ACTA MATHEMATICA VIETNAMICA
Volume 22, Number 1, 1997, pp. 159–182

A PSEUDO-POLYNOMIAL ALGORITHM FOR SOLVING RANK THREE CONCAVE PRODUCTION-TRANSPORTATION PROBLEMS

TAKAHITO KUNO

Dedicated to Hoang Tuy on the occasion of his seventieth birthday

Abstract. In this paper, we extend the parametrization technique of Tuy et al. into a class of concave production-transportation problems with m (≥ 3) sources, n terminals and three nonlinear variables. We develop a depth-first-search algorithm for finding a globally optimal solution of this rank three concave minimization problem and show that the algorithm is pseudo-polynonomial in the problem input length but polynomial in m and n.

1. Introduction

Many optimization problems encountered in real-world applications have some special structures, which can often be exploited to design efficient algorithms. In global optimization, one of the most favorable structure is the *low rank monotonicity* studied by Tuy [11, 16, 17]. The nonconvexity of any rank k quasiconcave function g is located in a subspace of dimension k even if g is defined on a subset of a higher dimensional space than k. Therefore, the problem size that can be handled when the objective function is low rank is much larger than when it is full rank.

The class of low rank quasiconcave minimization includes multiplicative programming [10, 24], facility location [18], multilevel programming [23] and so forth [11]. Especially on networks, this class can take full advantage of another special structure, i.e. the *network* structure, and can be solved further efficiently [8, 9, 12-15, 19-22]. In fact, Tuy et al. have shown in a

Received December 31, 1996
Key words. Global optimization, rank three concave minimization, production-transportation problem, minimum concave cost flow problem, pseudo-polynomial algorithm.
The author was partially supported by Grant-in-Aid for Scientific Research of the Ministry of Education, Science, Sports and Culture, Grant No. (C)07680447.

series of articles [19, 20, 21] that a parametrization technique provides strongly polynomial algorithms for solving rank k *concave production-transportation problems*. The purpose of this paper is to extend their technique into a more general problem. Our problem setting is as follows:

Suppose a firm has m sources of a certain commodity, k of which are factories and the rest are warehouses. The decision maker of this firm has to cope with the demands at n terminal markets, so as to minimize the total cost of producing the commodity and of distributing it to each terminal. While the transportation cost is linear, the production cost is a nonlinear and concave function in the output due to economies of scale. As in [20], we assume in this paper that the number k of factories, each corresponding to a nonlinear variable, is fixed at three. Nevertheless, it would be hard for the algorithms in [19, 20, 21] to solve this problem of large size because their algorithms, designed for the case where $m = k$, are polynomial in n but exponential in the number m of sources.

As shown in [14], our problem can be reduced into a minimum concave-cost capacitated flow problem with k nonlinear arcs. Hence, it can also be solved by algorithms developed for the general minimum concave-cost flow problem, e.g. a branch-and-bound algorithm by Gallo et al. [5] and a dynamic programming algorithm by Erickson et al. [3]. In contrast to the ones in [19, 20, 21], both of these algorithms are strongly influenced by the number of terminals. Readers are also referred to [6, 7] for the current state-of-the-art of concave network optimization.

The algorithm developed in this paper for the above production-transportation problem is pseudo-polynomial in the problem input length but polynomial in both m and n. In Section 2, we apply the parametrization technique of Tuy et al. and transform the problem to an equivalent master problem via a parametric Hitchcock problem with three parameters. In Section 3, we characterize the pieces of linearity of the optimal value function of the parametric Hitchcock problem and define a plane graph associated with a family of those pieces. Among the vertices in this graph exists a global minimizer of the master problem. To find it, we develop a depth-first-search algorithm using dual pivot operations in Section 4. If the problem is nondegenerate, the proposed algorithm requires $O((m+n)\delta^2 + H(m,n))$ arithmetic operations, where $H(m,n)$ is the running time needed to solve a Hitchcock problem and δ is the difference between the total demand at the markets and the total supply at the warehouses. We close the paper with discussing how to avoid degeneracy in Section 5.

2. Formulation of the problem

We have m sources and n terminals of the commodity. Sources 1, 2, and 3 are factories, which produce y_1, y_2 and y_3 units, respectively, at a cost $g(y_1, y_2, y_3)$. We assume that the production cost $g : \mathbf{R}^3 \to \mathbf{R}$ is a concave function, and for simplicity that the production capacity of each factory is sufficiently large. The rest of sources are warehouses, each of which produces nothing but has a supply of a_i units, $i = 4, \ldots, m$. Each terminal represents a market with a demand of b_j units, $j = 1, \ldots, n$. We also know the unit cost c_{ij} of shipping the commodity from source i to terminal j. Figure 2.1 shows an example of the problem with $m = 4$ and $n = 6$.

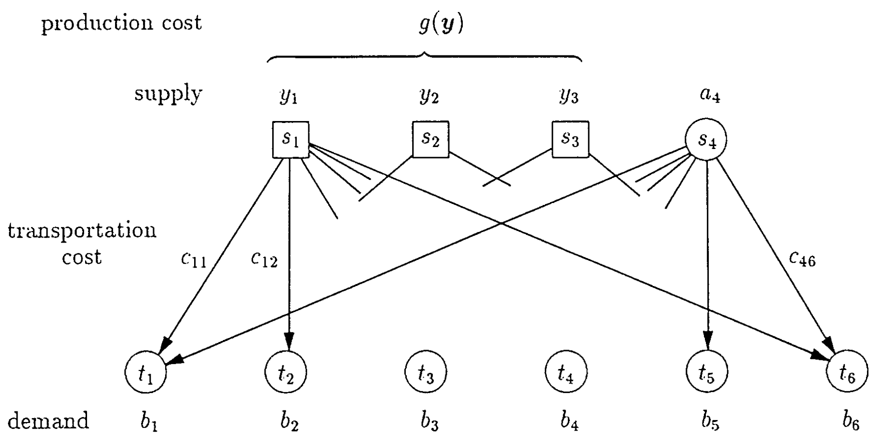

Figure 2.1. Example of the problem

Let x_{ij} denote the number of units shipped from source i to terminal j. Then our problem is formulated as follows:

$$
[\mathrm{P}] \quad \left|\begin{array}{l}
\text{minimize } \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + g(\mathbf{y}) \\
\text{subject to } \displaystyle\sum_{j=1}^{n} x_{ij} = \begin{cases} y_i, & i = 1, 2, 3 \\ a_i, & i = 4, \ldots, m \end{cases} \\
\qquad\qquad \displaystyle\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, \ldots, n \\
\qquad\qquad \mathbf{x} \geq \mathbf{0}, \quad \mathbf{y} \geq \mathbf{0},
\end{array}\right.
$$

where $\mathbf{x} \in \mathbf{R}^{m\times n}$ and $\mathbf{y} \in \mathbf{R}^3$ consist of variables x_{ij}'s and y_i's, respectively. We assume throughout the paper that constants a_i's, b_j's and c_{ij}'s are all positive integers, and that

$$\delta = \sum_{j=1}^{n} b_j - \sum_{i=4}^{m} a_i > 0. \tag{2.1}$$

This implies that [P] is feasible and has an optimal solution, because the objective function is continuous and bounded from below over the feasible region. Note that, to balance the total supply and demand, any feasible production $\mathbf{y}$ must lie in a two-dimensional simplex:

$$\Delta = \{\mathbf{y} \in \mathbf{R}^3 \mid y_1 + y_2 + y_3 = \delta,\ \mathbf{y} \geq \mathbf{0}\}. \tag{2.2}$$

Remark. Letting $\overline{g}(y_1, y_2) = g(y_1, y_2, \delta - y_1 - y_2)$, we can rewrite the objective function of [P] as $\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \overline{g}(y_1, y_2)$. Also, let

$$\begin{aligned}
\mathbf{c}^1 &= (\mathbf{0},\ 1,\ 0) \in \mathbf{R}^{m\times n} \times \mathbf{R} \times \mathbf{R},\\
\mathbf{c}^2 &= (\mathbf{0},\ 0,\ 1) \in \mathbf{R}^{m\times n} \times \mathbf{R} \times \mathbf{R},\\
\mathbf{c}^3 &= (\mathbf{c},\ 0,\ 0) \in \mathbf{R}^{m\times n} \times \mathbf{R} \times \mathbf{R},
\end{aligned}$$

where $\mathbf{c} \in \mathbf{R}^{m\times n}$ consists of c_{ij}'s. We then see that $\overline{g}$ is concave and c^i's are linearly independent; moreover,

