Journal of Global Optimization **8**: 67–80, 1996.

A Decomposition Algorithm for Solving Certain Classes of Production-Transportation Problems with Concave Production Cost

TAKAHITO KUNO* and TAKAHIRO UTSUNOMIYA
Institute of Information Sciences and Electronics, University of Tsukuba, 1-1-1 Tennoh-dai, Tsukuba, 305 Ibaraki, Japan

(Received: 20 September 1994; accepted: 26 June 1995)

Abstract. This paper addresses a method for solving two classes of production-transportation problems with concave production cost. By exploiting a special network structure both problems are reduced to a kind of resource allocation problem. It is shown that the resultant problem can be solved by using dynamic programming in time polynomial in the number of supply and demand points and the total demand.

Key words: Concave minimization, global optimization, production-transportation problem, resource allocation problem, dynamic programming.

1. Introduction

In this paper we will discuss special classes of production-transportation problems which arise in many practical applications, for instance:

Suppose a corporation has one factory and a number of warehouses in each of several regions. Every factory produces a certain amount of goods, and can transport them only to warehouses in its assigned region. In addition to these branch factories, there is a head factory which can transport the product to every warehouse. This corporation has to decide how much goods each factory should produce, and which warehouses the head factory should supply, so as to minimize the total production and transportation cost.

In the above situation (see also Figure 1), we are concerned with two cases:

(P1): The production cost of the head factory need not be considered but its production capacity is restricted.

(P2): The production capacity of the head factory is not restricted but its production cost has to be considered.

* The author was partially supported by Grand-in-Aid for Scientific Research of the Ministry of Education, Science and Culture, Grant No. (C)05650061.

The production cost of each factory is in general a nondecreasing and concave function of the output, whence both the problems (P1) and (P2) have multiple locally optimal solutions, many of which need not be globally optimal.

In their recent series of articles [9, 10, 11], Tuy *et al.* have proposed a strongly polynomial algorithm for solving a production-transportation problem similar to (P2), where each factory is allowed to supply any warehouses but the number of factories is assumed to be a constant. The cost function of their problem possesses rank-k property [8], where k is the number of factories, and its global minimum can be found in the course of solving a transportation problem parametrically. In this paper, without assuming the fixed number of branch factories, we will show that both (P1) and (P2) can be solved in time polynomial in the number of factories and warehouses and the total demand of warehouses.

The organization of the paper is as follows: In Section 2, we will transform the problem (P1) into a kind of resource allocation problem, referred to as the *master problem* of (P1), by exploiting the special network structure stated above. Its objective function is defined by solving m Hitchcock transportation problems, where m represents the number of branch factories. In Section 3, to solve the master problem we will propose an algorithm using dynamic programming, and show that it requires $O(mnb)$ arithmetic operations and $O(nb)$ evaluations of the production cost function of each factory, where n and b are the number of warehouses and the production capacity of the head factory respectively. In Section 4, we will show that the problem (P2) can also be transformed into a resource allocation problem. The number of arithmetic operations needed for solving the resultant problem is $O(mnd)$, where d is the total demand of warehouses.

2. Decomposition of (P1) into Subproblems

The problem we first consider is formulated below:

$$\left|\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=0}^{m}\sum_{j\in V_i} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i(z_i) & \\ \text{subject to} & \displaystyle\sum_{i=1}^{m} y_i \leqslant b, & \\ & \displaystyle\sum_{j\in V_i} x_{0j} = y_i, \quad \sum_{j\in V_i} x_{ij} = z_i, & i = 1,\ldots,m, \\ & x_{0j} + x_{ij} = d_j, \quad j \in V_i, & i = 1,\ldots,m, \\ & x_{0j} \geqslant 0, \quad x_{ij} \geqslant 0, \quad j \in V_i, & i = 1,\ldots,m, \\ & y_i \geqslant 0, \quad z_i \geqslant 0, & i = 1,\ldots,m, \end{array}\right. \tag{2.1}$$

