Journal of Global Optimization **18:** 59–73, 2000.

A Lagrangian Based Branch-and-Bound Algorithm for Production-transportation Problems

TAKAHITO KUNO* and TAKAHIRO UTSUNOMIYA†
Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan (*e-mail*: *takahito@is.tsukuba.ac.jp*)

(Received 4 February 1999; accepted in revised form 5 October 1999)

Abstract. We propose a branch-and-bound algorithm of Falk–Soland's type for solving the minimum cost production-transportation problem with concave production costs. To accelerate the convergence of the algorithm, we reinforce the bounding operation using a Lagrangian relaxation, which is a concave minimization but yields a tighter bound than the usual linear programming relaxation in $O(mn \log n)$ additional time. Computational results indicate that the algorithm can solve fairly large scale problems.

Key words: Global optimization; Production-transportation problem; Minimum concave-cost flow problem; Branch-and-bound algorithm; Lagrangian relaxation

1. Introduction

The production-transportation problem is a model for determining both optimal production at some factories manufacturing a single commodity and optimal transportation of the products to warehouses with known demands. Let $G = (M, N, A)$ be a bipartite graph consisting of a set M of m factories, a set N of n warehouses and a set $A = M \times N$ of mn transportation routes. At each factory $i \in M$ the cost of producing y_i units is $f_i(y_i)$ and the production capacity is u_i units. Due to economy of scale, the production cost $f_i : \mathbb{R} \to \mathbb{R}$ is assumed to be a nonlinear, concave and nondecreasing function. At each warehouse $j \in N$ there is a demand of b_j units. The cost of shipping a unit by route $(i, j) \in A$ is c_{ij}. The problem is then formulated as follows:

$$\begin{array}{ll} \text{minimize} & z = \displaystyle\sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{i\in M} f_i(y_i) \\ \text{subject to} & \displaystyle\sum_{j\in N} x_{ij} \leqslant y_i\,, \quad 0 \leqslant y_i \leqslant u_i\,,\ i \in M \\ & \displaystyle\sum_{i\in M} x_{ij} = b_j\,,\ j \in N \\ & x_{ij} \geqslant 0\,,\ (i, j) \in A\,, \end{array} \tag{1.1}$$

where x_{ij}s and y_is are variables to be determined.

* The author was supported in part by Grant-in-Aid for Scientific Research of the Ministry of Education, Science, Sports and Culture, Grant No. (C2)09680413.

† Now at the Prefectural Office of Ibaraki.

The problem (1.1) belongs to the class of capacitated minimum concave-cost flow problems. The main characteristic of (1.1) is that the number m of nonlinear concave variables is rather small in comparison with the number mn of linear variables. Over the past ten years, seveal parametric algorithms using this *low-rank concavity* [14] have been proposed to solve the problem with fixed m [9, 13, 16, 17, 23–25]. Some of them work even if the total production cost is given as a nonseparable function $f : \mathbb{R}^m \to \mathbb{R}$ (e.g. [23–25]). This family of algorithms is polynomial or pseudo-polynomial in n and the most efficient for small m, especially when $m \leqslant 3$. Unfortunately, however, it is exponential in m and will be of no practical use if m exceeds five at most. Therefore, brnach-and-bound is still thought of as a reliable and effective approach to the problem (1.1) with larger m, as to general minimum concave-cost flow problems. Branch-and-bound algorithms applicable to (1.1) are classified roughly into two types. The first type uses the fact that at least one basic solution is optimal to (1.1), and implicitly enumerates the spanning trees of G [5, 8]. The second type exploits the sebarability of the objective function and successively improves its linear underestimator by dividing the feasible set. The basis of this approach is found in Falk and Soland [4]. While they did not assume any network structures, Soland modified their algorithm later to handle a production-transportation problem with concave transportation costs [20]. In a textbook [11], Horst et al. have also applied Falk–Soland's algorithm to (1.1). The readers are referred to [7, 18] for comprehensive reviews on minimum concave-cost flow problems.

In this paper, we develop a branch-and-bound algorithm of the second type to solve the problem (1.1) with m larger than five. The branching operation in our algorithm is similar to that of Soland, which divides the feasible set of nonlinear variables to generate subproblems; but the bounding operation fully exploits the problem structures and is implemented through two stages: the first stage based on a linear programming relaxation and the second stage based on a Lagrangian relaxation. Although the Lagrangian relaxation of each subproblem is a concave minimization, this two-stage bounding operation provides a lower bound for it much tighter than that of Soland in $O(mn \log n)$ additional computational time. In Section 2, both the relaxations are given in detail. Section 3 is devoted to the algorithm incorporating the two-stage bounding operation. Computational results of the algorithm are reported in Section 4. Some final remarks are discussed in Section 5.

2. Relaxations

We assume throughout the paper that both the production capacity u_i at each factory $i \in M$ and the demand b_j at each warehouse $j \in N$ are positive integers; and that the unit transportation cost c_{ij} by each route $(i, j) \in A$ is a nonnegative real number. We also assume that

$$B \equiv \sum_{j \in N} b_j \leqslant \sum_{i \in M} u_i \,. \tag{2.1}$$

Otherwise, the problem (1.1) has no feasible solutions. Since the objective function is continuous and the feasible set is a polytope, (1.1) always has an optimal solution under condition (2.1). In addition to this, the concave objective function achieves its minimum at some vertex of the feasible set. By total unimodularity, all the vertices are integral vectors as long as b_js and u_is are integers (see e.g. [2]). Therefore, once we assume (2.1), the problem (1.1) has an integral optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ such that

$$\sum_{j \in N} x_{ij}^{*} = y_i^* , \quad i \in M \tag{2.2}$$

because of the monotonicity of f_is.