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \overline{g}(y_1, y_2) \leq \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x'_{ij} + \overline{g}(y'_1, y'_2)$$

holds if $\mathbf{z} = (\mathbf{x}, y_1, y_2) \in \mathbf{R}^{m\times n} \times \mathbf{R} \times \mathbf{R}$ and $\mathbf{z}' = (\mathbf{x}', y'_1, y'_2) \in \mathbf{R}^{m\times n} \times \mathbf{R} \times \mathbf{R}$ satisfy

$$\langle \mathbf{c}^i, \mathbf{z} - \mathbf{z}' \rangle = 0, \quad i = 1, 2; \quad \langle \mathbf{c}^3, \mathbf{z} - \mathbf{z}' \rangle \leq 0.$$

Therefore, the objective function of [P] has rank three monotonicity with respect to $\mathbf{c}^i$, $i = 1, 2, 3$ [11, 16, 17]. □

2.1. Reduction to master problem

If we fix the values of y_i's in [P], we have Hitchcock problem:

$$
[\mathrm{P}(\mathbf{y})] \quad \left|\begin{array}{l}
\text{minimize } \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\
\text{subject to } \displaystyle\sum_{j=1}^{n} x_{ij} = \begin{cases} y_i, & i = 1,2,3 \\ a_i, & i = 4,\ldots,m \end{cases} \\
\qquad\qquad\quad \displaystyle\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,\ldots,n \\
\qquad\qquad\quad \mathbf{x} \geq \mathbf{0}.
\end{array}\right.
$$

We can solve $[\mathrm{P}(\mathbf{y})]$ efficiently using available algorithms, and obtain an optimal solution $\mathbf{x}^*(\mathbf{y})$ if and only if $\mathbf{y} \in \Delta$. Let us denote the optimal value by

$$
f(\mathbf{y}) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x^*_{ij}(\mathbf{y}). \tag{2.3}
$$

Then [P] is reduced to

$$
\text{minimize } \{f(\mathbf{y}) + g(\mathbf{y}) \mid \mathbf{y} \in \Delta\}, \tag{2.4}
$$

which we call the *master problem* of [P]. An immediate consequence is the following:

Theorem 2.1. *Let* $\mathbf{y}^*$ *be a global minimizer of* (2.4). *Then* $(\mathbf{x}^*(\mathbf{y}^*), \mathbf{y}^*)$ *solves* [P], *where* $\mathbf{x}^*(\mathbf{y}^*)$ *is an optimal solution of* $[\mathrm{P}(\mathbf{y}^*)]$.

Although the master problem (2.4) has only three variables, the objective function is more complicated than the original one. The key to finding its global minimum is offered by a well-known result on parametric linear programming (see e.g. [4]).

Lemma 2.2. *Function* $f : \Delta \to \mathbf{R}$ *is convex and polyhedral.*

The lemma implies that f is a pointwise maximum of finitely many affine functions. Let $\mathbf{d}_k \in \mathbf{R}^3$ and $d_{0k} \in \mathbf{R}$ for $k \in K$, where K is a finite set of indices, and suppose that f is expressed in the form

$$
f(\mathbf{y}) = \max\{\mathbf{d}_k^T\mathbf{y} + d_{0k} \mid k \in K\}, \quad \forall \mathbf{y} \in \Delta. \tag{2.5}
$$

Also, let

$$
F_k = \{\mathbf{y} \in \Delta \mid f(\mathbf{y}) = \mathbf{d}_k^T\mathbf{y} + d_{0k}\}, \quad k \in K. \tag{2.6}
$$

Then F_k is a convex polygon expressed as the intersection of Δ and $|K|-1$ halfspaces for each $k \in K$. We call a polyhedral subset of Δ a *linearity piece*, or a *piece* for short, of f if f is an affine function on it, as on F_k's. Any sum of affine and concave functions is concave, so that the objective function of (2.4) is concave on each F_k and attains the minimum at some vertex. We also see from (2.6) that F_k's is a covering of Δ, i.e. $\bigcup_{k\in K} F_k = \Delta$. Let $V(F_k)$ denote the set of vertices of F_k. Then we have the following:

Theorem 2.3. *Let*

$$\mathbf{y}^* \in \arg\min\{f(\mathbf{y}) + g(\mathbf{y}) \mid \mathbf{y} \in V(F_k),\ k \in K\}. \tag{2.7}$$

Then $\mathbf{y}^*$ *is a global minimizer of the master problem* (2.4)*.*

From Theorem 2.1 and 2.3, to solve the problem [P], we need only to enumerate the vertices of F_k's. Note that F_k's is not a unique family of linearity pieces that Theorem 2.3 applies to. Leaving the computational efficiency out of consideration, we can employ any finite family of pieces as long as it covers Δ. The readers are refered to [11, 19-21] for the formal proofs of the theorems.

3. Structure of a family of linearity pieces

Let us consider a Hitchcock problem associated with [P]:

$$[\mathrm{P}(\mathbf{y}')] \quad \left| \begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} x_{ij} = \begin{cases} y_i', & i = 1,2,3 \\ a_i, & i = 4,\ldots,m \end{cases} \\ & \displaystyle\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,\ldots,n \\ & \mathbf{x} \geq \mathbf{0}, \end{array} \right.$$

where $\mathbf{y}'$ is an arbitrary vector in Δ. Suppose an optimal basic solution $\mathbf{x}^*(\mathbf{y}')$ is given. Let B' denote the index set of basic variables in $\mathbf{x}^*(\mathbf{y}')$. Then the reduced cost $\overline{c}_{ij}$ satisfies

$$\overline{c}_{ij} = \begin{cases} 0 & \text{if } (i,j) \in B', \\ c_{ij} - \alpha_i - \beta_j \geq 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

where α_i and β_j are simplex multipliers, e.g. computed from

$$\alpha_1 = 0; \quad \alpha_i + \beta_j = c_{ij}, \quad (i,j) \in B'. \tag{3.2}$$

We see from (3.1) and (3.2) that the dual feasibility is not affected by any change in $\mathbf{y}'$. Hence, B' remains optimal to [P($\mathbf{y}$)] as long as the primal feasibility

$$x^*_{ij}(\mathbf{y}) \geq 0, \quad \forall (i,j) \in B' \tag{3.3}$$

holds for $\mathbf{y} \in \Delta$. Let

$$F_{B'} = \{\mathbf{y} \in \Delta \mid x^*_{ij}(\mathbf{y}) \geq 0,\ (i,j) \in B'\}. \tag{3.4}$$

Then $F_{B'}$ is obviously nonempty because $\mathbf{y}' \in F_{B'}$. For any $\mathbf{y} \in F_{B'}$, the optimal value of [P($\mathbf{y}$)] is given by

$$f(\mathbf{y}) = \sum_{(i,j) \in B'} c_{ij} x^*_{ij}(\mathbf{y}). \tag{3.5}$$

Since for each $(i,j) \in B'$ the value $x^*_{ij}(\mathbf{y})$ depends on $\mathbf{y}$ affinely, $F_{B'}$ is a polyhedral subset of Δ and f is an affine function on $F_{B'}$. These facts imply that $F_{B'}$ is a linearity piece of f. In the sequel, we will observe some properties of this piece $F_{B'}$.