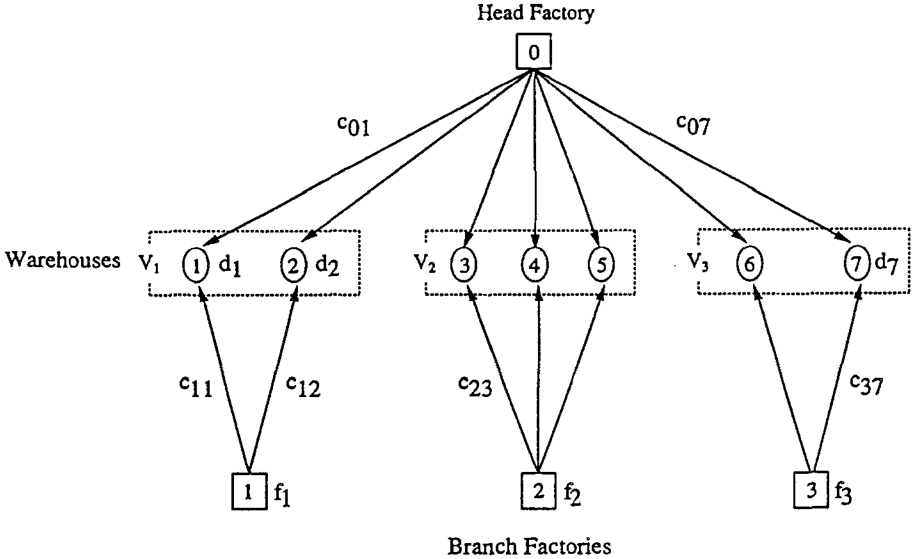

Fig. 1. Example of the problem.

where $b, d_j > 0, j \in V_i, i = 1, \ldots, m$, are integral, $c_{ij} \geqslant 0, j \in V_i, i = 0, 1, \ldots, m$, are real, $f_i : \mathbb{R}^1 \to \mathbb{R}^1$, $i = 1, \ldots, m$, are nondecreasing and concave functions, and V_i, $i = 0, 1, \ldots, m$, are index sets such that

$$V_s \bigcap V_t = \emptyset, \quad s \neq t, \quad s, t = 1, \ldots, m; \quad \bigcup_{i=1}^{m} V_i = V_0 = \{1, \ldots, n\}. \tag{2.2}$$

Using the example in Section 1, we can illustrate (2.1) as follows: For $i = 1, \ldots, m$, warehouse j in region V_i requires d_j units of the product and receives x_{0j} and x_{ij} units from factories 0 and i respectively. Factory 0, which represents the head factory, produces at most b units and supplies warehouses in V_i with y_i units. Factory i produces z_i units at a cost of $f_i(z_i)$. The decision maker of the corporation has to determine x_{ij}, $j \in V_i$, $i = 0, 1, \ldots, m$, y_i and z_i, $i = 1, \ldots, m$, which minimizes the objective function of (2.1) expressing the total cost. Figure 1 shows a network of the problem when $m = 3$ and $n = 7$.

A special case of (2.1), where $m = 1$, involves the problem studied by Tuy *et al.* in [9], and can be solved in $O(n \log n)$ arithmetic operations and n evaluations of function f_1 if we use the algorithm proposed in [9].

Any feasible solution of (2.1) has to satisfy

$$z_i = a_i - y_i, \quad i = 1, \ldots, m, \tag{2.3}$$

where $a_i = \sum_{j \in V_i} d_j$. We can therefore eliminate all z_i's from (2.1) by defining

$$\bar{f}_i(y_i) = f_i(a_i - y_i), \quad i = 1, \ldots, m. \tag{2.4}$$

Obviously $\bar{f}_i$'s are still concave but nonincreasing. The first problem is then as follows:

$$
(\mathrm{P1})\quad
\begin{array}{|lll}
\text{minimize} & \displaystyle\sum_{i=0}^{m}\sum_{j\in V_i} c_{ij}x_{ij} + \sum_{i=1}^{m} \bar{f}_i(y_i) & \\
\text{subject to} & \displaystyle\sum_{i=1}^{m} y_i \leqslant b, & \\
& \displaystyle\sum_{j\in V_i} x_{0j} = y_i, \quad \sum_{j\in V_i} x_{ij} = a_i - y_i, & i = 1,\dots,m, \\
& x_{0j} + x_{ij} = d_j, \quad j \in V_i, & i = 1,\dots,m, \\
& x_{0j} \geqslant 0, \quad x_{ij} \geqslant 0, \quad j \in V_i, & i = 1,\dots,m, \\
& y_i \geqslant 0, & i = 1,\dots,m.
\end{array}
\qquad (2.5)
$$

2.1. Definition of Master Problem

For any fixed $\mathbf{y} = (y_1, \dots, y_m)$, we have a linear programming problem:

$$
(\mathrm{P}(\mathbf{y}))\quad
\begin{array}{|lll}
\text{minimize} & \displaystyle\sum_{i=0}^{m}\sum_{j\in V_i} c_{ij}x_{ij} & \\
\text{subject to} & \displaystyle\sum_{j\in V_i} x_{0j} = y_i, \quad \sum_{j\in V_i} x_{ij} = a_i - y_i, & i = 1,\dots,m, \\
& x_{0j} + x_{ij} = d_j, \quad j \in V_i, & i = 1,\dots,m, \\
& x_{0j} \geqslant 0, \quad x_{ij} \geqslant 0, \quad j \in V_i, & i = 1,\dots,m.
\end{array}
\qquad (2.6)
$$