Using (2.2), let us rewrite (1.1) as follows:

$$\text{PTP} \left| \begin{array}{ll} \text{minimize} & z = f(\boldsymbol{x}, \boldsymbol{y}) \equiv \displaystyle\sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{i \in M} f_i(y_i) \\ \text{subject to} & \displaystyle\sum_{j \in N} x_{ij} = y_i , \quad i \in M \\ & \displaystyle\sum_{i \in M} x_{ij} = b_j , \quad j \in N \\ & \boldsymbol{x} \geqslant \boldsymbol{0} , \qquad \boldsymbol{y} \in Y , \end{array} \right.$$

where $\boldsymbol{x} = (x_{ij} \,|\, (i, j) \in A)$ and $\boldsymbol{y} = (y_i \,|\, i \in M)$ are the vectors of linear and nonlinear variables, respectively, and

$$Y = [0, u_1] \times \cdots \times [0, u_m] .$$

As mentioned in Section 1, we apply a branching operation on the set Y recursively and divide it into a number of subsets Y^k, $\boldsymbol{k} = 1, \ldots, r$. The division $\mathcal{Y} = \{Y^k \,|\, k = 1, \ldots, r\}$ we employ is an *integral rectangular partition* of Y, i.e. for each $k = 1, \ldots, r$, we have

$$Y^k = I_1^k \times \cdots \times I_m^k ; \quad I_i^k = [l_i^k, u_i^k] , \quad i \in M ,$$

where l_i^ks and u_i^ks are integers; and

$$\bigcup_{k=1}^{r} Y^k = Y; \quad \text{int } Y^k \cap \text{int } Y^h = \emptyset \text{ if } k \neq h .$$

Associated with each partition set $Y^k \in \mathcal{Y}$ we define a subproblem:

$$\text{P}^k \left| \begin{array}{ll} \text{minimize} & z = f(\boldsymbol{x}, \boldsymbol{y}) \\ \text{subject to} & \displaystyle\sum_{j \in N} x_{ij} = y_i , \quad i \in M \\ & \displaystyle\sum_{i \in M} x_{ij} = b_j , \quad j \in N \\ & \boldsymbol{x} \geqslant \boldsymbol{0} , \qquad \boldsymbol{y} \in Y^k . \end{array} \right.$$

The following are immediate consequences:

PROPOSITION 2.1. (i) *Problem P^k has an optimal solution if and only if*

$$\sum_{i\in M} l_i^k \leqslant B \leqslant \sum_{i\in M} u_i^k . \tag{2.3}$$

If (2.3) holds, P^k has at least one integral optimal solution $(\boldsymbol{x}^k, \boldsymbol{y}^k)$.

(ii) *Let $z^k = f(\boldsymbol{x}^k, \boldsymbol{y}^k)$ if (2.3) holds; $z^k = +\infty$ otherwise. Then the optimal value of PTP is given by*

$$z^* = \min\{z^k \,|\, k = 1, \ldots, r\} .$$

Given a rectangular partition $\mathcal{Y} = \{Y^k \,|\, k = 1, \ldots, r\}$ of Y, let us take a set $Y^k \in \mathcal{Y}$ satisfying condition (2.3) and consider the associated subproblem P^k. As a beginning, we will present a linear programming relaxation of P^k, which minimizes the convex envelop of the objective function f as do both Falk–Soland's and Soland's relaxations [4, 20].

2.1. LINEAR PROGRAMMING RELAXATION

The convex envelop of f_i over the interval $I_i^k = [l_i^k, u_i^k]$ is defined by

$$\bar{f}_i(y_i) = \frac{f_i(u_i^k) - f_i(l_i^k)}{u_i^k - l_i^k}(y_i - l_i^k) + f_i(l_i^k) , \quad i \in M ,$$

which satisfies

$$\bar{f}_i(y_i) \leqslant f_i(y_i) \text{ if } y_i \in I_i^k ; \quad \bar{f}_i(y_i) \geqslant f_i(y_i) \text{ if } y_i \notin \operatorname{int} I_i^k . \tag{2.4}$$

Hence, replacing f_i by $\bar{f}_i$ in P^k for each $i \in M$, we have a linear program yielding a lower bound for z^k. Actually, in Falk–Soland [4] this linear program is used directly as a relaxation of P^k. In Soland [20], the set of constraints $\boldsymbol{y} \in Y^k$ is further relaxed into $\boldsymbol{y} \in Y$, to make a Hitchcock problem of the resulting problem. We adopt a compromise of these relaxations and replace $\boldsymbol{y} \in Y^k$ by

$$0 \leqslant y_i \leqslant u_i^k , \quad i \in M . \tag{2.5}$$

If we delete y_i by substituting $y_i = \Sigma_{j\in N} x_{ij}$ into (2.5) for each $i \in M$, our relaxation is also reduced to a Hitchcock problem:

$$\bar{\mathrm{P}} \left| \begin{array}{ll} \text{minimize} & z = \bar{f}(\boldsymbol{x}) \equiv \displaystyle\sum_{(i,j)\in A} c_{ij}^k x_{ij} + d^k \\ \text{subject to} & \displaystyle\sum_{j\in N} x_{ij} \leqslant u_i^k , \quad i \in M \\ & \displaystyle\sum_{i\in M} x_{ij} = b_j , \quad j \in N \\ & \boldsymbol{x} \geqslant \boldsymbol{0} , \end{array} \right.$$

where

$$\left.\begin{aligned} c_{ij}^k &= c_{ij} + \frac{f_i(u_i^k) - f_i(l_i^k)}{u_i^k - l_i^k}, \quad (i, j) \in A \\ d^k &= \sum_{i \in M} \left(f_i(l_i^k) - \frac{f_i(u_i^k) - f_i(l_i^k)}{u_i^k - l_i^k} l_i^k \right). \end{aligned}\right\} \tag{2.6}$$

THEOREM 2.2. *Problem* $\bar{P}$ *has an integral optimal solution* $\bar{\boldsymbol{x}}$. *Let*

$$\bar{y}_i = \sum_{j \in N} \bar{x}_{ij}, \ i \in M; \quad \bar{z} = \bar{f}(\bar{\boldsymbol{x}}).$$

Then $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ *is feasible to PTP and we have*

$$\bar{z} \leqslant z^k; \quad f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \geqslant z^*.$$

Proof. Follows from the construction of $\bar{\mathrm{P}}$. □

We can thus use $\bar{z}$ as a lower bound for z^k and terminate branching at subproblem P^k unless $\bar{z}$ is less than the value of each feasible solution of PTP in hand. Although the bound $\bar{z}$ might not be as tight as Falk–Soland's, it is tighter than Soland's; and besides, $\bar{\mathrm{P}}$ yielding $\bar{z}$ is a simple Hitchcock problem like Soland's. Nevertheless, the bound $\bar{z}$ is not tight enough yet to improve branch-and-bound algorithms of the second type drastically. To solve PTP with large m, we have to devise another relaxation of P^k yielding a much tighter bound than $\bar{\mathrm{P}}$.