3.1. Characterization of the piece $F_{B'}$

Let $G = (S, T, A)$ be the bipartite graph underlying the problem, where $S = \{s_i | i = 1, \ldots, m\}$ is the set of source nodes, $T = \{t_j | j = 1, \ldots, n\}$ is the set of terminal nodes and $A = S \times T$ (see Figure 2.1). For the optimal basis B' of [P($\mathbf{y}'$)], let

$$A_{B'} = \{(s_i, t_j) \mid (i,j) \in B'\}. \tag{3.6}$$

Then, as is well known [1, 2], the arc set $A_{B'}$ constitutes a spanning tree $G_{B'} = (S, T, A_{B'})$. Conversely, given a spanning tree consisting of an arc set A_B, we can compute a basic solution of [P($\mathbf{y}'$)] corresponding to the basis

$$B = \{(i,j) \mid (s_i, t_j) \in A_B\}, \tag{3.7}$$

by using the following procedure [1]:

procedure SOLUTION (A_B);

begin

$a(s_i) := y'_i$ for $i = 1, 2, 3$, and $a(s_i) := a_i$ for $i = 4, \ldots, m$;

$a(t_j) := -b_j$ for $j = 1, \ldots, n$;

$N := S \cup T$, $E := A_B$ and $G_E := (N, E)$;

for each $(s_i, t_j) \in A \setminus E$ **set** $x(s_i, t_j) := 0$;

while $N \neq \{s_1\}$ **do begin**

select a leaf node p_1 $(\neq s_1)$ in the subtree G_E and let p_2 be its adjacent node;

if $p_1 \in S$ **then** $x(p_1, p_2) := a(p_1)$ and $E := E \setminus \{(p_1, p_2)\}$

else $x(p_2, p_1) := -a(p_2)$ and $E := E \setminus \{(p_2, p_1)\}$;

$a(p_2) := a(p_2) + a(p_1)$;

$N := N \setminus \{p_1\}$ and $G_E := (N, E)$

end

end;

When node p_1 is selected as a leaf node, the number $a(p_1)$ represents the cumulative supply minus the cumulative demand at nodes connected with p_1 by paths in a subgraph $(S, T, A_B \setminus E)$. Therefore, if $A_{B'}$ is input to the procedure SOLUTION, it yields an optimal basic solution $\mathbf{x}^*(\mathbf{y}')$ in the form

$$x^*_{ij}(\mathbf{y}') = x(s_i, t_j) = \mathbf{d}_{ij}^T \mathbf{y}' + d_{0ij}, \quad (i, j) \in B', \tag{3.8}$$

where $\mathbf{d}_{ij}$ is a vector in either $\{0, 1\}^3$ or $\{0, -1\}^3$ and d_{0ij} is some integer. From (3.8), we can make a sketch of the piece $F_{B'}$.

Lemma 3.1. *The linearity piece $F_{B'}$ is a convex polygon such that each vertex is integral and each edge is parallel to some edge of the simplex Δ.*

Proof. It follows from (3.4) and (3.8) that $F_{B'}$ is the intersection of Δ and $m + n - 1$ halfspaces defined by

$$\mathbf{d}_{ij}^T \mathbf{y} + d_{0ij} \geq 0, \quad (i, j) \in B'.$$

Some of either boundary planes correspond to edges of $F_{B'}$. Since either $\mathbf{d}_{ij} \in \{0, 1\}^3$ or $\mathbf{d}_{ij} \in \{0, -1\}^3$, such a boundary plane is of the form either $y_{i_1} = d$ or $y_{i_1} + y_{i_2} = d$, where $\{i_1, i_2, i_3\}$ is a permutation of $\{1, 2, 3\}$ and

d is some integer. Hence, it determines an edge parallel to either the edge $y_{i_1} = 0$ or $y_{i_3} = 0$ of Δ. We also see that two distinct edges of $F_{B'}$ can only intersect at an integral point. □

Example 3.1. If $m = 3$, i.e. the case where warehouses are absent, Tuy et al. have shown in [20] that the piece $F_{B'}$ is either a triangle or parallelogram. In contrast to this, $F_{B'}$ can be a trapezoid, pentagon or hexagon as well in our problem with $m > 3$. Actually, for an instance with constraints:

(3.9)

$$\sum_{j=1}^{6} x_{ij} = \begin{cases} y_i, & i = 1,2,3 \\ 7, & i = 4 \end{cases}; \ \sum_{i=1}^{4} x_{ij} = \begin{cases} 2, & j = 1,2,3 \\ 4, & j = 4,5,6 \end{cases}; \ \mathbf{x} \geq \mathbf{0}; \ \mathbf{y} \geq \mathbf{0},$$

we have

$$\begin{aligned} x_{11} &= 2, & x_{14} &= y_1 - 2, & x_{44} &= 6 - y_1 \\ x_{22} &= 2, & x_{25} &= y_2 - 2, & x_{45} &= 6 - y_2 \\ x_{33} &= 2, & x_{36} &= y_3 - 2, & x_{46} &= 6 - y_3 \end{aligned}$$

with respect to a basic B' given by the spanning tree in Figure 3.1 (a). Then

$$F_{B'} = \{\mathbf{y} \in \Delta \mid 2 \leq y_i \leq 6, \ i = 1,2,3\}$$

is a hexagon as shown in Figure 3.1 (b). □

3.2. Nondegeneracy assumption

Since $\mathbf{y}' \in \Delta$ is arbitrary in the above observation, for any $\mathbf{y} \in \Delta$ we can obtain a linearity piece F_B with $\mathbf{y} \in F_B$ from an optimal basic solution $\mathbf{x}^*(\mathbf{y})$ of [P($\mathbf{y}$)]. In other words, those pieces form a covering of Δ, denoted by $\mathcal{F}$. We should also note that $\mathcal{F}$ is a finite family because the number of optimal bases of [P($\mathbf{y}$)]'s, each of which corresponds to a member of $\mathcal{F}$, is finite. Hence, we can compute a global minimizer $\mathbf{y}^*$ of the master problem (2.4) by enumerating the vertices of each $F_B \in \mathcal{F}$ in accordance with Theorem 2.3. To state this systematically, we impose a nondegeneracy assumption on [P($\mathbf{y}$)] hereafter.

Assumption 3.1. For any $\mathbf{y} \in \Delta$, problem [P($\mathbf{y}$)] has a unique optimal solution $\mathbf{x}^*(\mathbf{y})$, which has at least $m + n - 3$ positive components.

Therefore, $\mathbf{x}^*(\mathbf{y})$ is an optimal basic solution with at most two zero-valued basic variables. This assumption is certainly a big one, especially in combinatorial problems like [P($\mathbf{y}$)]. In Section 5, we will discuss this matter again and show how to avoid degeneracy in detail.

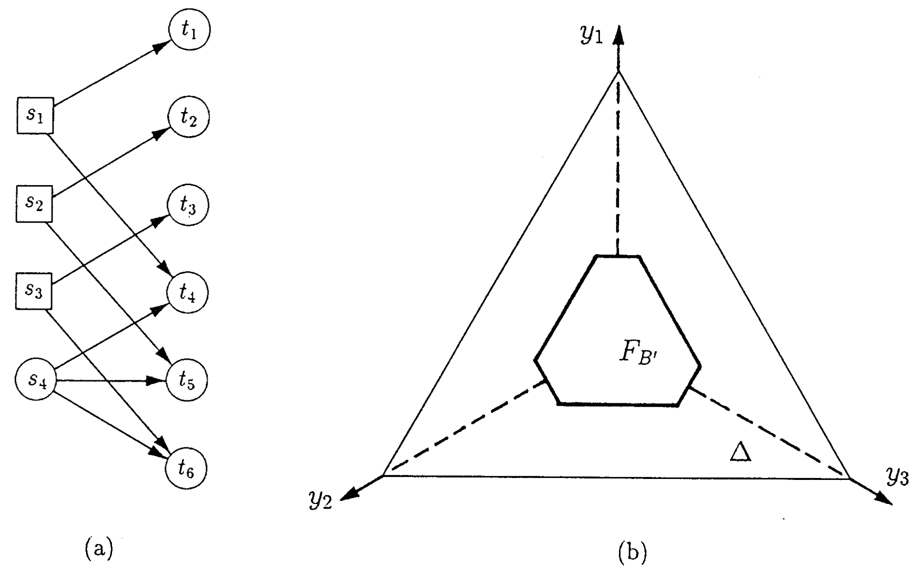

Figure 3.1. Example of the piece $F_{B'}$

Due to Assumption 3.1, the structure of $\mathcal{F}$ is rather orderly as follows:

Lemma 3.2. *Each $F_B \in \mathcal{F}$ has a nonempty interior relative to Δ.*

Proof. If F_B has no interior points, each $\mathbf{y} \in F_B$ lies in a line determined by two zero-valued basic variables in $\mathbf{x}^*(\mathbf{y})$. Since F_B included in the triangle Δ, it must have end points in that line. At each of the end points, one more basic variable vanishes, which contradicts Assumption 3.1. $\square$