Due to the condition (2.2), problem $(\mathrm{P}(\mathbf{y}))$ can be decomposed into m subproblems $(\mathrm{P}_i(y_i))$, $i = 1, \dots, m$, each of which is a Hitchcock transportation problem with two supply points:

$$
(\mathrm{P}_i(y_i))\quad
\begin{array}{|ll}
\text{minimize} & \displaystyle\sum_{j\in V_i} (c_{0j}x_{0j} + c_{ij}x_{ij}) \\
\text{subject to} & \displaystyle\sum_{j\in V_i} x_{0j} = y_i, \quad \sum_{j\in V_i} x_{ij} = a_i - y_i, \\
& x_{0j} + x_{ij} = d_j, \qquad j \in V_i, \\
& x_{0j} \geqslant 0, \quad x_{ij} \geqslant 0, \quad j \in V_i.
\end{array}
\qquad (2.7)
$$

If $0 \leqslant y_i \leqslant a_i$, then $(\mathrm{P}_i(y_i))$ has an optimal solution. We denote it by a vector $\mathbf{x}_i^*(y_i)$, whose components are $x_{0j}^*(y_i)$, $x_{ij}^*(y_i)$, $j \in V_i$, and by $g_i(y_i)$ its optimal value. Obviously $\mathbf{x}^*(\mathbf{y}) = (\mathbf{x}_1^*(y_1), \dots, \mathbf{x}_m^*(y_m))$ is an optimal solution of $(\mathrm{P}(\mathbf{y}))$ and $\sum_{i=1}^{m} g_i(y_i)$ is the optimal value. The original problem (P1) can be solved if we solve $(\mathrm{P}(\mathbf{y}))$ for all $\mathbf{y}$ satisfying $\sum_{i=1}^{m} y_i \leqslant b$ and $0 \leqslant y_i \leqslant a_i$ for every i. Let

$$
h_i(y_i) = \bar{f}_i(y_i) + g_i(y_i), \quad i = 1, \dots, m. \qquad (2.8)
$$

Then (P1) is reduced to a kind of resource allocation problem with m variables:

$$(\text{MP1})\quad \left|\begin{array}{l} \text{minimize } \sum_{i=1}^{m} h_i(y_i) \\ \text{subject to } \sum_{i=1}^{m} y_i \leqslant b, \\ \qquad 0 \leqslant y_i \leqslant a_i, \quad i = 1, \ldots, m, \end{array}\right. \tag{2.9}$$

which we call the *master problem* of (P1). Without loss of generality we may assume that

$$b \leqslant \sum_{i=1}^{m} a_i \left(= \sum_{j=1}^{n} d_j \right). \tag{2.10}$$

The following lemma summarizes the above arguments:

LEMMA 2.1. *If* $\mathbf{y}^*$ *is an optimal solution of (MP1), then* $(\mathbf{x}^*(\mathbf{y}^*), \mathbf{y}^*)$ *solves (P1), where* $\mathbf{x}^*(\mathbf{y}^*) = (\mathbf{x}_1^*(y_1^*), \ldots, \mathbf{x}_m^*(y_m^*))$ *and* $\mathbf{x}_i^*(y_i^*)$ *is an optimal solution of* $(P_i(y_i^*))$. □

2.2. ANALYTIC FORM OF FUNCTION h_i

To solve (MP1) we have to know the analytic form of function h_i, which is a composition of two functions $\bar{f}_i$ and g_i. While the former is given beforehand, the latter requires solving the Hitchcock transportation problem $(\mathrm{P}_i(y_i))$ as varying the value of y_i in the interval $[0, a_i]$.

Note that the constraint $\sum_{j \in V_i} x_{ij} = a_i - y_i$ is implied by the others and hence can be deleted from the definition of $(\mathrm{P}_i(y_i))$, i.e.,

$$\left|\begin{array}{ll} \text{minimize} & \sum_{j \in V_i} (c_{0j} x_{0j} + c_{ij} x_{ij}) \\ \text{subject to} & \sum_{j \in V_i} x_{0j} = y_i, \\ & x_{0j} + x_{ij} = d_j, \qquad j \in V_i, \\ & x_{0j} \geqslant 0, x_{ij} \geqslant 0, \quad j \in V_i. \end{array}\right. \tag{2.11}$$

