v) + \delta\Big(\sum_{(i,j)\in\pi\cap E_1} c_{ij} - \sum_{(j,i)\in\pi\cap E_2} c_{i,j}\Big). \end{aligned} \tag{3.18}$$

This expression implies that $g(\cdot\,; v)$ is an affine function over the interval $[y',\, y' + \bar{\delta}]$, and hence no $y_1 \in (y',\, y' + \bar{\delta})$ can provide a global minimum of $h(\cdot\,;\, v)$. We then let $\bar{\boldsymbol{x}}(y' + \delta'; v) = \boldsymbol{x}'(\delta')$ and jump to the next point $y' + \delta'$, where $\delta' = \min\{\bar{\delta},\, v - y'\}$.

To solve $(\mathrm{Q}(y_1;\, v))$ as increasing the value of y_1 from $\bar{y}_1(v)$, we begin with $y' = \bar{y}_1(v)$ and $\bar{\boldsymbol{x}}(y'; v) = \bar{\boldsymbol{x}}(v)$, and apply the above procedure iteratively. In case we cannot augment the flow from node 1 to 2 in the auxiliary network $\mathcal{N}$, we solve $(\mathrm{Q}(y_1;\, v))$ as decreasing y_1 from $\bar{y}_1(v)$ instead. This can be done in a similar way, where we need to find an augmenting path in $\mathcal{N}$ not from node 1 to 2 but from node 2 to 1.

Let us summarize the above procedure, which receives an integer $v \in [0, v_{\max}]$ and yields $\boldsymbol{y}^*(v) \in \operatorname{argmin}\{h(\boldsymbol{y}) \mid y_1 + y_2 = v, \ \boldsymbol{y} \in \Omega\}$, $\boldsymbol{x}^*(v) = \boldsymbol{x}^*(\boldsymbol{y}^*(v))$ and $h^*(v) = h(\boldsymbol{y}^*(v))$.

Procedure B(v).

0° Compute an optimal solution $(\bar{\boldsymbol{x}}(v), \bar{\boldsymbol{y}}(v))$ of (Q(v)). Let $(p, q) = (1, 2)$, $y' = \bar{y}_1(v)$ and $\bar{\boldsymbol{x}}(y'; v) = \bar{\boldsymbol{x}}(v)$. Also let $\boldsymbol{y}^*(v) = (y', v - y')$, $\boldsymbol{x}^*(v) = \bar{\boldsymbol{x}}(y'; v)$ and $h^*(v) = h(y'; v)$.

1° Construct the auxiliary network $\mathcal{N} = (G' = (V, E_1 \cup E_2), p, q, \boldsymbol{c}', \boldsymbol{u}')$ with respect to $\bar{\boldsymbol{x}}(y'; v)$ according to rules 1 and 2.

2° If there is no directed path from node p to q in G', then go to 5°. Otherwise, compute a shortest path π in G with arc length $\boldsymbol{c}'$ and let $\bar{\delta} = \min\{u'_{ij} \mid (i, j) \in \pi\}$.

3° Let $\delta' = \min\{\bar{\delta}, v - y'\}$. For each $(i, j) \in E$, let

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

Also let $y' = y' + \delta'$.

4° If $h(y'; v) < h^*(v)$, then update the incumbent:

$$\boldsymbol{y}^*(v) = \begin{cases} (y', v - y') & \text{if} \quad (p, q) = (1, 2), \\ (v - y', y') & \text{otherwise,} \end{cases}$$
$$\boldsymbol{x}^*(v) = \bar{\boldsymbol{x}}(y'; v), \quad h^*(v) = h(y'; v).$$

If $y' < v$, return to 1°.

5° If $(p, q) = (2, 1)$, then yield $(\boldsymbol{x}^*(v), \boldsymbol{y}^*(v), h^*(v))$. Otherwise, let $(p, q) = (1, 2)$, $y' = \bar{y}_2(v)$, $\bar{\boldsymbol{x}}(y'; v) = \bar{\boldsymbol{x}}(v)$ and return to 1°. □

Lemma 3.6. *Procedure B(v) requires $O(S(m, n)mU)$ arithmetic operations and $O(mU)$ evaluations of f, where $S(m, n)$ is the running time of a shortest path algorithm and* $U = \max\{u_{ij} \mid (i, j) \in E\}$.

Proof: Step 0° solves a maximum flow problem (Q(v)) in less than $S(m, n)mU$ arithmetic operations. Steps 1° – 4° are essentially the same as those of the augmenting-path algorithm for solving a minimum cost flow problem with m arcs and n nodes (see e.g.