Lemma 3.3. *Let F_B and $F_{B'}$ be two distinct pieces in $\mathcal{F}$. Then*

$$\operatorname{int} F_B \cap \operatorname{int} F_{B'} = \emptyset, \tag{3.10}$$

where int. *denotes the interior relative to Δ.*

Proof. Since $F_B \neq F_{B'}$, the corresponding bases B and B' are also distinct. If there is a point $\mathbf{y} \in \operatorname{int} F_B \cap \operatorname{int} F_{B'}$, we have

$$\begin{aligned} &x^*_{ij}(\mathbf{y}) > 0,\ \forall (i,j) \in B; \quad x^*_{ij}(\mathbf{y}) = 0,\ \forall (i,j) \notin B, \\ &x^*_{ij}(\mathbf{y}) > 0,\ \forall (i,j) \in B'; \quad x^*_{ij}(\mathbf{y}) = 0,\ \forall (i,j) \notin B'. \end{aligned}$$

This is impossible, because [P($\mathbf{y}$)] cannot have two optimal solutions. $\square$

Lemma 3.4. *Let F_B and $F_{B'}$ be two distinct pieces in $\mathcal{F}$. If $F_B \cap F_{B'} \neq \emptyset$, they share either a vertex or an edge.*

Proof. Assumming the contrary, we have a vertex $\mathbf{v}$ of F_B lying in the relative interior of some edge of $F_{B'}$. Then the optimal solution $\mathbf{x}^*(\mathbf{v})$ of [P($\mathbf{v}$)] corresponding to B has $m+n-3$ positive components while that corresponding to B' has $m+n-2$ positive components. This is a contradiction. □

3.3. Associated plane graph

From Lemmas 3.2 and 3.3, we see that under Assumption 3.1 the family $\mathcal{F}$ is minimal among those that cover Δ, i.e. $\mathcal{F}$ is a partition of Δ. Let $\mathcal{V} \subset \Delta$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denote the set of vertices and the set of edges, respectively, of all $F_B \in \mathcal{F}$. Then, from Lemma 3.4, the pair $(\mathcal{V}, \mathcal{E})$ constitutes a connected graph embedded in the triangle Δ. Obviously, it involves the three vertices $\mathbf{v}^1 = (\delta, 0, 0)^T$, $\mathbf{v}^2 = (0, \delta, 0)^T$ and $\mathbf{v}^3 = (0, 0, \delta)^T$ of Δ. Also, from Lemma 3.1, the edge set $\mathcal{E}$ is partitioned into three subsets:
(3.11)

$$\mathcal{E}_i = \{(\mathbf{v}, \mathbf{w}) \in \mathcal{E} \,|\, (\mathbf{v}, \mathbf{w}) \text{ is parallel to the edge } y_i = 0 \text{ of } \Delta\}, \ i = 1, 2, 3.$$

This plane graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ gives us an insight into the master problem (2.4) (see also Figure 4.2 in Section 4). Let $\{i_1, i_2, i_3\}$ denote any permutation of $\{1, 2, 3\}$.

Lemma 3.5. *For each $\mathbf{v} \in \mathcal{V}$ with $v_{i_1} > 0$, there is some $\mathbf{w} \in \mathcal{V}$ with $w_{i_1} < v_{i_1}$ such that the segment $(\mathbf{v}, \mathbf{w})$ belongs to either $\mathcal{E}_{i_2}$ or $\mathcal{E}_{i_3}$.*

Proof. Let $F_B \in \mathcal{F}$ be a piece with the vertex $\mathbf{v}$. If $v_{i_1} > y_{i_1}$ for some $\mathbf{y} \in F_B$, then F_B obviously has an edge $(\mathbf{v}, \mathbf{w})$ with $w_{i_1} < v_{i_1}$, which belongs to either $\mathcal{E}_{i_2}$ or $\mathcal{E}_{i_3}$. Suppose $v_{i_1} \leq y_{i_1}$ for all $\mathbf{y} \in F_B$. Since $v_{i_1} > 0$ and $\mathcal{F}$ covers Δ, there is a piece $F_{B'} \in \mathcal{F}$ such that the segment $(\mathbf{v}, \mathbf{v} - \varepsilon \mathbf{e}_{i_1})$ is included in $F_{B'}$ for sufficiently small $\varepsilon > 0$, where $\mathbf{e}_{i_1} \in \mathbf{R}^3$ is the i_1th unit vector. From Lemma 3.4, the point $\mathbf{v}$ is a vertex of $F_{B'}$ and hence is adjacent to some vertex $\mathbf{w}$ with $w_{i_1} < v_{i_1}$ of $F_{B'}$. Therefore, $(\mathbf{v}, \mathbf{w})$ belongs to either $\mathcal{E}_{i_2}$ or $\mathcal{E}_{i_3}$. □

Using inductive arguments, we can obtain the following:

Theorem 3.6. *Vertex $\mathbf{v}^{i_1}$ of Δ and all other vertices in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ are connected by paths consisting of only edges in $\mathcal{E}_{i_2} \cup \mathcal{E}_{i_3}$.*

In the next section, based upon the above observations, we will construct an algorithm for visiting all the vertices in $\mathcal{G}$.

4. Enumeration of the vertices of linearity pieces

So far we have seen that a way to solve the master problem (2.4) is the enumeration of the vertices in $\mathcal{G}$. For this purpose, we apply a depth-first-search procedure to a subgraph $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_2 \cup \mathcal{E}_3)$. Starting from the vertex $\mathbf{v}^1$ of Δ, the procedure recursively visits each unexplored vertex in $\mathcal{V}$ along an edge in $\mathcal{E}_2 \cup \mathcal{E}_3$. According to Theorem 3.6, all the vertices in $\mathcal{G}$ turn explored ones by the end of the procedure. Let us suppose that $\mathbf{v}$ with $v_1 > 0$ is the most recently visited vertex and try to locate each unexplored vertex $\mathbf{w}$ with $w_1 < v_1$ adjacent to $\mathbf{v}$ in $\mathcal{G}_1$ (see Figure 4.1, where the vertices are numbered in the order of visit). Before proceeding to the procedure, we have to make a remark on the relation between $\mathbf{v}$ and the graph $G = (S, T, A)$.

Let B be an optimal basis of $[\mathrm{P}(\mathbf{v})]$. As seen in the previous section, $A_B = \{(s_i, t_j) | (i, j) \in B\}$ constitutes a spanning tree $G_B = (S, T, A_B)$. Let $\Pi_B(p_1, p_2)$ denote the path from node p_1 to node p_2 in G_B. Also, let $D_B(\mathbf{v})$ be the set of degenerate arcs, i.e., arcs with zero flow of the commodity in G_B with respect to $\mathbf{x}^*(\mathbf{v})$. Regardless of Assumption 3.1, we have the following:

Lemma 4.1. *For each pair (s_{i_1}, s_{i_2}) with $\{i_1, i_2\} \subset \{1, 2, 3\}$,*

$$\Pi_B(s_{i_1}, s_{i_2}) \cap D_B(\mathbf{v}) \neq \emptyset \tag{4.1}$$

holds if and only if $\mathbf{v}$ is a vertex of the piece F_B.

Proof. Suppose $\Pi_B(s_{i_1}, s_{i_2}) \cap D_B(\mathbf{v}) = \emptyset$ for some (s_{i_1}, s_{i_2}). Obviously, $\Pi_B(s_{i_2}, s_{i_1}) \cap D_B(\mathbf{v}) = \emptyset$ holds as well. Hence, we can send a positive amount of flow along both directions from s_{i_1} to s_{i_2} and from s_{i_2} to s_{i_1} in G_B, while keeping the flow nonnegative on each tree arc. This implies that $\mathbf{v}$ is expressed as a convex combination of points in F_B. Conversely, if (4.1) holds for each (s_{i_1}, s_{i_2}), the flow between s_{i_1} and s_{i_2} is allowed to change along at most one direction in G_B, which implies that $\mathbf{v}$ is extremal in F_B. □