We should also note that any feasible solution satisfies

$$x_{ij} = d_j - x_{0j}, \quad \forall j \in V_i. \tag{2.12}$$

Then, by substituting (2.12) into (2.11), we have an equivalent problem with $|V_i|$ variables:

$$(\mathrm{Q}_i(y_i))\quad \left|\begin{array}{ll} \text{minimize} & \sum_{j\in V_i} \bar{c}_j x_{0j} + \sum_{j\in V_i} c_{ij} d_j \\ \text{subject to} & \sum_{j\in V_i} x_{0j} = y_i, \\ & 0 \leqslant x_{0j} \leqslant d_j, \quad j \in V_i. \end{array}\right. \tag{2.13}$$

where $\bar{c}_j = c_{0j} - c_{ij}$, $j \in V_i$. This is a continuous knapsack problem. If $0 \leqslant y_i \leqslant a_i$, it is well known (see, e.g. [1]) that the optimal value of $(\mathrm{Q}_i(y_i))$ is given by

$$g_i(y_i) = \sum_{l=1}^{p-1} \bar{c}_{j_l} d_{j_l} + \bar{c}_p \left(y_i - \sum_{l=1}^{p-1} d_{j_l} \right) + \sum_{j\in V_i} c_{ij} d_j \tag{2.14}$$

for some p such that $\sum_{l=1}^{p-1} d_{j_l} \leqslant y_i < \sum_{l=1}^{p} d_{j_l}$, where

$$\bar{c}_{j_1} \leqslant \bar{c}_{j_2} \leqslant \cdots \leqslant \bar{c}_{j_{|V_i|}}. \tag{2.15}$$

Let

$$a_{i0} = 0, \quad a_{ik} = \sum_{l=1}^{k} d_{j_l}, \quad k = 1, \ldots, |V_i|, \tag{2.16}$$

and let

$$I_{ik} = [a_{i,k-1}, a_{ik}], \quad k = 1, \ldots, |V_i|. \tag{2.17}$$

The analytic form of h_i is then identified by the following:

LEMMA 2.2. *Function h_i is concave on I_{ik} for every $k = 1, \ldots, |V_i|$.*

Proof. We immediately see from (2.14) that g_i is a convex and piecewise affine function with break points among a_{ik}, $k = 0, 1, \ldots, |V_i|$. Hence $h_i = \bar{f}_i + g_i$ is concave on each linear piece $I_{ik} = [a_{i,k-1}, a_{ik}]$ of g_i, because $\bar{f}_i$ is a concave function defined on $\mathbb{R}$. □

In [9] Tuy *et al.* have derived the same result as Lemma 2.2. They have straightforward used the network structure of $(\mathrm{P}_i(y_i))$ instead of transforming it into the continuous knapsack problem.

3. Solution Method for the Master Problem (MP1)

Let us proceed to the algorithm for solving the master problem:

$$(\mathrm{MP1})\quad \left|\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} h_i(y_i) \\ \text{subject to} & \sum_{i=1}^{m} y_i \leqslant b, \\ & 0 \leqslant y_i \leqslant a_i, \quad i = 1, \ldots, m. \end{array}\right.$$

We will show that (MP1) can be solved using dynamic programming. For this purpose let us observe some properties of its optimal solutions.

LEMMA 3.1. *Problem (MP1) has an optimal solution* $\mathbf{y}^* = (y_1^*, \ldots, y_m^*)$, *at least* $m - 1$ *components of which are elements of the set* $\{a_{ik} | k = 0, \ldots, |V_i|, i = 1, \ldots, m\}$.

Proof. Since b and all a_i's are positive, the feasible region of (MP1) is nonempty and bounded. Every h_i is continuous on $[0, a_i]$, and hence the objective function of (MP1) attains the minimum at some $\mathbf{y}^*$ in the feasible region. Suppose there are two components of $\mathbf{y}^*$, say y_p^* and y_q^*, which are not in a_{ik}'s. Let $y_p^* \in \text{int}\, I_{ps} = (a_{p,s-1}, a_{ps})$, $y_q^* \in \text{int}\, I_{qt} = (a_{q,t-1}, a_{qt})$, and let

$$h_{pq}(y) = h_p(y) + h_q(\beta - y),$$

where $\beta = y_p^* + y_q^*$. Also let

$$\underline{\alpha} = \max\{a_{p,s-1}, \beta - a_{qt}\}, \quad \overline{\alpha} = \min\{a_{ps}, \beta - a_{q,t-1}\}.$$

Then $y_p^* \in (\underline{\alpha}, \overline{\alpha})$ and h_{pq} is concave on $[\underline{\alpha}, \overline{\alpha}]$. Hence we have

$$h_{pq}(y_p^*) \geqslant \min\{h_{pq}(\underline{\alpha}), h_{pq}(\overline{\alpha})\},$$

which implies that if we replace y_p^*, y_q^* by either $\underline{\alpha}$, $\beta - \underline{\alpha}$ or $\overline{\alpha}$, $\beta - \overline{\alpha}$ then another optimal solution $\mathbf{y}'$ of (MP1) is provided. In this case, either y_p' or y_q' coincides with an extreme point of its interval. □