[1, 3]). By integrity of u_{ij}'s, these steps are repeated at most $O(mU)$ times. At each iteration, the procedure solves a shortest path problem and evaluates f to compute $h(y'; v)$. □

Note that $(\boldsymbol{x}^*(v), \boldsymbol{y}^*(v))$ generated by the procedure is an integral flow in network $(\bar{G}, s, n, \mathbf{c}, \boldsymbol{u})$ so long as v is an integer.

3.3. ALGORITHM FOR SOLVING (P_2)

We are now ready to present the algorithm for solving (P_2).

Algorithm C.

Step 0. Compute $v_{\max}$ by solving problem (3.10). Let $\ell = 0$ and $r = v_{\max}$.

Step 1. Let $v = \lfloor(\ell + r)/2\rfloor$. Check (T($v$)) in the following manner:

(i) Call Procedure B(v) and obtain $(\boldsymbol{x}^*(v), \boldsymbol{y}^*(v), h^*(v))$.

(ii) If $h^*(v) > b$, then (T(v)) is true.

Step 2. if $v = \ell$, then terminate. (If (T(v)) is true, then (P_2) is infeasible. Otherwise, $(\boldsymbol{x}^*(v), \boldsymbol{y}^*(v))$ is optimal to (P_2).)

Step 3. If (T(v)) is true, then $r = v$. Otherwise, let $\ell = v + 1$. Return to Step 1. □

Theorem 3.7. *Algorithm C solves (P_2) in $O(S(m, n)mU \log mU)$ arithmetic operations and $O(mU \log mU)$ evaluations of f.*

Proof: Step 0 requires less than $S(m, n)mU$ arithmetic operations. The binary search procedure corresponding to Steps 1 – 3 is applied to the interval $[0, v_{\max}]$, and hence requires $\log v_{\max}$ iterations. This together with Lemma 3.6 proves the running time of the algorithm. The correctness follows Corollary 3.3, Lemmas 3.4 and 3.5. □

In contrast to (P_1), the concavity of f properly works in (P_2), and so we have to examine every local minimum of $h(\cdot\,; v)$ by using Procedure B(v), Consequently, the worst-case number of arithmetic operations required by Algorithm C is not polynomial but pseudo-polynomial in the problem input length. However, Procedure B(v) is nothing but the augmenting-path algorithm of Ford and Fulkerson [3], which is known to be practically efficient for networks of middle sizes. We can therefore expect that Algorithm C is also reasonably efficient for such a network unless evaluations of f are extremely expensive.

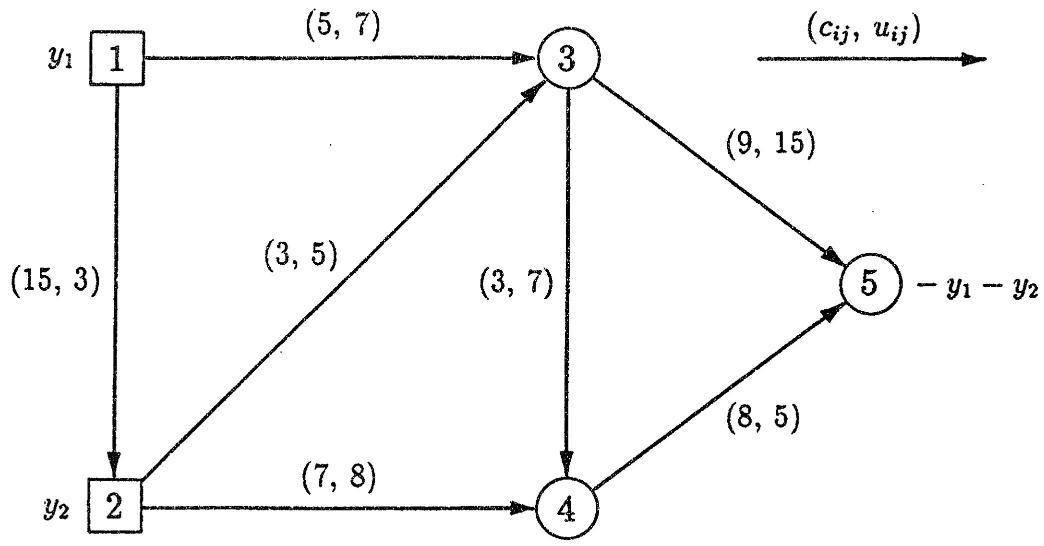

Figure 3.1. Example of (P_2).

3.4. Numerical example

Before concluding this section, let us illustrate Algorithm C by using a simple instance of (P_2), given by the network in Figure 3.1. The production cost of factories 1 and 2 is assumed to be

$$f(y) = 8.0\,(y_1^{1/2} + 2.0\,y_2^{1/2}),$$

and the budget capacity $b = 120.0$.