2.2 LAGRANGIAN RELAXATION

Let us introduce a Lagrangian multiplier vector $\boldsymbol{\lambda} = (\lambda_j \mid j \in N)$ for relaxing the set of constraints $\Sigma_{i \in M} x_{ij} = b_j$, $j \in N$. By noting that x_{ij} cannot exceed b_j for each $(i, j) \in A$, we write the resulting problem as follows:

$$\mathrm{L}(\boldsymbol{\lambda}) \left| \begin{array}{ll} \text{minimize} & z = \phi(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda}) \equiv \displaystyle\sum_{(i,j) \in A} c_{ij}(\boldsymbol{\lambda}) x_{ij} + \sum_{i \in M} f_i(y_i) + \sum_{j \in N} \lambda_j b_j \\ \text{subject to} & \displaystyle\sum_{j \in N} x_{ij} = y_i, \quad i \in M \\ & 0 \leqslant x_{ij} \leqslant b_j, \quad (i, j) \in A, \ \boldsymbol{y} \in Y^k, \end{array} \right.$$

where

$$c_{ij}(\boldsymbol{\lambda}) = c_{ij} - \lambda_j, \quad (i, j) \in A. \tag{2.7}$$

This is the second relaxation of P^k and plays the central role in our algorithm. As is well-known (see e.g. [19]), we have the following:

LEMMA 2.3. *Let* $(\boldsymbol{x}(\boldsymbol{\lambda}), \boldsymbol{y}(\boldsymbol{\lambda}))$ *denote an optimal solution of* $L(\boldsymbol{\lambda})$ *and let* $z(\boldsymbol{\lambda}) = \phi(\boldsymbol{x}(\boldsymbol{\lambda}), \boldsymbol{y}(\boldsymbol{\lambda}); \boldsymbol{\lambda})$. *Then*

$$z(\boldsymbol{\lambda}) \leqslant z^k, \quad \forall \boldsymbol{\lambda} \in \mathbb{R}^n.$$

The question here is how we should choose a value of λ_i for each $i \in N$ such that $z(\boldsymbol{\lambda}) > \bar{z}$. To answer this, we need to consider a linear programming relaxation of $\mathrm{L}(\boldsymbol{\lambda})$. In the same way as we have constructed $\bar{\mathrm{P}}$, we can linearize $\mathrm{L}(\boldsymbol{\lambda})$ using $\bar{f}_i$s into

$$\left|\begin{array}{ll} \text{minimize} & \bar{\phi}(\boldsymbol{x};\, \boldsymbol{\lambda}) \equiv \displaystyle\sum_{(i,j)\in A} (c_{ij}^k - \lambda_j) x_{ij} + d^k + \sum_{j\in N} \lambda_j b_j \\ \text{subject to} & \displaystyle\sum_{j\in N} x_{ij} \leqslant u_i^k, \quad i \in M \\ & 0 \leqslant x_{ij} \leqslant b_j, \quad (i,j) \in A, \end{array}\right. \tag{2.8}$$

where c_{ij}^ks and d^k are defined in (2.6). The dual problem of (2.8) is then written as follows:

$$\left|\begin{array}{ll} \text{maximize} & \psi(\boldsymbol{\mu}, \boldsymbol{\nu};\, \boldsymbol{\lambda}) \equiv -\displaystyle\sum_{i\in M} u_i^k \mu_i - \sum_{(i,j)\in A} b_j \nu_{ij} + d^k + \sum_{j\in N} \lambda_j b_j \\ \text{subject to} & -\mu_i - \nu_{ij} \leqslant c_{ij}^k - \lambda_j, \quad (i,j) \in A \\ & \boldsymbol{\mu} \geqslant \mathbf{0}, \ \ \boldsymbol{\nu} \geqslant \mathbf{0}, \end{array}\right. \tag{2.9}$$

where $\boldsymbol{\mu} = (\mu_i \,|\, i \in M)$ and $\boldsymbol{\nu} = (\nu_{ij} \,|\, (i,j) \in A)$ are the vectors of dual variables. Let $\bar{\boldsymbol{x}}(\boldsymbol{\lambda})$ and $(\bar{\boldsymbol{\mu}}(\boldsymbol{\lambda}), \bar{\boldsymbol{\nu}}(\boldsymbol{\lambda}))$ denote optimal solutions of (2.8) and (2.9), respectively. From the duality theorem in linear programming, we have

$$\begin{aligned} \max_{\lambda\in\mathbb{R}^n} \bar{\phi}(\bar{\boldsymbol{x}}(\boldsymbol{\lambda});\, \boldsymbol{\lambda}) &= \max_{\lambda\in\mathbb{R}^n} \psi(\bar{\boldsymbol{\mu}}(\boldsymbol{\lambda}), \bar{\boldsymbol{\nu}}(\boldsymbol{\lambda});\, \boldsymbol{\lambda}) \\ &= \max_{\mu,\nu,\lambda} \left\{ \psi(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\lambda}) \,\middle|\, \begin{array}{l} -\mu_i - \nu_{ij} \leqslant c_{ij}^k - \lambda_j, \ (i,j) \in A \\ \boldsymbol{\mu} \geqslant \mathbf{0}, \ \ \boldsymbol{\nu} \geqslant \mathbf{0} \end{array} \right\}. \end{aligned} \tag{2.10}$$

LEMMA 2.4. *The right-hand-side of (2.10) has a maximum point* $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}}, \bar{\boldsymbol{\lambda}})$ *with* $\bar{\boldsymbol{\nu}} = \mathbf{0}$.