Consider m discrete optimization problems $(\text{DP}_i(y_i))$, $i = 1, \ldots, m$, associated with (MP1):

$$(\text{DP}_i(y_i)) \quad \left| \begin{array}{ll} \text{minimize} & \sum_{l \neq i} h_l(y_l) \\ \text{subject to} & \sum_{l \neq i} y_l \leqslant b - y_i, \\ & y_l \in \{a_{lk} | k = 0, 1, \ldots, |V_l|\}, \quad l \neq i. \end{array} \right. \tag{3.1}$$

We denote by $H_i(y_i)$ the optimal value of $(\text{DP}_i(y_i))$. It follows from Lemma 3.1 that an optimal solution $\mathbf{y}^*$ of (MP1) is found if we solve every $(\text{DP}_i(y_i))$ for $y_i \in [0, a_i]$. Namely,

$$\min\{\min\{h_i(y_i) + H_i(y_i) | y_i \in [0, a_i]\} | i = 1, \ldots, m\} \tag{3.2}$$

is the minimum value of the objective function of (MP1).

LEMMA 3.2. *For each* i *there exists an integer* $y_i' \in [0, a_i]$ *such that*

$$h_i(y_i') + H_i(y_i') = \min\{h_i(y_i) + H_i(y_i) | y_i \in [0, a_i]\}. \tag{3.3}$$

Proof. Let $y_i' \in I_{is}$ and suppose y_i' is not integral. Since b and all a_{lk}'s are integral, it must hold that

$$h_i(y_i') + H_i(\lceil y_i' \rceil) \geqslant H_i(\lfloor y_i' \rfloor),$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ represent the integers obtained by rounding up and down · respectively. Hence we have

$$h_i(y_i') + H_i(y_i') \geqslant \min\{h_i(\lceil y_i' \rceil) + H_i(\lceil y_i' \rceil), h_i(\lfloor y_i' \rfloor) + H_i(\lfloor y_i' \rfloor)\}$$

by noting the concavity of h_i on $[\lfloor y_i' \rfloor, \lceil y_i' \rceil] \subset I_{is}$. □

Thus (3.2) turns out to be

$$\min\{\min\{h_i(y_i) + H_i(y_i) | y_i = 0, 1, \ldots, a_i\} | i = 1, \ldots, m\}. \tag{3.4}$$

3.1. Dynamic Programming Recursion

Let us define a partial problem of $(\mathrm{DP}_i(y_i))$:

$$(\mathrm{DP}_i^q(y_i)) \quad \left| \begin{array}{ll} \text{minimize} & \sum_{l \in M(i,q)} h_l(y_l) \\ \text{subject to} & \sum_{l \in M(i,q)} y_l \leqslant b - y_i, \\ & y_l \in \{a_{lk} | k = 0, 1, \ldots, |V_l|\}, \quad l \in M(i, q), \end{array} \right. \tag{3.5}$$

where $M(i, q) = \{1, \ldots, i-1, i+1, \ldots, q\}$. Denote by $H_i^q(y_i)$ the optimal value of $(\mathrm{DP}_i^q(y_i))$ and let

$$H_i^q(y_i) = \begin{cases} 0 & \text{if } y_i \leqslant b, q = 0, \\ +\infty & \text{if } y_1 > b, q = 0 \text{ or } y_i \geqslant b, i \neq q > 0, \\ H_i^{q-1}(y_i) & \text{if } i = q. \end{cases} \tag{3.6}$$

LEMMA 3.3. *The values $H_i^q(y_i)$'s satisfy the following recursive formula:*

$$H_i^q(y_i) = \min\{h_q(a_{qk}) + H_i^{q-1}(y_i + a_{qk}) | k = 0, 1, \ldots, |V_q|\}. \tag{3.7}$$

Proof. By definition we have

$$\begin{aligned} H_i^q(y_i) &= \min\{h_q(y_q) + H_i^{q-1}(y_i + y_q) | y_q \in \{a_{qk} | k = 0, 1, \ldots, |V_q|\}\} \\ &= \min\{h_q(a_{qk}) + H_i^{q-1}(y_i + a_{qk}) | k = 0, 1, \ldots, |V_q|\} \end{aligned}$$ □

Since $H_i(y_i) = H_i^m(y_i)$, to obtain $H_i(y_i)$ for all $y_i \in [0, a_i]$ we need only to compute $H_i^q(y_i)$ for $q = 1, \ldots, i-1, i+1, \ldots, m$ and $y_i = b, b-1, \ldots, 1, 0$.

Note that ($\mathrm{DP}_i(y_i)$) can be transformed into a multiple-choice knapsack problem [6]. Through such a transformation, we will obtain a recursive formula like (3.7) (see [2, 6] for further details).

We are now ready to present the algorithm for solving the target problem (P1):

ALGORITHM A.

Step 1. For $i = 1, \dots, m$ do the following:

1° Compute $\bar{c}_j = c_{0j} - c_{ij}$, $j \in V_i$, and sort them as $\bar{c}_{j_1} \leqslant \bar{c}_{j_2} \leqslant \cdots \bar{c}_{j_{|V_i|}}$.