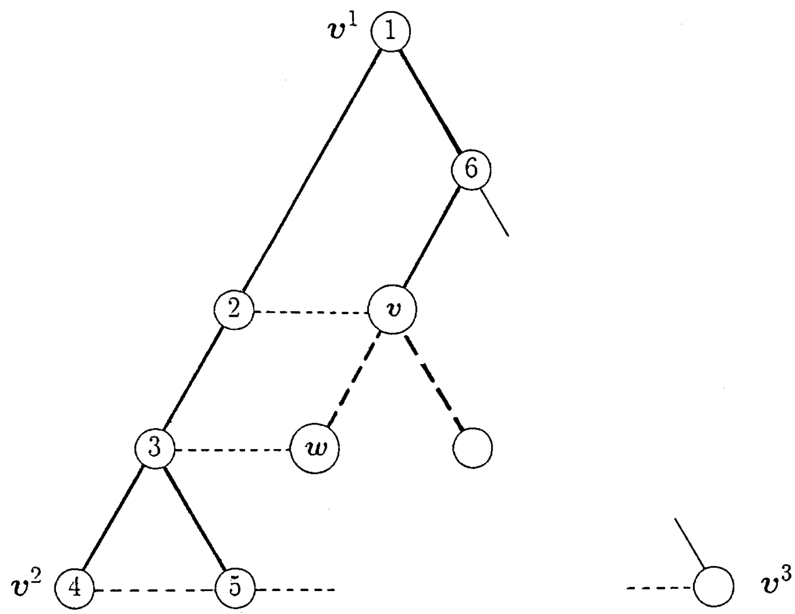

Figure 4.1. The depth-first search for $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_2 \cup \mathcal{E}_3)$

4.1. Dual pivot operation

Let us start with the search of $\mathcal{E}_2$ for an edge $(\mathbf{v}, \mathbf{w})$. Recall that each edge in $\mathcal{E}$ is determined by a single zero-valued basic variable. Hence, by Assumption 3.1, moving from $\mathbf{v}$ to $\mathbf{w}$ along $(\mathbf{v}, \mathbf{w}) \in \mathcal{E}_2$, if it exists, amounts to augmenting the flow from s_2 to s_1 in some spanning tree while keeping one tree arc degenerate. From Lemma 4.1, the path $\Pi_B(s_2, s_1)$ contains at least one degenerate arc. If such an arc is a backward one, it blocks the augmentation from s_2 to s_1 in G_B. However, by performing at most two dual pivot operations on G_B, we can obtain an alternative spanning tree, in which the path from s_2 to s_1 contains no blocking arcs for $\mathbf{x}^*(\mathbf{v})$.

Let $(s_q, t_r) \in D_B(\mathbf{v})$ be the blocking arc closest to s_2 in $\Pi_B(s_2, s_1)$. By dropping (s_q, t_r) from G_B, we have two subtrees G_1 and G_2 such that s_1, $s_q \in G_1$ and s_2, $t_r \in G_2$. Let N_1 and N_2 be the sets of nodes spanned by G_1 and G_2, respectively. Then we have an s_1-s_2 cutset:

$$[N_1, N_2] = (N_1, N_2) \cup (N_2, N_1), \tag{4.2}$$

where

$$(4.3)\qquad \begin{cases} (N_1,N_2) &= \{(s_i,t_j) \mid s_i \in N_1,\ t_j \in N_2\}, \\ (N_2,N_1) &= \{(s_i,t_j) \mid s_i \in N_2,\ t_j \in N_1\}. \end{cases}$$

Let

$$(4.4)\qquad (s_k,t_\ell) \in \arg\min\{\overline{c}_{ij} \mid (s_i,t_j) \in (N_2,N_1)\},$$

where $\overline{c}_{ij}$ is the reduced cost relative to B. Note that $\overline{c}_{ij} \geq 0$ for each $(s_i,t_j) \in A$ because B is an optimal basis. By adding the arc (s_k,t_ℓ) to G_1 and G_2, we have a spanning tree $G_{B'}$, where $B' = (B \setminus \{(p,q)\}) \cup \{(k,\ell)\}$. We see from (3.1) and (3.2) that this operation decreases the simplex multiplier α_i by $\overline{c}_{k\ell}$ if $s_i \in N_1$, and increases β_j by $\overline{c}_{k\ell}$ if $t_j \in N_1$. Consequently, the reduced cost changes into

$$(4.5)\qquad \overline{c}'_{ij} = \begin{cases} \overline{c}_{ij} + \overline{c}_{k\ell} & \text{if } (s_i,t_j) \in (N_1,N_2), \\ \overline{c}_{ij} - \overline{c}_{k\ell} & \text{if } (s_i,t_j) \in (N_2,N_1), \\ \overline{c}_{ij} & \text{otherwise.} \end{cases}$$

It follows from (4.4) that $\overline{c}'_{ij} \geq 0$ for each $(s_i,t_j) \in A$. Hence, B' is another optimal basis of [P($\mathbf{v}$)].

The path $\Pi_{B'}(s_2,s_1)$ in the resulting tree $G_{B'}$ might still contain a blocking arc. In that case, we have to perform the dual pivot operation on $G_{B'}$ once more. Then each backward degenerate arc is replaced by a forward degenerate arc and the tree path from s_2 to s_1 has no blocking arcs.

4.2. Moving from vertex v to w

We can now assume that $\Pi_B(s_2,s_1)$ is an s_2-s_1 augmenting path in G with respect to $\mathbf{x}^*(\mathbf{v})$, though it contains one or two degenerate arcs as forward arcs. We then have two cases to consider.

Case 1 : $|\Pi_B(s_2,s_1) \cap D_B(\mathbf{v})| = 1$.

Let $\underline{\Pi}_B(s_2,s_1)$ and $\overline{\Pi}_B(s_2,s_1)$ denote the sets of backward and forward arcs, respectively, in $\Pi_B(s_2,s_1)$. By assumption, a degenerate arc, say (s_q,t_r), belongs to $\overline{\Pi}_B(s_2,s_1)$. Let

$$(4.6)\qquad \sigma = \min\{x^*_{ij}(\mathbf{v}) \mid (s_i,t_j) \in \underline{\Pi}_B(s_2,s_1)\}.$$

Then σ is the maximum amount of flow that can be sent from s_2 to s_1 along $\Pi_B(s_2, s_1)$. Since $\overline{c}_{ij} = 0$ for each $(i, j) \in B$, sending σ units along $\Pi_B(s_2, s_1)$ preserves both the primal and dual feasibility. Therefore,

$$x^*_{ij}(\mathbf{w}) = \begin{cases} x^*_{ij}(\mathbf{v}) - \sigma & \text{if } (s_i, t_j) \in \underline{\Pi}_B(s_2, s_1), \\ x^*_{ij}(\mathbf{v}) + \sigma & \text{if } (s_i, t_j) \in \overline{\Pi}_B(s_2, s_1), \\ x^*_{ij}(\mathbf{v}) & \text{otherwise}, \end{cases} \tag{4.7}$$

remains optimal to [P($\mathbf{w}$)] for

$$\mathbf{w} = (v_1 - \sigma, v_2 + \sigma, v_3)^T. \tag{4.8}$$

If $x^*_{k\ell}(\mathbf{v}) = \sigma$ for $(s_k, t_\ell) \in \underline{\Pi}_B(s_2, s_1)$, this operation replaces (s_q, t_r) by (s_k, t_ℓ) as a degenerate arc. However, it never changes the flow on the other degenerate arc, not contained in $\Pi_B(s_2, s_1)$. Thus, we have $\mathbf{w} \in \mathcal{V}$ and $(\mathbf{v}, \mathbf{w}) \in \mathcal{E}_2$.

Case 2 : $|\Pi_B(s_2, s_1) \cap D_B(\mathbf{v})| = 2$.

Even if $\Pi_B(s_2, s_1)$ contains two degenerate arcs, we can send a sufficiently small amount of flow, say ε units, along $\Pi_B(s_2, s_1)$. Doing so, however, makes all the tree arcs nondegenerate. The point $(v_1 - \varepsilon, v_2 + \varepsilon, v_3)^T$ then lies in the interior of the piece F_B. Since $\mathcal{F}$ is a partition of Δ, we can conclude that no edges in $\mathcal{E}_2$ are incident from $\mathbf{v}$ to $\mathbf{w}$ with $w_1 < v_1$.