Proof. Let $(\boldsymbol{\mu}', \boldsymbol{\nu}', \boldsymbol{\lambda}')$ with $\boldsymbol{\nu}' \neq \mathbf{0}$ be a maximum point. Also let

$$\bar{\lambda}_j = \lambda'_j - \sum_{i\in M} \nu'_{ij}, \ \ j \in N; \quad \bar{\mu}_i = \mu'_i, \ \ i \in M; \ \ \bar{\nu}_{ij} = 0, \ \ (i,j) \in A.$$

Then we have

$$\begin{aligned} &-\bar{\mu}_i - \bar{\nu}_{ij} \leqslant c_{ij}^k - \lambda'_j + \nu'_{ij} \leqslant c_{ij}^k - \lambda'_j + \sum_{h\in M} \nu'_{hj} = c_{ij}^k - \bar{\lambda}_j, \quad (i,j) \in A \\ &\psi(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}}, \bar{\boldsymbol{\lambda}}) = -\sum_{i\in M} u_i^k \bar{\mu}_i + d^k + \sum_{j\in N} \bar{\lambda}_j b_j \\ &\qquad = -\sum_{i\in M} u_i^k \mu' + d^k + \sum_{j\in N} \Big(\lambda'_j - \sum_{i\in M} \nu'_{ij}\Big) b_j = \psi(\boldsymbol{\mu}', \boldsymbol{\nu}', \boldsymbol{\lambda}'). \end{aligned}$$

This implies that $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}}, \bar{\boldsymbol{\lambda}})$ is a maximum point as well. □

Note that, if we set $\boldsymbol{\nu} = \mathbf{0}$ in (2.10), it coincides with the dual problem of $\bar{\mathrm{P}}$:

$$\left|\begin{array}{ll} \text{maximize} & g(\boldsymbol{\mu}, \boldsymbol{\lambda}) \equiv \sum_{j \in N} b_j \lambda_j - \sum_{i \in M} u_i^k \mu_i + d^k \\ \text{subject to} & \lambda_j - \mu_i \leqslant c_{ij}^k, \quad (i, j) \in A \\ & \boldsymbol{\mu} \geqslant \mathbf{0}. \end{array}\right. \tag{2.11}$$

Hence, we immediately have the following lemma:

LEMMA 2.5. *Let* $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}})$ *be an optimal solution of (2.11). Then*

$$\max_{\lambda \in \mathbb{R}^n} \bar{\phi}(\bar{\boldsymbol{x}}(\boldsymbol{\lambda}); \boldsymbol{\lambda}) = g(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}) = \bar{z}.$$

THEOREM 2.6. *Let* $(\boldsymbol{x}(\boldsymbol{\lambda}), \boldsymbol{y}(\boldsymbol{\lambda}))$ *denote an optimal solution of* $L(\boldsymbol{\lambda})$ *and let* $z(\boldsymbol{\lambda}) = \phi(\boldsymbol{x}(\boldsymbol{\lambda}), \boldsymbol{y}(\boldsymbol{\lambda}); \boldsymbol{\lambda})$. *Then*

$$\bar{z} \leqslant z(\bar{\boldsymbol{\lambda}}) \leqslant z^k,$$

where the first inequality holds strictly if f_i *is strictly concave on* $I_i^k = [l_i^k, u_i^k]$ *and* $y_i(\bar{\boldsymbol{\lambda}}) \in \operatorname{int} I_i^k$ *for some* $i \in M$.

Proof. It follows from Lemma 2.5 that

$$\bar{\boldsymbol{\lambda}} \in \arg \max_{\lambda \in \mathbb{R}^n} \bar{\phi}(\bar{\boldsymbol{x}}(\boldsymbol{\lambda}); \boldsymbol{\lambda}).$$

Since (2.8) with $\boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}}$ is a relaxation of $L(\bar{\boldsymbol{\lambda}})$, we have

$$z(\bar{\boldsymbol{\lambda}}) = \phi(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}), \boldsymbol{y}(\bar{\boldsymbol{\lambda}}); \bar{\boldsymbol{\lambda}}) \geqslant \bar{\phi}(\bar{\boldsymbol{x}}(\bar{\boldsymbol{\lambda}}); \bar{\boldsymbol{\lambda}}) = \bar{z}.$$

If $y_i(\bar{\boldsymbol{\lambda}}) \in \operatorname{int} I_i^k$ and f_i is strictly concave on I_i^k for some $i \in M$, we have $f_i(y_i(\bar{\boldsymbol{\lambda}})) > \bar{f}_i(y_i(\bar{\boldsymbol{\lambda}}))$ by definition; hence,

$$\phi(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}), \boldsymbol{y}(\bar{\boldsymbol{\lambda}}); \bar{\boldsymbol{\lambda}}) > \bar{\phi}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}); \bar{\boldsymbol{\lambda}}) \geqslant \bar{\phi}(\bar{\boldsymbol{x}}(\bar{\boldsymbol{\lambda}}); \bar{\boldsymbol{\lambda}}). \qquad \square$$

While the bound $z(\bar{\boldsymbol{\lambda}})$ turned out to be tighter than $\bar{z}$, problem $L(\bar{\boldsymbol{\lambda}})$ yielding it is a concave minimization in contrast to $\bar{P}$. This means that $L(\bar{\boldsymbol{\lambda}})$ can have multiple local minima, many of which fail to be global ones. Therefore, as does P^k, the relaxed problem $L(\bar{\boldsymbol{\lambda}})$ itself belongs to the class of multiextremal global optimization which is known to be hard to solve in general [11, 12]. In the next section, however, we will show that the global minimum $z(\boldsymbol{\lambda})$ of $L(\boldsymbol{\lambda})$ can be computed in $O(mn \log n)$ arithmetic operations and in $O(mn)$ evaluations of f_is for a given $\boldsymbol{\lambda}$.

3. Algorithm

Let us again look at problem $L(\boldsymbol{\lambda})$ in detail. We then see that it can be decomposed into m minimization problems, each of which is of the form:

$$\left|\begin{array}{ll} \text{minimize} & z_i = \sum_{j\in N} c_{ij}(\boldsymbol{\lambda})x_{ij} + f_i(y_i) \\ \text{subject to} & \sum_{j\in N} x_{ij} = y_i \\ & 0 \leqslant x_{ij} \leqslant b_j\,, \quad j \in N\,,\ y_i \in I_i^k\,. \end{array}\right. \tag{3.1}$$

Even though the objective function is concave, (3.1) turns into a continuous linear knapsack problem once we fix the value of y_i in the interval $I_i^k = [l_i^k, u_i^k]$. Suppose that the variables x_{ij}, $j \in N$, are arranged in the order

$$c_{ij_1}(\boldsymbol{\lambda}) \leqslant \cdots \leqslant c_{ij_p}(\boldsymbol{\lambda}) \leqslant 0 \leqslant c_{ij_{p+1}}(\boldsymbol{\lambda}) \leqslant \cdots \leqslant c_{ij_n}(\boldsymbol{\lambda})\,. \tag{3.2}$$