2° Let $a_{i0} = 0$, $a_{ik} = \sum_{l=1}^{k} d_{j_l}$, $k = 1, \dots, |V_i|$.

Step 2. For $i = 1, \dots, m$ do the following:

1° Compute $H_i^q(y_i)$ according to (3.6) and (3.7) in the order $q = 1, \dots, i-1, i+1, \dots, m$; $y_i = b, b-1, \dots, 1, 0$.

2° Let

$$y_i' = \operatorname{argmin}\{h_i(y_i) + H_i^m(y_i) | y_i = 0, 1, \dots, a_i\} \tag{3.8}$$

and let $v_i = h(y_i') + H_i^m(y_i')$.

Step 3. Let

$$v_r = \min\{v_1, v_2, \dots, v_m\},$$

and let $y_r^* = y_r'$. Also let $y_i^*, i \in M(r, m)$, be an optimal solution of ($\mathrm{DP}_r(y_r')$). Then an optimal solution $\mathbf{x}^*(\mathbf{y}^*)$ of ($\mathrm{P}(\mathbf{y}^*)$) is optimal to (P1). □

THEOREM 3.4. *Algorithm A requires $O(mnb)$ arithmetic operations and $O(nb)$ evaluations of f_i for each $i = 1, \dots, m$.*

Proof. To sort $\bar{c}_j$'s Step 1 requires $O(n \log n)$ arithmetic operations. If $\bar{c}_j$'s are sorted, then for any $\mathbf{y}^*$ we will have $\mathbf{x}^*(\mathbf{y}^*)$ in time $O(\log n)$ using binary search. The total computational time of the algorithm is therefore dominated by Step 2.1°. It takes $2|V_q|$ additions, $|V_q| - 1$ comparisons and $|V_q|$ evaluations of f_i to compute $H_i^q(y_i)$. For each i these numbers are bounded by

$$\sum_{y_i=0}^{b} \sum_{q \in M(i,m)} O(|V_q|) = O(nb).$$

Hence the total number of arithmetic operations is $\sum_{i=1}^{m} O(nb) = O(nmb)$. □

In general, Algorithm A does not run in time polynomial in the problem input length, even though the values of f_i are provided by an oracle. However, when all d_j's have a common value, say δ, the number of arithmetic operations is a polynomial function of only m and n. In this case, the value of b is bounded by $\delta \sum_{i=1}^{m} |V_i| = \delta n$ under the assumption (2.10), and hence the total number of arithmetic operations becomes $O(mnb) = O(mn^2)$.

REMARK. We have assumed that all f_i's are continuous and nondecreasing. Although f_i's are certainly nondecreasing in many applications, it is inessential to Algorithm A. In contrast to this, we could not prove any lemmas in Section 3 without assuming the continuity of f_i's. However, one might reasonably expect f_i's to be piecewise concave and discontinuous (e.g. fixed-charge cost functions). If f_i's are lower semi-continuous, Algorithm A can be modified in order to handle discontinuous f_i's.

If we divide the intervals $I_{ik} = [a_{i,k-1}, a_{ik}]$'s of (2.17) further at discontinuous points of $\bar{f}_i$, then m_i intervals $\tilde{I}_{ik} = [\tilde{a}_{i,k-1}, \tilde{a}_{ik}]$, $k = 1, \ldots, m_i$, are generated, where $m_i \geqslant |V_i|$ and $\tilde{a}_{ik}$ is either some $a_{ik'}$ or a discontinuous point of $\bar{f}_i$. Since f_i is concave on continuous pieces, h_i is also concave on the interior of each $\tilde{I}_{ik}$. Moreover, h_i attains the minimum at some extreme points of $\tilde{I}_{ik}$'s by the lower semi-continuity. These can prove all the lemmas in Section 3 if we replace a_{ik}'s by $\tilde{a}_{ik}$'s. The modified algorithm is polynomial in m, n, b and the number of continuous pieces of f_i's. □

3.2. Numerical Example

Before concluding this section, let us illustrate Algorithm A using a simple example of (P1) with $m = 3$, $n = 7$ and $b = 10$, whose network is given by Figure 1. Coefficients of the problem are

$$(c_{0j}) = (1, 7, 5, 6, 3, 2, 7), \qquad (c_{1j}) = (2, 8, \infty, \infty, \infty, \infty, \infty),$$
$$(c_{2j}) = (\infty, \infty, 7, 6, 1, \infty, \infty), \quad (c_{3j}) = (\infty, \infty, \infty, \infty, \infty, 5, 8),$$
$$(d_j) = (2, 5, 6, 3, 2, 7, 4).$$