After completing the search of $\mathcal{E}_2$, we next search $\mathcal{E}_3$ for an edge $(\mathbf{v}, \mathbf{w})$. From Lemma 4.1, we have

$$\Pi_B(s_2, s_1) \cap D_B(\mathbf{v}) \neq \Pi_B(s_3, s_1) \cap D_B(\mathbf{v}). \tag{4.9}$$

Otherwise, there is a node p in $\Pi_B(s_2, s_1) \cap \Pi_B(s_3, s_1)$ such that the tree path from s_2 to s_3 via p contains no degenerate arcs. Since $|D_B(\mathbf{v})| = 2$, the number of degenerate arcs in $\Pi_B(s_3, s_1) \setminus \Pi_B(s_2, s_1)$ is at most one. Moreover, if $\Pi_B(s_3, s_1)$ shares a degenerate arc with $\Pi_B(s_2, s_1)$ for $\mathbf{x}^*(\mathbf{v})$, it must be a forward arc in both the paths. Therefore, at most one dual pivot operation on G_B gives an s_3-s_1 augmenting path with respect to $\mathbf{x}^*(\mathbf{v})$. The rest is the same as the search of $\mathcal{E}_2$.

4.3. Checking of vertex w

If the first component of the vertex $\mathbf{w}$ located from $\mathbf{v}$ is zero, $\mathbf{w}$ has no descendants in $\mathcal{G}_1$. In addition to this, we have to terminate the search and backtrack to $\mathbf{v}$ when we find that $\mathbf{w}$ has already been visited. Since the procedure searches Δ in the direction from $\mathbf{v}^2$ to $\mathbf{v}^3$, the vertex $\mathbf{w}$ can be visited twice only if $(\mathbf{v}, \mathbf{w}) \in \mathcal{E}_2$ and $w_3 > 0$ (see Figure 4.1). In such a case, we construct an s_1-s_3 augmenting path with respect to $\mathbf{x}^*(\mathbf{w})$ in order to check if $\mathbf{w}$ is a visited vertex or not.

Since for $\mathbf{x}^*(\mathbf{w})$ the path $\Pi_B(s_2, s_1)$ contains just one degenerate arc (s_k, t_ℓ) as a backward arc, the other degenerate arc lies in $\Pi_B(s_3, s_1)$, as either a forward or backward arc. Hence, by performing a dual pivot operation on G_B if necessary, we have a spanning tree $G_{B'}$ and an s_1-s_3 augmenting path $\Pi_{B'}(s_1, s_3)$. If $\Pi_{B'}(s_1, s_3)$ contains only one degenerate arc, we can send a positive amount of flow along $\Pi_{B'}(s_1, s_3)$ while keeping the flow zero on the other degenerate arc. This implies that there is a vertex $\mathbf{v}' \in \mathcal{V}$ such that $(\mathbf{v}', \mathbf{w}) \in \mathcal{S}_3$ and $v'_1 > w_1$. In other words, $\mathbf{w}$ must have been visited from $\mathbf{v}'$. If $\Pi_{B'}(s_1, s_3)$ contains two degenerate arcs, we accept $\mathbf{w}$ as an unexplored vertex and initiate a new search from $\mathbf{w}$. In the latter case, we should note that $\Pi_{B'}(s_2, s_1)$ contains only one blocking arc. Hence, one dual pivot operation on $G_{B'}$ gives an s_2-s_1 augmenting path with respect to $\mathbf{x}^*(\mathbf{w})$.

4.4. Depth-first-search algorithm

The entire algorithm is summarized below. Incorporating all the above operations, the algorithm 3FACTORIES enumerates the vertices in $\mathcal{G}_1$ using the recursive procedure SEARCH and yields an optimal solution $(\mathbf{x}^*, \mathbf{y}^*)$ of [P] as well as a global minimizer $\mathbf{y}^*$ of the master problem (2.4).

algorithm 3FACTORIES;

begin

compute $\mathbf{x}^*(\mathbf{v}^1)$ by solving a Hitchcock problem $[\mathrm{P}(\mathbf{v}^1)]$;

$z^* := +\infty$;

SEARCH $(\mathbf{v}^1)$

end;

procedure SEARCH $(\mathbf{v})$;

begin

if $f(\mathbf{v}) + g(\mathbf{v}) < z^*$ **then** update $(\mathbf{x}^*, \mathbf{y}^*) := (\mathbf{x}^*(\mathbf{v}), \mathbf{v})$ and
$z^* := f(\mathbf{v}) + g(\mathbf{v})$;

if $v_1 > 0$ **then begin**

construct an s_2-s_1 augmenting path $\Pi_B(s_2, s_1)$ in G with respect to $\mathbf{x}^*(\mathbf{v})$;

if the number of degenerate arcs in $\Pi_B(s_2, s_1)$ is one **then begin**

Let $\underline{\Pi}_B(s_2, s_1)$ be the set of backward arcs in $\Pi_B(s_2, s_1)$;

$\sigma := \min\{x^*_{ij}(\mathbf{v}) \mid (s_i, t_j) \in \underline{\Pi}_B(s_2, s_1)\}$;

$\mathbf{w} := (v_1 - \sigma, v_2 + \sigma, v_3)^T$;

compute $\mathbf{x}^*(\mathbf{w})$;

if $w_3 = 0$ **then** SEARCH ($\mathbf{w}$)

else begin

construct an s_1-s_3 augmenting path $\Pi_{B'}(s_1, s_3)$ with respect to $\mathbf{x}^*(\mathbf{w})$;

if the number of degenerate arcs in $\Pi_{B'}(s_1, s_3)$ is two **then**

SEARCH ($\mathbf{w}$)

end

end;

construct an s_3-s_1 augmenting path $\Pi_B(s_3, s_1)$ with respect to $\mathbf{x}^*(\mathbf{v})$;

if the number of degenerate arcs in $\Pi_B(s_3, s_1)$ is one **then begin**

let $\underline{\Pi}_B(s_3, s_1)$ be the set of backward arcs in $\Pi_B(s_3, s_1)$;

$\sigma := \min\{x^*_{ij}(\mathbf{v}) \mid (s_i, t_j) \in \underline{\Pi}_B(s_3, s_1)\}$;

$\mathbf{w} := (v_1 - \sigma, v_2, v_3 + \sigma)^T$;

compute $\mathbf{x}^*(\mathbf{w})$;

SEARCH ($\mathbf{w}$)

end

end

end;

Theorem 4.2. *Under Assumption 3.1, the algorithm 3FACTORIES requires* $O((m+n)\delta^2 + H(m+n))$ *arithmetic operations and* $O(\delta^2)$ *evaluations of* g*, where* $\delta = \sum_{j=1}^{n} b_j - \sum_{i=4}^{m} a_i$ *and* $H(m,n)$ *is the running time needed to solve a Hitchcock problem.*

Proof. After solving a Hitchcock problem [P($\mathbf{v}^1$)], the algorithm begins the depth-first search of $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_2 \cup \mathcal{E}_3)$ at the vertex $\mathbf{v}^1$. Each element of $\mathcal{V}$ is a vertex of some linearity piece $F_B \in \mathcal{F}$ and hence is an integral point in the triangle Δ (Lemma 3.1). Therefore, $|\mathcal{V}|$ is bounded by $O(\delta^2)$. For each $\mathbf{v} \in \mathcal{V}$, the procedure SEARCH evaluates $g(\mathbf{v})$ and executes $O(1)$ dual pivot operations, each of which requires $O(m+n)$ arithmetic operations (see e.g. [1]). □

While $H(m,n)$ is known to be strongly polynomial (see e.g. [1]), the number δ^2 cannot be bounded by any polynomial in the problem input length. The algorithm 3FACTORIES is therefore not a polynomial but pseudo-polynomial algorithm even if the value of f is provided by an oracle. However, it is still worth noting that the running time is a lower-order polynomial in (m,n). This will guarantee the performance of 3FACTORIES for instances with rather large (m,n)'s as long as δ is a relatively small number. Computational experiments are now underway, the detail of which will be reported elsewhere, together with the extension of 3FACTORIES to the case of k (> 3) factories.