In Step 0 of the algorithm, we install an artificial node s and arcs $(s,\ 1)$, $(s,\ 2)$ in G and denote the resulting graph by $\bar{G}$. The maximum flow value from s to 5 in $\bar{G}$ is 17. Hence the interval to be searched is $[\ell,\ r] = [0,\ 17]$.

Iteration 1: We let $v = \lfloor 17/2 \rfloor = 8$ and call Procedure B(v) to check if (T(8)) is true or false.

In Procedure B(8), we first solve a minimum cost flow problem (Q(8)) in network $(\bar{G},\ s,\ 5,\ \boldsymbol{c},\ \boldsymbol{u})$ and obtain an optimal solution $(\bar{\boldsymbol{x}}(8),\ \bar{\boldsymbol{y}}(8))$ as shown in Figure 3.2. Since $\bar{\boldsymbol{x}}(8)$ is also optimal to (Q(3; 8)), we let $\bar{\boldsymbol{x}}(3;\ 8) = \bar{\boldsymbol{x}}(8)$ and

$$f(3;\ 8) = f(3,\ 5) = 49.63,$$
$$g(3;\ 8) = g(3,\ 5) = \sum c_{ij}\bar{x}_{ij}(3;\ 8) = 102,$$
$$h(3;\ 8) = h(3,\ 5) = f(3,\ 5) + g(3,\ 5) = 151.63.$$

Based on $\bar{\boldsymbol{x}}(3;\ 8)$, we construct the auxiliary network $\mathcal{N}$ as shown in Figure 3.3.

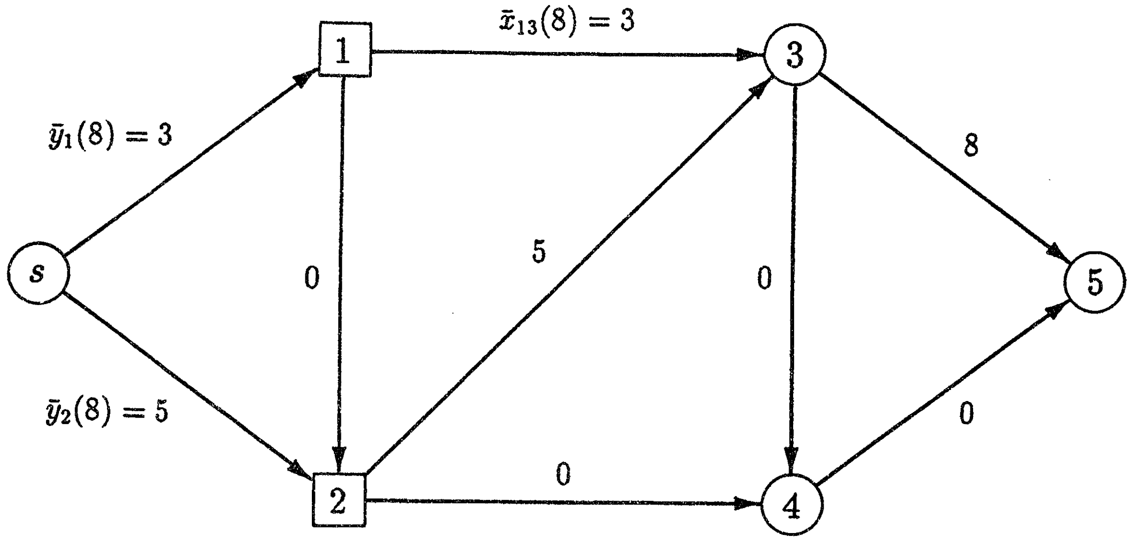

Figure 3.2. Optimal solution of (Q(8)).