and the production cost of factory i ($\neq 0$) is

$$f_i(z_i) = \alpha_i \cdot z_i^{\beta_i},$$

where

$$(\alpha_i) = (2, 3, 5), \qquad (\beta_i) = (0.8, 0.3, 0.2).$$

To solve the problem, we first compute

$$(\bar{c}_j) = (-1, -1, -2, 0, 2, -3, -1),$$

$$(a_{1k}) = (0, 2, 7), \quad (a_{2k}) = (0, 6, 9, 11), \quad (a_{3k}) = (0, 7, 11),$$

and sort $\bar{c}_j$'s as follows:

$$\bar{c}_1 \leqslant \bar{c}_2, \quad \bar{c}_3 \leqslant \bar{c}_4 \leqslant \bar{c}_5, \quad \bar{c}_6 \leqslant \bar{c}_7.$$

Next, we compute $H_i^q(y_i)$ for each i in Step 2. For example, $H_1^2(10)$ is given by

$$\min\begin{Bmatrix} h_2(a_{20}) + H_1^1(10+a_{20}), \\ h_2(a_{21}) + H_1^1(10+a_{21}), \\ h_2(a_{22}) + H_1^1(10+a_{22}), \\ h_2(a_{23}) + H_1^1(10+a_{23}), \end{Bmatrix} = \min\begin{Bmatrix} h_2(0) + H_1^1(10), \\ h_2(9) + H_1^1(16), \\ h_2(9) + H_1^1(19), \\ h_2(11) + H_1^1(21), \end{Bmatrix}.$$

It follows from (3.6) that

$$H_1^1(10) = H_1^0(10) = 0, \quad H_1^1(16) = H_1^1(19) = H_1^1(21) = +\infty,$$

and from (2.4) and (2.8) that

$$h_2(0) = f_2(11-0) + g_2(0) = 6.159 + 62 = 68.159.$$

Hence we have

$$H_1^2(10) = h_2(0) + H_1^1(10) = 68.159.$$

Similarly,

$$H_1^2(9) = H_1^2(8) = H_1^2(7) = H_1^2(6) = H_1^2(5) = 68.159,$$

$$H_1^2(4) = H_1^2(3) = H_1^2(2) = 54.862, \quad H_1^2(1) = H_1^2(0) = 53.693.$$

Substituting these values into the recursive formula (3.7), we can compute the values of H_1^3, i.e.,

$$H_1^3(10) = H_1^3(9) = H_1^3(8) = H_1^3(7) = H_1^3(6) = H_1^3(5) = 143.236,$$

$$H_1^3(4) = 129.939, \quad H_1^3(3) = H_1^3(2) = H_1^3(1) = H_1^3(0) = 120.757.$$

On the other hand, the values of h_1 are

$$h_1(10) = h_1(9) = h_1(8) = h_1(7) = 37.000,$$

$$h_1(6) = 40.000, \quad h_1(5) = 40.482, \quad h_1(4) = 44.816, \quad h_1(3) = 47.063,$$

$$h_1(2) = 49.248, \quad h_1(1) = 51.386, \quad h_1(0) = 53.487.$$

Thus, for $i = 1$, we obtain

$$y_1' = 3 = \operatorname{argmin}\{h_1(y_i) + H_i^3(y_i) | y_1 = 0, 1, \ldots, 7\},$$

$$v_1 = h_1(3) + H_1^3(3) = 167.820.$$

For $i = 2, 3$, we have the following in the same manner:

$$y_2' = 3, \quad v_2 = 167.682,$$

$$y_3' = 7, \quad v_3 = 168.636.$$

Finally, we obtain

$$\mathbf{y}^* = (0,3,7), \quad \mathbf{x}_1^*(\mathbf{y}^*) = (2,5), \quad \mathbf{x}_2^*(\mathbf{y}^*) = (3,3,2), \quad \mathbf{x}_3^*(\mathbf{y}^*) = (0,4),$$

and the globally optimal value 167.682 in Step 3.

4. Application of the Algorithm to (P2) and Other Problems

The second problem is as follows:

$$(\text{P2}) \quad \left| \begin{array}{lll} \text{minimize} & \sum_{i=0}^{m} \sum_{j \in V_i} c_{ij} x_{ij} + \sum_{i=1}^{m} \bar{f}_i(y_i) + f_0(z_0) & \\ \text{subject to} & \sum_{i=1}^{m} y_i = z_0, & \\ & \sum_{j \in V_i} x_{0j} = y_i, \quad \sum_{j \in V_i} x_{ij} = a_i - y_i, & i = 1, \ldots, m, \\ & x_{0j} + x_{ij} = d_j, \quad j \in V_i, & i = 1, \ldots, m, \\ & x_{0j} \geqslant 0, \quad x_{ij} \geqslant 0, \quad j \in V_i, & i = 1, \ldots, m, \\ & z_0 \geqslant 0, \quad y_i \geqslant 0, & i = 1, \ldots, m, \end{array} \right. \qquad (4.1)$$