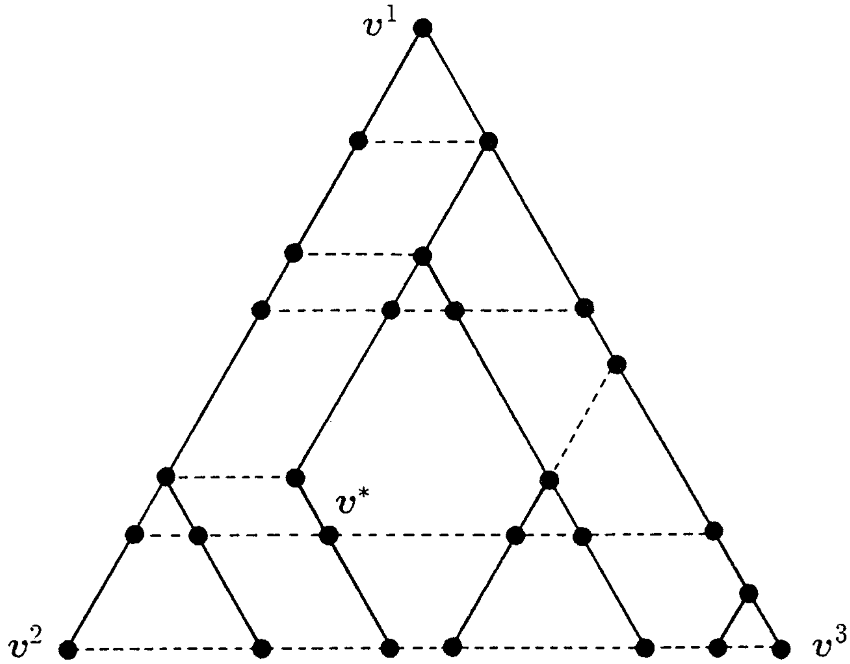

Figure 4.2. Illustration of the algorithm 3FACTORIES

Example 4.1. Figure 4.2 shows a search tree of $\mathcal{G}$ depicted by the algorithm 3FACTORIES when it solves an instance of minimizing

$$\sum_{i=1}^{4}\sum_{j=1}^{6} 2^{\rho_{ij}} x_{ij} + \sum_{i=1}^{3} 10^i \sqrt{y_i}$$

under the constraints (3.9) in Example 3.1, where ρ_{ij}'s are given by

$$[\rho_{ij}] = \begin{bmatrix} 1 & 20 & 17 & 5 & 13 & 21 \\ 22 & 2 & 10 & 18 & 6 & 14 \\ 15 & 23 & 3 & 11 & 19 & 7 \\ 12 & 16 & 24 & 4 & 8 & 9 \end{bmatrix}.$$

Starting from the vertex $\mathbf{v}^1 = (11, 0, 0)^T$ of $\Delta = \{\mathbf{y} \in \mathbf{R}^3 | y_1 + y_2 + y_3 = 11, \mathbf{y} \geq \mathbf{0}\}$, the procedure SEARCH traverses the tree from the left to the right in preorder. The vertex $\mathbf{v}^* = (2, 6, 3)^T$ provides a globally optimal solution $\mathbf{x}^*(\mathbf{v}^*)$, each component is as follows:

$$[x^*_{ij}(\mathbf{v}^*)] = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 4 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 4 & 0 & 3 \end{bmatrix}.$$

5. Disposal of degeneracy

Before closing the paper, we have to return to the postponed matter, i.e. how to deal with degenerate problems not satisfying Assumption 3.1.

5.1. Perturbed problem

Let us slightly perturb the constants in [P] and define

$$(5.1) \quad \left| \begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}(\varepsilon) x_{ij} + g(\mathbf{y}) \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} x_{ij} = \begin{cases} y_i, & i = 1, 2, 3 \\ a_i, & i = 4, \ldots, m \end{cases} \\ & \displaystyle\sum_{i=1}^{m} x_{ij} = b_j(\varepsilon), \quad j = 1, \ldots, n \\ & \mathbf{x} \geq \mathbf{0}, \quad \mathbf{y} \geq \mathbf{0}, \end{array} \right.$$

where

$$(5.2)\qquad \begin{aligned} b_j(\varepsilon) &= b_j + \varepsilon, && j = 1, \dots, n \\ c_{ij}(\varepsilon) &= c_{ij} + \varepsilon^{(i-1)n+j}, && i = 1, \dots, m;\ j = 1, \dots, n, \end{aligned}$$

and $\varepsilon = 1/(n+1)$. Let $[\mathrm{P}(\mathbf{y};\varepsilon)]$ denote the subproblem of (5.1) for any fixed $\mathbf{y}$. As $[\mathrm{P}(\mathbf{y})]$ does, $[\mathrm{P}(\mathbf{y};\varepsilon)]$ has an optimal solution if and only if $\mathbf{y}$ lies in a simplex:

$$(5.3)\qquad \Delta(\varepsilon) = \{\mathbf{y} \in \mathbf{R}^3 \mid y_1 + y_2 + y_3 = \delta + n\varepsilon,\ \mathbf{y} \geq \mathbf{0}\}.$$

Lemma 5.1. *For any* $\mathbf{y} \in \Delta(\varepsilon)$*, the problem* $[\mathrm{P}(\mathbf{y};\varepsilon)]$ *has a unique optimal solution.*

Proof. Let $\mathbf{x}^*(\mathbf{y};\varepsilon)$ be an optimal basic solution with basic variables x_{ij}, $(i,j) \in B$. Then the reduced costs satisfy

$$(5.4)\qquad \overline{c}_{ij}(\varepsilon) = c_{ij}(\varepsilon) - \alpha_i - \beta_j \geq 0, \quad \forall (i,j) \notin B,$$

where $\alpha_1 = 0$ and $\alpha_i + \beta_j = c_{ij}(\varepsilon)$ for each $(i,j) \in B$. Since the powers of ε in $c_{ij}(\varepsilon)$'s are all distinct, they never cancel out in the process of computing $\overline{c}_{ij}(\varepsilon)$'s. Therefore, $\overline{c}_{ij}(\varepsilon)$'s are polynomial in ε and their degrees are distinct each other. This, together with the fact that c_{ij}'s are positive integers, implies that all the inequalities in (5.4) hold strictly. Hence, $\mathbf{x}^*(\mathbf{y};\varepsilon)$ is a dual nondegenerate and unique optimal solution of $[\mathrm{P}(\mathbf{y};\varepsilon)]$. □

Lemma 5.2. *For any* $\mathbf{y} \in \Delta(\varepsilon)$*, the optimal solution* $\mathbf{x}^*(\mathbf{y};\varepsilon)$ *of* $[\mathrm{P}(\mathbf{y};\varepsilon)]$ *has at least* $m + n - 3$ *positive components.*

Proof. Suppose $\mathbf{x}^*(\mathbf{y};\varepsilon)$ has k zero-valued basic variables and let B be an optimal basis. Then, by dropping k degenerate arcs from the spanning tree $G_B = (S, T, A_B)$, we have $k+1$ subtrees, in each of which supply and demand must balance. If $k > 2$, however, there is at least one subtree containing neither s_1, s_2 nor s_3. In such a subtree, the total supply is integral valued but the total demand is not. This is a contradiction. Hence, the number of zero-valued basic variables in $\mathbf{x}^*(\mathbf{y};\varepsilon)$ is at most two. □

Thus, the perturbed problem (5.1) has turned out to satisfy Assumption 3.1. We can easily check that all the lemmas and theorem in Section 3 can be applied to (5.1) except that the vertices of each linearity piece $F_B(\varepsilon)$ of

$$(5.5)\qquad f(\mathbf{y};\varepsilon) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}(\varepsilon) x_{ij}^*(\mathbf{y};\varepsilon)$$

are vectors of not integers but multiples of ε.

5.2. Modification of the algorithm

Let us denote by $\mathcal{F}(\varepsilon)$ the family of $F_B(\varepsilon)$'s and by $\mathcal{G}(\varepsilon) = (\mathcal{V}(\varepsilon), \mathcal{E}(\varepsilon))$ the plane graph associated with $\mathcal{F}(\varepsilon)$. There is the following correspondence between the two graph $\mathcal{G}(\varepsilon)$ and $\mathcal{G}$:

For an arbitrary $\mathbf{v} \in \mathcal{V}(\varepsilon)$, let B be an optimal basis of $[\mathrm{P}(\mathbf{y}; \varepsilon)]$. Dropping the two degenerate arcs from G_B gives three subtrees G_i, with $s_i \in G_i$, $i = 1, 2, 3$. Since $0 < \varepsilon < 1$ and G_i contains no degenerate arcs, the flow on each arc in G_i remains nonnegative if we replace $b_j(\varepsilon)$ by b_j. Hence, B is an optimal basis of $[\mathrm{P}(\mathbf{v}')]$ for $\mathbf{v}' = (v_1 - n_1\varepsilon, v_2 - n_2\varepsilon, v_3 - n_3\varepsilon)^T$, where n_i is the number of terminal nodes in G_i. For $\mathbf{x}^*(\mathbf{v}')$, each of the paths between s_1, s_2 and s_3 in G_B still contains a degenerate arc. We see from Lemma 4.1 that $\mathbf{v}'$ is a vertex of F_B. As will be shown below, this correspondence, denoted by $\phi : \mathcal{V}(\varepsilon) \to \mathcal{V}$, is surjective. Therefore, we can make a complete search of $\mathcal{V}$ by enumerating the vertices in $\mathcal{G}(\varepsilon)$.

