ACTA MATHEMATICA VIETNAMICA
Volume 26, Number 1, 2001, pp. 17-26

A CLASS OF MINIMAX PROBLEMS SOLVABLE IN POLYNOMIAL TIME

TRAN VU THIEU AND TRAN THI HUE

Abstract. We develop a polynomial-time algorithm for solving a class of 0-1 production-transportation problems. The objective function is the maximum of n monotonic functions of the production volume. The transportation cost is assumed to be small as compared to the production cost and is omitted. The proposed algorithm is based on a labeling technique for improving feasible solutions.

1. Introduction

Given an $m \times n$ matrix $A = (a_{ij})_{m\times n}$, where $a_{ij} \in \{0, 1\}$, and given positive integer numbers p_i $(0 < p_i \leq n)$, $i = 1, 2, \ldots, m$, we consider the following optimization problem

$$\text{(P)} \quad f(x) \equiv \max_{1\leq j\leq n} f_j(y_j) \longrightarrow \min \tag{1}$$

subject to

$$\sum_{i=1}^{m} x_{ij} = y_j, \quad j \in N \equiv \{1, 2, \ldots, n\}, \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = p_i, \quad i \in M \equiv \{1, 2, \ldots, m\}, \tag{3}$$

$$x_{ij} \in \{0, 1\} \text{ and } x_{ij} \leq a_{ij} \text{ for all } i \in M \text{ and } j \in N, \tag{4}$$

where for each $j \in N$, $f_j(y_j)$ is a univariate function satisfying

$$f_j(y) \leq f_j(y') \text{ for all } y, y' \in \{0, 1, \ldots, m\},\ y \leq y'. \tag{5}$$

For instance, condition (5) holds if f_j is of the form

$$f_j(y) = \begin{cases} 0 & \text{if } y = 0, \\ c_j y + d_j & \text{if } y > 0, \end{cases}$$

Received July 26, 1999; in revised form October 2, 2000.
1991 *Mathematics Subject Classification.* Primary 90B06, 90C10, 90C30; Secondary 65K05, 49M37.
Key words and phrases. 0-1 production-transportation problem, maximum flow problem, monotonic function, polynomial-time algorithm.
This work was supported in part by the National Science Council of Vietnam.

where c_j, d_j are given positive numbers.

Problem (P) has some applications in scheduling theory and was studied in [2], [6], [8] for the case $f_j(y) \equiv y$ for all $j \in N$. A polynomial-time algorithm is described in [6], which reduces problem (P) with $f_j(y) \equiv y$ to solving a finite number of maximum flow problems. Its running time is $O(\log_2 n \times O_{MF})$, where O_{MF} is the running time of any polynomial-time maximum flow algorithm. For instance, for Edmonds-Karp's (1969) algorithm, O_{MF} is $O(pq^2)$ and for Dinic's (1970) algorithm it is $O(p^2q)$, where p is the number of nodes and q the number of arcs in the network (see [1], [3] for more details). Another class of minimax problems has also been investigated in [7]. To our knowledge, a number of other polynomial-time algorithms for various versions of the convex cost flow problem have been developed, including those of Minoux [4] and [5].

The purpose of this paper is to show that the algorithm developed in [2] for the case $f_j(y) \equiv y$ for all $j \in N$ can be modified to solve (P) with f_j satisfying (5). Basically, the proposed method proceeds according to the same scheme as that presented in [2] with, however, a major improvement in the definition of the full and deficient columns and in the proofs of the main propositions. Moreover, the present method can also be directly applied to problems with f_j being either increasing or decreasing in $[0, m]$.

2. Main results

As usual a matrix $x = \{x_{ij}\}$ whose entries satisfy (3) and (4) is called a *feasible solution* of (P), a feasible solution achieving the minimum of (1) is called an *optimal solution* of (P).

Let

$$a_i = \sum_{j=1}^{n} a_{ij},\ i \in M, \quad b_j = \sum_{i=1}^{m} a_{ij},\ j \in N, \quad p = \sum_{i=1}^{m} p_i > 0.$$

As has been proved in [6], a necessary and sufficient condition for the existence of an optimal solution of (P) is that

$$a_i \geq p_i \quad \text{for all } i \in M. \tag{6}$$

Condition (6) is very simple and easy to check. So, in the sequel we assume that (P) satisfies this condition. It is also natural to suppose that $b_j > 0$ for all $j \in N$.

Since y_j defined by (2) is integral and

$$0 \leq y_j = \sum_{i=1}^{m} x_{ij} \leq \sum_{i=1}^{m} a_{ij} = b_j, \quad j \in N,$$

all the function values $f_j(y_j)$ can be listed as

$$f_1(0), f_1(1), \ldots, f_1(b_1), f_2(0), f_2(1), \ldots, f_2(b_2), \ldots, f_n(0), f_n(1), \ldots, f_n(b_n). \tag{7}$$

Let $f_1 = \max\{f_j(0) : j \in N\}$. Suppose that there are q distinct values in (7) which are greater than or equal to f_1 and these values are arranged in the increasing order as

$$f_1 < f_2 < \cdots < f_q = \max\{f_j(b_j) : j \in N\}. \tag{8}$$

We first observe that the optimal function value of (P), say f^*, is one of the q values in (8).