Then an optimal solution of (3.1) with a fixed y_i is given by

$$\tilde{x}_{ij_h}(y_i) = \begin{cases} b_{j_h}\,, & h = 1, \ldots, q-1 \\ y_i - \sum_{h=1}^{q-1} b_{j_h}\,, & h = q \\ 0\,, & h = q+1, \ldots, n\,, \end{cases} \tag{3.3}$$

if there is an index $q \leqslant p$ such that $\Sigma_{h=1}^{q-1} b_{j_h} \leqslant y_i < \Sigma_{h=1}^{q} b_{j_h}$; otherwise,

$$\tilde{x}_{ij_h}(y_i) = \begin{cases} b_{j_h}\,, & h = 1, \ldots, p \\ 0\,, & h = p+1, \ldots, n\,. \end{cases} \tag{3.4}$$

Let

$$F_i(y_i) = \sum_{j\in N} c_{ij}(\boldsymbol{\lambda})\tilde{x}_{ij}(y_i) + f_i(y_i)\,; \tag{3.5}$$

and let

$$\eta_0 = 0\,; \quad \eta_h = \eta_{h-1} + b_{j_h}\,, \quad h = 1, \ldots, n\,. \tag{3.6}$$

LEMMA 3.1. *The function F_i is concave on the interval $[\eta_{h-1}, \eta_h]$ for each $h = 1, \ldots, n$.*

Proof. We see from (3.2)–(3.6) that F_i is composed of a concave function f_i and n piecewise affine functions $c_{ij}(\boldsymbol{\lambda})\tilde{x}_{ij}(y_i)$, $j \in N$, each of which has at most one break point in $\{\eta_h \,|\, h = 0, 1, \ldots, n\}$. Since a sum of concave and affine functions is concave (see e.g. [3]), the function F_i is concave on each affine piece of $\Sigma_{j\in N} c_{ij}(\boldsymbol{\lambda})\tilde{x}_{ij}(y_i)$. □

Lemma 3.1 guarantees that F_i is minimized at some end point of $[\eta_h, \eta_{h-1}]$s over the whole interval $[\eta_0, \eta_n] = [0, B]$. Therefore, the optimal value of (3.1) is given by

$$z_i(\boldsymbol{\lambda}) = \min\{F_i(y_i) \,|\, y_i \in (I_i^k \cap \{\eta_h \,|\, h = 0, 1, \ldots, n\}) \cup \{l_i^k, u_i^k\}\}\,; \tag{3.7}$$

and the optimal value of $\mathrm{L}(\boldsymbol{\lambda})$ is

$$z(\boldsymbol{\lambda}) = \sum_{i \in M} z_i(\boldsymbol{\lambda}) + \sum_{j \in N} \lambda_j b_j \,.$$

THEOREM 3.2. *Given* $\boldsymbol{\lambda} \in \mathbb{R}^n$, *the lower bound* $z(\boldsymbol{\lambda})$ *can be computed in* $O(mn \log n)$ *arithmetic operations and* $O(mn)$ *evaluations of* f_i*s.*

Proof. For each $i \in M$, sorting $c_{ij}(\boldsymbol{\lambda})$ in the order (3.2) requires $O(n \log n)$ arithmetic operations; and (3.7) requires $O(n)$ evaluations of f_i. Their total numbers are $O(mn \log n)$ and $O(mn)$, respectively. □

The polynomial-time solvability of the concave minimization problem $\mathrm{L}(\boldsymbol{\lambda})$ is totally due to the *rank-two monotonicity* [14, 21] possessed by the objective function of (3.1). Functions of this class are certainly concave on their domains; but the concavity can be embedded into only a two-dimensional subspace, which enable us to effectively apply parametric programming like the above (see [14, 22] for further details). In Chapter 7.9 of his recent book [22], Tuy has discussed rank-two monotonic problems of the form (3.1). Theorem 3.2 can be thought of as a consequence of his result (Proposition 7.11 in [22]).

3.1. DESCRIPTION OF THE BRANCH-AND-BOUND ALGORITHM

Using the results obtained so far, we implement the bounding operation through two stages. Let z^0 denote the least value among feasible solutions of PTP in hand. At the first stage, we solve the linear programming relaxed problem $\bar{\mathrm{P}}$ for a given subproblem P^k; if the lower bound $\bar{z}$ is less than z^0, we set z^0 to $\min\{z^0, f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})\}$ and proceed to the second stage. At the second stage, we solve the Lagrangian relaxed problem $\mathrm{L}(\bar{\boldsymbol{\lambda}})$. Here, $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}})$ is an optimal dual solution of $\bar{\mathrm{P}}$ and hence can be computed in the process of the first stage. Unless the lower bound $z(\bar{\boldsymbol{\lambda}})$ is less than z^0, the subproblem P^k is fathomed.

If $z(\bar{\boldsymbol{\lambda}}) < z^0$, we implement the branching operation in the same way as in Soland [20]. Namely, we choose a factory node $s \in M$ such that

$$s \in \arg\max_{i \in M} \{f_i(\bar{y}_i) - \bar{f}_i(\bar{y}_i)\}\,, \tag{3.8}$$

and divide the corresponding interval $I_s^k = [l_s^k, u_s^k]$ into two subintervals $I_s^{k_1} = [l_s^k, \bar{y}_s]$ and $I_s^{k_2} = [\bar{y}_s, u_s^k]$. We then update the integral rectangular partition of Y as follows:

$$Y^{k_h} = I_1^k \times \cdots \times I_{s-1}^k \times I_s^{k_h} \times I_{s+1}^k \times \cdots \times I_m^k\,, \quad h = 1, 2$$
$$\mathcal{Y}' = (\mathcal{Y} \setminus Y^k) \cup \{Y^{k_1}, Y^{k_2}\}\,.$$

We should note that $\bar{y}_s$ falls in the open interval (l_s^k, u_s^k). Since we have

$$\bar{z} = \sum_{(i,j) \in A} c_{ij} \bar{x}_{ij} + \sum_{i \in M} \bar{f}_i(\bar{y}_i) < z^0 \leqq f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) = \sum_{(i,j) \in A} c_{ij} \bar{x}_{ij} + \sum_{i \in M} f_i(\bar{y}_i)\,,$$

at least one of the differences in (3.8) is positive. We see from (2.4) that this can happen only when $\bar{y}_s \in \operatorname{int} I_s^k$. Hence, by the integrality of $\bar{y}_i$, l_i^k and u_i^k for all $i \in M$,

the number of branching operations on each partition set must be finite. Also note that $\bar{\boldsymbol{y}}$ lies in both the partition sets Y^{k_1} and Y^{k_2} newly generated. This implies that $\bar{\boldsymbol{x}}$ is feasible to both $\bar{\mathrm{P}}^{k_1}$ and $\bar{\mathrm{P}}^{k_2}$ associated with Y^{k_1} and Y^{k_2}, respectively. We can therefore save the time and memory needed to solve one of them by using $\bar{\boldsymbol{x}}$ as a starting feasible solution if we employ the depth-first-search rule to choose a partition set from $\mathcal{Y}'$.