Lemma 5.3. *The correspondence ϕ maps $\mathcal{V}(\varepsilon)$ onto $\mathcal{V}$.*

Proof. We first show that for each $\mathbf{v} \in \mathcal{V}$ there is an optimal basis B' of $[\mathrm{P}(\mathbf{v})]$ such that $G_{B'}$ includes an augmenting path from s_1 to each terminal node t_j for $\mathbf{x}^*(\mathbf{v})$. Let B be any optimal basis of $[\mathrm{P}(\mathbf{v})]$ and $\overline{c}_{ij}$ the reduced cost relative to B. If $\mathbf{x}^*(\mathbf{v})$ has k zero-valued basic variables, G_B is decomposed into $k+1$ nondegenerate subtrees. Let G_ℓ denote the subtree, $\ell = 1, \ldots, k+1$, and suppose $s_i \in G_i$ for $i = 1, 2, 3$. Augmenting paths from s_1 to t_j's can be found if we solve a shortest path problem with respect to $\overline{c}_{ij}$ in the graph resulting from G by ignoring arc directions in G_ℓ's (see e.g. [1] for detail). Let $G_{B'}$ be the shortest path tree and $\ell(p)$ the tree path length from s_1 to p. Then

$$\ell(t_j) \leq \ell(s_i) + \overline{c}_{ij}, \quad \forall (s_i, t_j) \in A.$$

Hence, by revising the simplex multipliers α_i, β_j into $\alpha'_i = \alpha_i - \ell(s_i)$ and $\beta'_j = \beta + \ell(t_j)$, we have the optimality of the basis B' corresponding to $G_{B'}$, i.e.,

$$\overline{c}'_{ij} = c_{ij} - \alpha'_i - \beta'_j \geq 0, \quad \forall (s_i, t_j) \in A.$$

For each $\ell = 2, \ldots, k+1$, all the terminal nodes in G_ℓ are now connected from s_1 by augmenting paths in $G_{B'}$; these paths share a tree path $\Pi_{B'}(s_1, t_{j\ell})$ from s_1 to some terminal node $t_{j\ell} \in G_\ell$. When the perturbation is introduced, the shortage of supply in G_ℓ for $\ell \geq 4$ can be covered by the source closest to $t_{j\ell}$ among s_1, s_2 and s_3 in $\Pi_{B'}(s_1, t_{j\ell})$. As a result of this, G_ℓ's are merged into three nondegenerate subtrees $\tilde{G}_i$, with $s_i \in \tilde{G}_i$, $i = 1, 2, 3$. Let n_i be the number of terminal nodes in $\tilde{G}_i$. Then B' is optimal to $[\mathrm{P}(\mathbf{v}'; \varepsilon)]$ for $\mathbf{v}' = (v_1 + n_1\varepsilon, v_2 + n_2\varepsilon, v_3 + n_3\varepsilon)^T \in \mathcal{V}(\varepsilon)$. Since $\mathbf{v} \in \mathcal{V}$ is arbitrary, we conclude that $\phi(\mathcal{V}(\varepsilon)) = \mathcal{V}$. □

When applying the algorithm 3FACTORIES to (5.1), we need to modify the evaluations of f and g. Namely, at the beginning of the procedure SEARCH, we do the following:

determine the nondegenerate subtrees G_i of G, with $s_i \in G_i$, $i = 1, 2, 3$, for $\mathbf{x}^*(\mathbf{v}; \varepsilon)$;

let n_i be the number of terminal nodes in G_i for $i = 1, 2, 3$;

$\mathbf{v}' := (v_1 - n_2\varepsilon, v_2 - n_2\varepsilon, v_3 - n_3\varepsilon)^T$;

if $f(\mathbf{v}') + g(\mathbf{v}') < z^*$ **then** $(\mathbf{x}^*, \mathbf{y}^*) := (\mathbf{x}^*(\mathbf{v}'), \mathbf{v}')$ and $z^* := f(\mathbf{v}') + g(\mathbf{v}')$;

As mentioned before, components of each vertex in $\mathcal{V}(\varepsilon)$ are multiples of ε, so that $|\mathcal{V}(\varepsilon)|$ is bounded by $O((\delta/\varepsilon)^2) = O(n^2\delta^2)$. This leads us to the following time complexity of 3FACTORIES without Assumption 3.1, in the same way as in Theorem 4.2:

Theorem 5.4. *The algorithm 3FACTORIES requires* $O((m + n)n^2\delta^2 + H(m, n))$ *arithmetic operations and* $O(n^2\delta^2)$ *evaluations of* g*, where* $\delta = \sum_{j=1}^{n} b_j - \sum_{i=4}^{m} a_i$ *and* $H(m, n)$ *is the running time needed to solve a Hitchcock problem.*

References

1. R. K. Ahuja, T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, N. J.,1993.
2. M. S. Bazaraa, J. J. Jarvis and H. D. Sherali, *Linear Programming and Network Flows*, John Wiley and Sons, N. Y., 1990.
3. R. E. Erickson, C. L. Monma and A. F. Veinott, *Send-and-split method for minimum-concave-cost network flows*, Mathematics of Operations Research **12** (1987), 634-664.
4. T. Gal, *Postoptimal Analysis, Parametric Programming and Related Topics*, McGraw-Hill N. Y., 1979.
5. G. Gallo, C. Sandi and C. Sodini, *An algorithm for the min concave cost flow problem*, European Journal of Operational Research **4** (1980), 248-255.
6. G. M. Guisewite, *Network Problems*, in R. Horst and P. M. Pardalos (eds.), Hand-

book of Global Optimization, Kluwer Academic Publishers, Dortrech, 1995.

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

8. G. M. Guisewite and P. M. Pardalos, *A polynomial time solvable concave network flow problem*, Network **23** (1993), 143-149.

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

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

11. H. Konno, P. T. Thach and H. Tuy, *Global Optimization: Low Rank Nonconvex Structure*, Kluwer Academic Publishers, Dortrech, 1996.

12. T. Kuno, *Parametric approach for maximum flow problems with an additional reverse convex constraint*, Technical Report ISE-TR-95-128, Institute of Information Sciences and Electronics, University of Tsukuba, Ibaraki, 1995, to appear in Annals of Operations Research.

13. T. Kuno and T. Utsunomiya, *A decomposition algorithm for solving certain classes of production-transportation problems with concave production cost*, Journal of Global Optimization **8** (1996), 67-80.

14. T. Kuno and T. Utsunimiya, *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, Ibaraki, 1995.

15. T. Kuno and T. Utsunomiya, *Minimizing a linear multiplicative-type function under network flow constraints*, Technical Report ISE-TR-95-124, Institute of Information Sciences and Electronics, University of Tsukuba, Ibaraki, 1995, to appear in Operations Research Letters.

16. H. Tuy, *Polyhedral annexation, dualization and dimension reduction technique in global optimization*, Journal of Global Optimization **1** (1991), 229-244.

17. H. Tuy, *The complementary convex structure in global optimization*, Journal of Global Optimization **2** (1992), 21-40.

18. H. Tuy and F. A. Al-Khayyal, *Global optimization of a nonconvex single facility location problem by sequential unconstrained convex minimization*, Journal of Global Optimization **2** (1992), 61-67.

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

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

21. H. Tuy, S. Ghannadan, A. Migdalas and P. Värbrand, *Strongly polynomial algo-*

rithm for a production-transportation problem with a fixed number of nonlinear variables, Preprint, Department of Mathematics, Linköping University, Linköping, 1993, to appear in Mathematical Programming.

22. H. Tuy, 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.

23. H. Tuy, A. Migdalas and P. Värbrand, *A quasiconcave minimization method for solving linear two level programs*, Journal of Global Optimization **4** (1994), 243-264.

24. H. Tuy and B. T. Tam, *An efficient solution method for rank two quasiconcave minimization problems*, Optimization **24** (1992), 43-56.

Institute of Information Sciences and Electronics,
University of Tsukuba