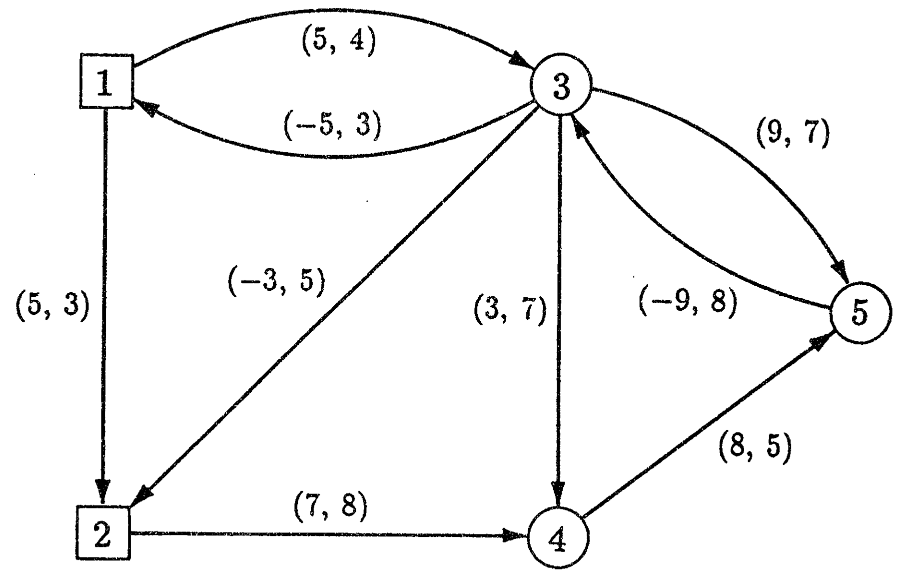

Figure 3.3. Auxiliary network $\mathcal{N}$ with respect $\bar{\boldsymbol{x}}(3;\, 8)$.

We next solve $(\mathrm{Q}(y'; 8))$ as increasing the value of y' from $\bar{y}_1(8) = 3$. By adding flows along augmenting paths from node 1 to 2 step by step in $\mathcal{N}$, we generate

$$f(7;\,8) = 37.17,\;\; g(7;\,8) = 110,\;\; h(7;\,8) = 147.17;$$
$$f(8;\,8) = 22.63,\;\; g(8;\,8) = 125,\;\; h(8;\,8) = 147.63.$$

Then we solve $(\mathrm{Q}(y'; 8))$ as decreasing y' from $\bar{y}_1(8) = 3$ and have

$$f(0;\,8) = 45.26,\;\; g(0;\,8) = 105,\;\; h(0;\,8) = 150.26.$$

Function $f(\,\cdot\,; 8)$ is affine on the intervals $[3,\,7]$, $[7,\,8]$ and $[0,\,3]$, on each of which $h(\,\cdot\,; 8)$ is concave. Since $h(7; 8) = \min\{h(3; 8),\, h(7; 8),\, h(8; 8),\, h(0; 8)\}$, the procedure yields $\boldsymbol{x}^*(8) = \bar{\boldsymbol{x}}(7; 8)$ and

$$\boldsymbol{y}^*(8) = (7,\, 8-7) = (7,\, 1),\;\; h^*(8) = h(7;\,8) = 147.17.$$

Thus we see that $h^*(8) > b = 120.0$ and $(\mathrm{T}(8))$ is true. We then reduce the interval $[\ell,\, r]$ to $[0,\, 8]$.

Iteration 2: Letting $v = \lfloor 8/2 \rfloor = 4$, we call Procedure $\mathrm{B}(v)$ to check $(\mathrm{T}(4))$.

The procedure examines $y' = 0,\, 4$, and compares

$$f(0;\,4) = 32.00,\;\; g(0;\,4) = 48,\;\; h(0;\,4) = 80.00;$$
$$f(4;\,4) = 16.00,\;\; g(4;\,4) = 56,\;\; h(4;\,4) = 72.00.$$

Then it yields $\bar{\boldsymbol{x}}^*(4) = \bar{\boldsymbol{x}}(4; 4)$ and

$$\boldsymbol{y}^*(4) = (4,\, 4-4) = (4,\, 0),\;\; h^*(4) = h(4;\,4) = 72.00.$$

Since $h^*(4) \leq b$ and $(\mathrm{T}(4))$ is false, we let $\ell = 4 + 1 = 5$.

Iteration 3: Letting $v = \lfloor (5+8)/2 \rfloor = 6$, we call Procedure $\mathrm{B}(v)$. Then it yields $\boldsymbol{x}^*(6) = \bar{\boldsymbol{x}}(6; 6)$ and

$$\boldsymbol{y}^*(6) = (6,\, 0),\;\; h^*(6) = 103.60.$$

Since $h^*(6) \leq b$ and $(\mathrm{T}(6))$ is false, we let $\ell = 6 + 1 = 7$.

Iteration 4: We let $v = \lfloor (7+8)/2 \rfloor = 7$. Since the value of v reaches $\ell = 7$, we call Procedure $\mathrm{B}(7)$ and then terminate Algorithm C. The procedure yields $\boldsymbol{x}^*(7) = \bar{\boldsymbol{x}}(7; 7)$ and

$$\boldsymbol{y}^*(7) = (7,\, 0),\;\; h^*(7) = 119.17.$$

Since $h^*(7) \leq b$, a globally optimal solution of our instance is given by $(\boldsymbol{x}^*(7),\, \boldsymbol{y}^*(7))$, shown in Figure 3.4.