The branch-and-bound algorithm for PTP is summarized into a recursive form:

algorithm LP_LAGRANGE;
begin
  $B := \Sigma_{j\in N}\, b_j$;
  if $B > \Sigma_{i\in M}\, u_i$ **then** PTP is infeasible
  else
  begin
    initialize the incumbent $(\boldsymbol{x}^0; \boldsymbol{y}^0) := (\mathbf{0}, \mathbf{0})$ and $z^0 := +\infty$;
    BRANCH/BOUND(Y);
    $\boldsymbol{x}^* := \boldsymbol{x}^0$ and $z^* := z^0$
  end
end;

procedure BRANCH/BOUND(Y^k);
begin
  let $[l_i^k, u_i^k]$, $i \in M$, denote the intervals defining Y^k;
  if $\Sigma_{i\in M}\, l_i^k \leqslant B \leqslant \Sigma_{i\in M}\, u_i^k$ **then**
  begin {The first-stage bounding operation}
    let P^k denote the subproblem associated with Y^k;
    define the convex envelop $\bar{f}_i$ of f_i over $[l_i^k, u_i^k]$ for each $i \in M$ and construct the linear programming relaxed problem $\bar{\mathrm{P}}$ of P^k;
    solve $\bar{\mathrm{P}}$ to obtain its optimal solution $\bar{\boldsymbol{x}}$, value $\bar{z}$ and dual solution $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}})$;
    if $\bar{z} < z^0$ **then**
    begin {The second-stage bounding operation}
      $\bar{y}_i := \Sigma_{j\in N}\, x_{ij}$ for each $i \in M$;
      if $f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) < z^0$ **then** update $(\boldsymbol{x}^0, \boldsymbol{y}^0) := (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ and $z^0 := f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$;
      construct the Lagrangian relaxed problem $\mathrm{L}(\bar{\boldsymbol{\lambda}})$ of P^k using $\bar{\boldsymbol{\lambda}}$;
      solve $\mathrm{L}(\bar{\boldsymbol{\lambda}})$ to obtain its optimal value $z(\bar{\boldsymbol{\lambda}})$;
      if $z(\bar{\boldsymbol{\lambda}}) < z^0$ **then**
      begin {The branching operation}
        choose $s \in \arg\max_{i\in M} \{f_i(\bar{y}_i) - \bar{f}_i(\bar{y}_i)\}$;
        divide $I_s^k = [l_s^k, u_s^k]$ into $I_s^{k_1} = [l_s^k, \bar{y}_s]$ and $I_s^{k_2} = [\bar{y}_s, u_s^k]$;
        for $h = 1, 2$ **do**
        begin
          $Y^{k_h} := I_1^k \times \cdots \times I_{s-1}^k \times I_s^{k_h} \times I_{s+1}^k \times \cdots \times I_m^k$;
          BRANCH/BOUND(Y^{k_h})

```
          end
        end
      end
    end
end;
```

The convex envelops $\bar{f}_i$, $i \in M$, can be obtained in $O(m)$ arithmetic operations and $O(m)$ evaluations of f_is. Problem $\bar{\mathrm{P}}$ can be constructed in $O(mn)$ arithmetic operations. Therefore, the number of arithmetic operations at the first stage of the bounding operation is dominated by that needed to solve a Hitchcock problem $\bar{\mathrm{P}}$; it can also be bounded by some lower order polynomial in (m, n), e.g. $H(m, n) = O((mn \log(m + n))(mn + (m + n) \log(m + n)))$ (see [2]). The second stage, as shown in Theorem 3.2, requires $O(mn \log n)$ arithmetic operations and $O(mn)$ evaluations of f_is. Consequently, if an evaluation of f_is can be done in a unit time, the total computational time needed in the procedure BRANCH/BOUND is bounded by $O(H(m, n))$ before its recursive calls.

4. Computational results

Let us report computational results of testing the algorithm LP_LAGRANGE on randomly generated problems of PTP.

The algorithm was coded in double precision C language according to the description in Section 3.1. In the procedure BRANCH/BOUND, the Hitchcock problem $\bar{\mathrm{P}}$ was solved by the stepping-stone algorithm; and $\mathrm{L}(\bar{\boldsymbol{\lambda}})$ was solved through sorting $c_{ij}(\bar{\boldsymbol{\lambda}})$, $j \in N$, by quicksort for each $i \in M$. Therefore, in our code, the first-stage bounding operation requires pseudo-polynomial time [2]; the second-stage requires $O(mn \log n)$ time on the average but $O(mn^2)$ time in the worst case [1]. In addition to the code LP_LAGRANGE, Soland's algorithm [20] was coded in the same way for the sake of comparison (denoted by SOLAND).

The test problems were generated in the following manner: c_{ij}s were integers drawn from the uniform distribution [1, 10]; u_is were all fixed at 200; b_js were set to the round-off value of $\alpha(\Sigma_{i \in M} u_i)/n$ for $\alpha = 0.6$, 0.75 and 0.9; and the concave production costs were defined by $f_i(y_i) = \beta\sqrt{y_i}$ for β uniformly random in [10, 20]. The size of (m, n) ranged from (5, 25) to (30, 100). For each size, ten instances were solved on a UNIX workstation (hyperSPARC, 150 MHz).