where f_0 is a nondecreasing and concave function of z_0, and all of the other notations are the same as (P1). As before, we can define the master problem of (P2):

$$(\text{MP2}) \quad \left| \begin{array}{l} \text{minimize} \ \sum_{i=1}^{m} h_i(y_i) + f_0(z_0) \\ \text{subject to} \ \sum_{i=1}^{m} y_i = z_0, \\ \qquad z_0 \geqslant 0, \quad 0 \leqslant y_i \leqslant a_i, \quad i = 1, \ldots, m, \end{array} \right. \qquad (4.2)$$

where $a_i = \sum_{j \in V_i} d_j$, $h_i(y_i) = \bar{f}_i(y_i) + g_i(y_i)$, and $g_i(y_i)$ is the optimal value of the Hitchcock transportation problem $(\text{P}_i(y_i))$. If we obtain an optimal solution $(\mathbf{y}^*, z_0^*)$ of (MP2), then $(\mathbf{x}^*(\mathbf{y}^*), \mathbf{y}^*, z_0^*)$ solves (P2), where $\mathbf{x}^*(\mathbf{y}^*)$ is an optimal solution of $(\text{P}(\mathbf{y}^*))$ defined by (2.6).

Let $d = \sum_{i=1}^{m} a_i$ and let

$$y_0 = d - z_0. \qquad (4.3)$$

For any feasible solution of (MP2) we have $0 \leqslant y_0 \leqslant d$, since $0 \leqslant z_0 \leqslant d$ must hold. Also let

$$h_0(y_0) = f_0(d - y_0). \tag{4.4}$$

Then h_0 is a concave function on $I_{01} = [0, d]$, and (MP2) is rewritten as

$$\left|\begin{array}{l} \text{minimize } \sum_{i=0}^{m} h_i(y_i) \\ \text{subject to } \sum_{i=0}^{m} y_i = d, \\ \qquad 0 \leqslant y_0 \leqslant d, \quad 0 \leqslant y_i \leqslant a_i, \quad i = 1, \ldots, m, \end{array}\right. \tag{4.5}$$

which is of just the same form as (MP1). We can again apply dynamic programming to (4.5). Then Algorithm A will generate an optimal solution of (P2) in $O(mnd)$ arithmetic operations and $O(nd)$ evaluations of f_i for $i = 0, 1, \ldots, m$.

4.1. Network flow problems associated with (P1) and (P2)

Minimum concave-cost flow problems is one of the most important and most difficult classes in both combinatorial and global optimization. To solve it many algorithms have been proposed so far (see [5, 3] and references therein), and some of them have turned out to be practically efficient for special problems. In particular, when the number of concave-cost arcs is fixed, one can solve the problem in polynomial time [4, 9, 12].

As is well known, every Hitchcock transportation problem can be transformed into a minimum cost flow problem and vice versa (see, e.g. [7]). Similarly, we can generate a minimum concave-cost flow problem from either (P1) or (P2) by equipping the underlying network with a super-source and m additional concave-cost arcs. The converse is also possible in the same way as in [9, 12], i.e., a certain class of minimum concave-cost network flow problems with m concave-cost arcs can be transformed into either (P1) or (P2), the detail of which will be discussed in the subsequent paper.

Acknowledgement

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

References

1. Dantzig, G.B., *Linear Programming and Extensions*, Princeton University Press (Princeton, NJ, 1963).
2. Dudziński, K. and S. Walukiewicz, 'Exact methods for the knapsack problem and its generalizations', *European Journal of Operational Research* **28** (1987), 3–21.

3. 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.
4. Guisewite, G.M. and P.M. Pardalos, 'A polynomial time solvable concave network flow problem', *Networks* **23** (1993), 143–149.
5. Horst, R. and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer-Verlag (Berlin, 1990).
6. Nauss, R.M., 'The 0–1 knapsack problem with multiple choice constraints', *European Journal of Operational Research* **2** (1978), 125–131.
7. Papadimitriou, C.H. and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Princeton-Hall Inc. (Englewood Cliffs, NJ, 1982).
8. Tuy, H., 'The complementarity convex structure in global optimization', *Journal of Global Optimization* **2** (1992), 21–40.
9. 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.
10. 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.
11. Tuy, H., S. Ghannadan, A. Migdalas, and P. Värbrand, 'Strongly polynomial algorithm for a production-transportation problem with a fixed number of nonlinear variables', Preprint, Department of Mathematics, Linköping University (Linköping 1993).
12. Tuy, H., S. Ghannadan, A. Migdalas, and P. Värbrand, 'The minimum concave cost network flow problem with a fixed number of sources and nonlinear arc costs', Preprint, Department of Mathematics, Linköping University (Linköping, 1993).