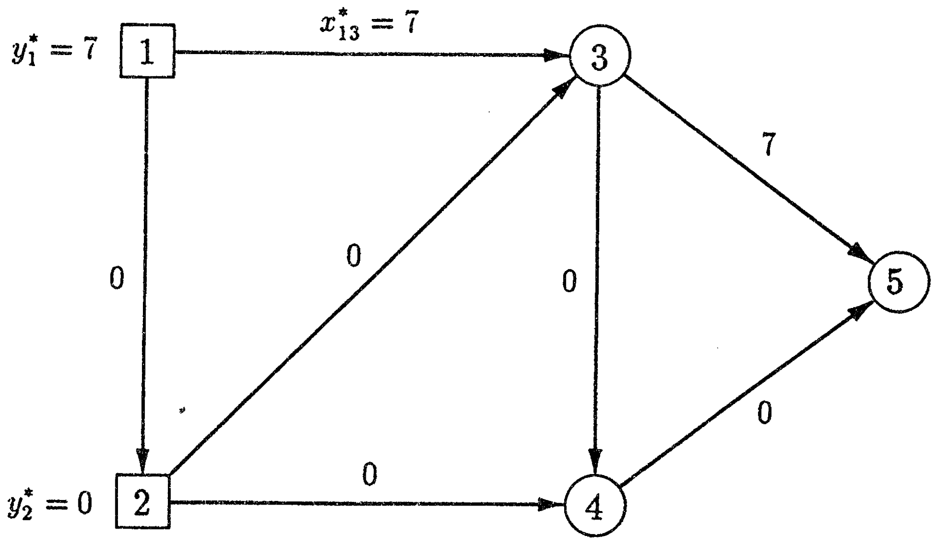

Figure 3.4. Optimal solution of the instance.

4. Conclusion

We showed in this paper that a parametric approach provides efficient algorithms for solving a class of network flow problems with an additional reverse convex constraint. Algorithm A we proposed to solve problem (P_1) can yield a globally optimal solution in polynomial time. While (P_1) with a single nonlinear variable is not essentially a global optimization problem, in problem (P_2) with two nonlinear variables the concavity of the constraint function works properly. Hence (P_2) can have multiple local minima, many of which fail to be globally optimal. To solve this multiextremal problem, we incorporated a global optimization technique into the algorithm. In consequence, a globally optimal solution of (P_2) turned out to be obtained in pseudo-polynomial time if we use Algorithm C. Computational experiments are now under way, the results of which will be reported elsewhere.

Besides problems (P_1) and (P_2), parametric approaches are very effective for solution to certain nonconvex network flow problems and studied in several articles [8, 11, 12, 13, 14, 18]. Especially in a series [15, 16, 17], Tuy *et al.* show that a parametric algorithm solves minimum concave-cost flow problems with a fixed number of sources and nonlinear arcs in strongly polynomial time. The readers are also referred to [6, 7] for the current state-of-the-art of general 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] Gal, T., *Postoptimal Analyses, Parametric Programming, and Related Topics*, McGraw-Hill (N.Y., 1979).

[5] Goldberg, A.V. and R.E. Tarjan, "Finding minimum-cost circulations by canceling negative cycles", *Journal of ACM* **36** (1988

[6] Guisewite, G.M., "Network problems", in R. Horst and P.M. Pardalos (eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers (Dortrecht, 1995), 609 - 648.

[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] Iri, M., "A new method of solving transportation network problems", *Journal of the Operations Research Society of Japan* **3** (1960), 27 - 87.

[11] Klinz, B. and H. Tuy, "Minimum concave-cost network flow problems with a single nonlinear arc cost," Dungzhu Du and P.M. Pardalos (eds.), *Network Optimization Problems*, World Scientific (Singapore, 1993), 125 - 143.

[12] 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.*

[13] 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).

[14] Kuno, T. and T. Utsunomiya, "Minimizing a linear multiplicative-type function under network flow constrains", Technical Report ISE-TR-95-124, Institute of Information Sciences and Electronics, University of Tsukuba (Tsukuba, 1995).

[15] 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.

[16] 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.

[17] Tuy, H., S. Ghannadan, A. Migdalas and P. Värbrand, "The minimum concave cost network flow problems with fixed number of sources and nonlinear arc costs", *Journal of Global Optimization* 6 (1995), 135 - 151.

[18] Yajima, Y. and H. Konno, "An algorithm for a concave production cost network flow problem", Research Report B-299, Department of Information Sciences, Tokyo Institute of Technology (Tokyo, 1995).