Table 1 shows the comparison of the codes LP_LAGRANGE and SOLAND on problems of six different sizes. In columns of LP_LAGRANGE, the average number of calls on the procedure BRANCH/BOUND and the average CPU time in seconds (and their maxima in the brackets) are listed for $\alpha = 0.6$, 0.75 and 0.9; in columns of SOLAND, the same statistics are listed for $\alpha = 0.75$. We see clearly that LP_LAGRANGE surpasses SOLAND in all respects. There is no doubt that this is caused by the second-stage bounding operation in LP_LAGRANGE, because it is

Table 1. Comparison of LP_LAGRANGE and SOLAND

$m \times n$	LP_LAGRANGE						SOLAND	
	$\alpha = 0.6$		$\alpha = 0.75$		$\alpha = 0.9$		$\alpha = 0.75$	
	# calls	time	# calls	time	# calls	time	# calls	time
5×25	126.5	0.207	22.0	0.077	1.2	0.042	138.0	0.187
	(239)	(0.367)	(119)	(0.183)	(3)	(0.050)	(225)	(0.317)
10×25	1244.2	3.13	42.4	0.297	1.0	0.117	8919.8	15.23
	(4439)	(8.03)	(143)	(0.783)	(1)	(0.133)	(31217)	(53.82)
15×25	30.8	0.441	8.8	0.297	1.0	0.243	416949.0	2453.93
	(139)	(1.25)	(19)	(0.433)	(1)	(0.283)	(1660667)	(8116.37)
5×50	171.4	1.647	67.6	1.04	16.8	0.603	392.2	2.31
	(243)	(2.43)	(141)	(1.50)	(41)	(1.08)	(649)	(3.78)
10×50	3033.0	71.46	169.2	6.85	7.4	1.07	78374.8	1340.97
	(13077)	(239.17)	(461)	(11.20)	(25)	(1.65)	(170401)	(3053.12)
15×50	1504.8	87.68	92.8	8.42	1.0	1.48	—	—*
	(4507)	(260.82)	(211)	(16.80)	(1)	(1.78)	(—)	(—)

* The aveage time exceeded 10 thousand seconds. When $\alpha = 0.9$, the number of calls was 88501.6 (291395); the time was 4054.14 (17093.9) seconds.

the essential difference between the two codes. More noteworthy is a decrease in the number of calls made by LP_LAGRANGE between $m = 10$ and 15 for each n and α. This tendency is indicated more clearly by Table 2.

Table 2 shows the results of LP_LAGRANGE on problems of larger sizes with α fixed at 0.75. For $n = 75$ and 100, the table gives the same statistics as Table 1 from $m = 5$ to 30. In either case, we see that the number of calls peaks at some m around 10–15 and then decreases as m increases. We can therefore expect that LP_LAGRANGE will keep its efficiency up to still larger (m, n) at least for randomly generated problems of PTP.

Table 2. Computational results of LP_LAGRANGE when $\alpha = 0.75$

m	$n = 75$				$n = 100$			
	# calls		time		# calls		time	
5	82.6	(135)	6.20	(10.38)	110.4	(227)	19.25	(30.48)
10	433.2	(885)	55.41	(115.10)	1530.6	(6539)	334.64	(1447.67)
15	711.8	(2309)	130.84	(395.43)	197.2	(515)	122.21	(273.50)
20	5.2	(21)	11.85	(17.32)	194.2	(1237)	134.98	(657.88)
25	3.0	(11)	12.76	(16.85)	71.6	(179)	90.78	(175.35)
30	4.6	(13)	16.36	(22.72)	8.2	(45)	46.33	(89.05)

5. Concluding remarks

As we have demonstrated in the preceding section, the Lagrangian relaxation $L(\bar{\boldsymbol{\lambda}})$ provides a fairly strong lower bound $z(\bar{\boldsymbol{\lambda}})$ for the value z^k of each subproblem P^k. In [6], Ghannadan et al. have also used a Lagrangian relaxation and proposed an efficient heuristic algorithm for problem PTP. They have relaxed the set of constraints $\Sigma_{j\in N} x_{ij} = y_i,\ i \in M$, instead of $\Sigma_{i\in M} x_{ij} = b_j,\ j \in N$. Very recently, Holmberg and Tuy have published a remarkably efficient branch-and-bound algorithm of Falk–Soland's type for a more general problem [10]. Their algorithm uses only linear programming relaxation but may be improved further by a well-devised Lagrangian relaxation.

Before closing the paper, we will show that the bound $z(\bar{\boldsymbol{\lambda}})$ can become still tighter. Since the total production at factories cannot be below the total demand at warehouses, any feasible solution $(\boldsymbol{x}, \boldsymbol{y})$ of the production-transportation problem (1.1) satisfies

$$\sum_{i\in M} y_i \geqslant B = \sum_{j\in N} b_j\,. \tag{5.1}$$

Similarly, if P^k has a feasible solution $(\boldsymbol{x}, \boldsymbol{y})$, it must satisfy (5.1). Therefore, the set of optimal solutions does not change even if we add the constraint (5.1) to P^k. The resulting Lagrangian relaxation with respect to $\Sigma_{i\in M} x_{ij} = b_j,\ j \in M$, is written as

$$L'(\boldsymbol{\lambda}) \left\| \begin{array}{ll} \text{minimize} & z = \phi(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda}) \\ \text{subject to} & \displaystyle\sum_{i\in M} y_i \geqslant B \\ & \displaystyle\sum_{j\in N} x_{ij} = y_i\,, \quad i \in M \\ & 0 \leqslant x_{ij} \leqslant b_j\,, \quad (i, j) \in A,\ \boldsymbol{y} \in Y^k\,. \end{array} \right.$$

The feasible set of $L'(\boldsymbol{\lambda})$ is obviously included in that of $L(\boldsymbol{\lambda})$. This, together with Theorem 2.6, leads to the following:

THEOREM 5.1. *Let* $(\boldsymbol{x}'(\boldsymbol{\lambda}), \boldsymbol{y}'(\boldsymbol{\lambda}))$ *denote an optimal solution of* $L'(\boldsymbol{\lambda})$ *and let* $z'(\boldsymbol{\lambda}) = \phi(\boldsymbol{x}'(\boldsymbol{\lambda}), \boldsymbol{y}'(\boldsymbol{\lambda}); \boldsymbol{\lambda})$. *Then*

$$\bar{z} \leqslant z(\bar{\boldsymbol{\lambda}}) \leqslant z'(\bar{\boldsymbol{\lambda}}) \leqslant z^k\,.$$

Due to the first constraint, $L'(\boldsymbol{\lambda})$ cannot be decomposed into rank-two monotonic problems like (3.1). However, we can transform it to a specially structured production-transportation problem, for which a pseudo-polynomial algorithm is available [15]. Let us make m copies of the set N, i.e. $N_i = N$ for each $i \in M$; and set $c_{0j}(\boldsymbol{\lambda})$ to zero for each $j \in \cup_{i\in M} N_i$. Introducing variables $x_{0j},\ j \in \cup_{i\in M} N_i$, and w_i, $i \in M$, we have a problem equivalent to $L'(\boldsymbol{\lambda})$:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in M \cup \{0\}} \sum_{j \in N_i} c_{ij}(\boldsymbol{\lambda}) x_{ij} + \sum_{i \in N} f_i(y_i) + \sum_{j \in N} \lambda_j b_j \\
\text{subject to} \quad & \sum_{i \in M} w_i \leqslant (m-1)B \\
& \sum_{j \in N_i} x_{0j} = w_i\,, \quad \sum_{j \in N_i} x_{ij} = y_i\,, \quad i \in M \\
& x_{0j} + x_{ij} = b_j\,, \quad j \in N_i,\ i \in M \\
& x_{0j}, x_{ij} \geqslant 0\,, \quad j \in N_i,\ i \in M \\
& y_i, w_i \geqslant 0\,, \quad i \in M\,.
\end{aligned}
\tag{5.2}
$$

It is easy to check that the value of (5.2) is equal to $z'(\boldsymbol{\lambda})$. This problem is a production-transportation problem in which the set of mn warehouses is partitioned into m subsets N_i, $i \in M$; each factory $i \in M$ is allowed to supply only warehouses in its assigned subset N_i; and factory 0 supplies warehouses in N_i for each $i \in M$ with a total of w_i units. Problems with such a special structure can be solved in $O(m^2nB)$ arithmetic operations and $O(mnB)$ evaluations of f_is if we apply an algorithm proposed in [15]. Although the algorithm is pseudo-polynomial, it will be an effective procedure in the branch-and-bound algorithm for solving PTP when n is beyond a hundred and B is relatively small. Computational experiments are now under way, the results of which will be reported elsewhere.

Acknowledgement

The authors are grateful to the anonymous reviewers for their valuable suggestions, which have considerably improved the earlier version of this paper.

References

1. Aho, A.V., Hopcroft, J.E. and Ullman, J.D. (1983), *Data Structures and Algorithms*, Addison-Wesley, M.A.
2. Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993), *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, N.J.
3. Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. (1993), *Nonlinear Programming: Theory and Algorithms*, 2nd ed. John Wiley and Sons, N.Y.
4. Falk, J.E. and Soland, R.M. (1969), An algorithm for separable nonconvex programming problems, *Management Science* 15: 550–569.
5. Gallo, G., Sandi, C. and Sodini, C. (1980), An algorithm for the min concave cost flow problem, *European Journal of Operational Research* 4: 248–255.
6. Ghannadan, S., Migdalas, A., Tuy, H. and Värbrand, P. (1994), Heuristics based on tabu search and Lagrangean relaxation for the concave production-transportation problem, *Studies in Regional and Urban Planning* 3: 127–140.
7. Guisewite, G.M. (1995), Network problems, in Horst, R. and Pardalos, P.M. (eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht.
8. Guisewite, G.M. and Pardalos, P.M. (1991), Global search algorithms for minimum concave-cost network flow problems, *Journal of Global Optimization*, 1: 309–330.

9. Guisewite, G.M. and Pardalos, P.M. (1993), A polynomial time solvable concave network flow problem, *Networks* 23: 143–149.
10. Holmberg, K. and Tuy, H. (1999), A production-transportation problem with stochastic demand and concave production costs, *Mathematical Programming* 85: 157–179.
11. Horst, R., Pardalos, P.M. and Thoai, N.V. (1995), *Introduction to Global Optimization*, Kluwer Academic Publishers, Dordrecht.
12. Horst, R. and Tuy, H. (1993), *Global Optimization: Deterministic Approaches*, 2nd edn. Springer-Verlag, Berlin.
13. Klinz, B. and Tuy, H. (1993), Minimum concave-cost network flow problems with a single nonlinear arc cost, in Dungzhu Du and Pardalos, P.M. (eds.), *Network Optimization Problems*, World Scientific, Singapore, pp. 125–143.
14. Konno, H., Thach, P.T. and Tuy, H. (1997), *Optimization on Low Rank Nonconvex Structures*, Kluwer Academic Publishers, Dordrecht.
15. Kuno, T. and Utsunomiya, T. (1996), A decomposition algorithm for solving certain classes of production-transportation problems with concave production cost, *Journal of Global Optimization* 8: 67–80.
16. Kuno, T. and Utsunomiya, T. (1997), A pseudo-polynomial primal-dual algorithm for globally solving a production-transportation problem, *Journal of Global Optimization* 11: 163–180.
17. Kuno, T. (1997), A pseudo-polynomial algorithm for solving rank three concave production-transportation problems, *Acta Mathematica Vietnamica* 22: 159–182.
18. Magnanti, T.L. and Wong, R.T. (1984), Network design and transportation planning: models and algorithms, *Transportation Science* 18: 1–55.
19. Nemhauser, G.L. and Wolsey, L.A. (1988), *Integer and Combinatorial Optimization*, John Wiley and Sons, N.Y.
20. Soland, R.M. (1974), Optimal facility location problems with concave costs, *Operations Research* 22: 373–382.
21. Tuy, H. (1991), Polyhedral annexation, dualization and dimension reduction technique in global optimization, *Journal of Global Optimization* 1: 229–244.
22. Tuy, H. (1998), *Convex Analysis and Global Optimization*, Kluwer Academic Publishers, Dordrecht.
23. Tuy, H., Dan, N.D. and Ghannadan, S. (1993), Strongly polynomial time algorithms for certain concave minimization problems on networks, *Operations Research Letters* 14: 99–109.
24. Tuy, H., Ghannadan, S., Migdalas, A. and Värbrand, P. (1993), Strongly polynomial algorithm for a production-transportation problem with concave production cost, *Optimization* 27: 205–227.
25. Tuy, H., Ghannadan, S., Migdalas, A. and Värbrand, P. (1996), Strongly polynomial algorithm for a production-transportation problem with a fixed number of nonlinear variables, *Mathematical Programming* 72: 229–258.