For the sake of convenience, we associate with each feasible solution $x = \{x_{ij}\}$ of (P) a table consisting of m rows and n columns. The cell at the intersection of row i and column j is denoted by (i, j). Then, $x = \{x_{ij}\}$ will correspond to a table consisting of zeros and ones in its cells. A cell (i, j) is called *black* if $a_{ij} = 0$ ($x_{ij} = 0$ for all black cells (i, j)). The remaining cells will be divided into two categories: *white cells* if $x_{ij} = 0$ and *blue cells* if $x_{ij} = 1$.

Remark 1. A feasible solution of (P) satisfying (6) can be obtained as follows. For each row i, from 1 to m, we write 1 in the non black cells of the row from left to right until a total of p_i ones have been assigned, then we write 0 in the remaining cells of the row.

Consider now a feasible solution $x = \{x_{ij}\}$ of (P). According to (2) we have

$$\sum_{j=1}^{n} y_j = \sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} p_i = p. \tag{9}$$

Column j is called *full* if $f_j(y_j) = f(x)$ and *deficient* if $y_j + 1 \le b_j$ and $f_j(y_j + 1) < f(x)$. The degree of column j with respect to x, denoted by $\rho_j(x)$, is defined to be the number of integers i such that $0 \le i \le y_j$ and $f_j(i) = f(x)$. Obviously, $0 \le \rho_j(x) \le b_j + 1$ for all $j \in N$, and $\rho_j(x) \ge 1$ if column j is full and $\rho_j(x) = 0$ otherwise. We define $\rho(x)$ to be $\sum_{j\in N} \rho_j(x)$ and call $\rho(x)$ the degree of x. Clearly, $\rho(x) \ge 1$ for every feasible solution x, as at least one full column exists. It should be noted that the notions of blue cells, white cells, full columns, deficient columns and degree of a column relate to a given feasible solution.

The following proposition gives a simple criterion for an optimal solution of (P).

Proposition 1. *Let x be a feasible solution of (P). If x has no deficient column, then x is optimal.*

Proof. Assume to the contrary that there exists a feasible solution x' of (P) better than x, i.e. such that

$$f(x') = \max_{j\in N} f_j(y'_j) < f(x) = \max_{j\in N} f_j(y_j) = f_{j_0}(y_{j_0}) \text{ for some } j_0 \in N, \tag{10}$$

where y_j and y'_j are defined by (2) with respect to x and x', respectively. From (10) we have $f_{j_0}(y'_{j_0}) < f_{j_0}(y_{j_0})$. It follows from (5) that $y'_{j_0} < y_{j_0}$. Since $\sum_{j=1}^{n} y_j = \sum_{j=1}^{n} y'_j = p$ by (9), there exists $j_1 \in N \setminus \{j_0\}$ such that $y_{j_1} < y'_{j_1} \le b_{j_1}$, which implies $y_{j_1} + 1 \le y'_{j_1} \le b_{j_1}$, as y_{j_1} and y'_{j_1} are integral. Also, from (5)

$$f_{j_1}(y_{j_1} + 1) \le f_{j_1}(y'_{j_1}) \le f(x') < f(x),$$

and so column j_1 is deficient, contrary to the assumption. $\square$

Let $\mathcal{S}$ be an alternating chain of white and blue cells with respect to x joining column j_0 to column j_k of the form

$$\mathcal{S} = \{(i_0, j_0), (i_0, j_1), \dots, (i_{k-1}, j_{k-1}), (i_{k-1}, j_k)\}, \quad (k \geq 1), \tag{11}$$

where (i_t, j_t), $t = 0, 1, \dots, k-1$, are white cells ($x_{i_t j_t} = 0$), while (i_t, j_{t+1}), $t = 0, 1, \dots, k-1$, blue cells ($x_{i_t j_{t+1}} = 1$). Here all the row indices $i_0, \dots, i_{k-1}$ and all the column indices $j_0, \dots, j_k$ are distinct. Let us introduce the following transformation of x in such a chain.

Transformation A. Change every white cell in the chain to blue one and every blue cell to white one. That is, we set

$$x'_{i_t j_t} = 1,\ x'_{i_t j_{t+1}} = 0,\ t = 0, 1, \dots, k-1,\ x'_{ij} = x_{ij},\ \forall (i,j) \notin \mathcal{S}.$$

Since in each of the rows i_t ($t = 0, 1, \dots, k-1$) there are just one white cell and one blue cell of $\mathcal{S}$, $x' = \{x'_{ij}\}$ satisfies (3), (4), i.e. x' is also a feasible solution of (P).

Proposition 2. *Let x be a feasible solution of* (P). *If there exists an alternating chain of white and blue cells joining a deficient column to a full column, then x can be changed to a new feasible solution x' which is either better or has smaller degree than x.*

Proof. Let $\mathcal{S}$ be a chain of the form (11) joining a deficient column j_0 to a full column j_k. Applying Transformation A in $\mathcal{S}$ yields a new feasible solution x'. Let y_j and y'_j be defined by (2) with respect to x and x' respectively. Since in each of the columns j_t ($t = 1, 2, \dots, k-1$) there are just one white cell and one blue cell of $\mathcal{S}$, we have

$$y'_j = y_j, \quad \text{for all } j \in N \setminus \{j_0, j_k\}. \tag{12}$$

On the other hand, as column j_0 has only one cell of $\mathcal{S}$ (white cell (i_0, j_0)), it is clear that

$$y'_{j_0} \ = \ y_{j_0} + 1, \tag{13}$$

and as column j_k has only one cell of $\mathcal{S}$ (blue cell (i_{k-1}, j_k)), it is clear that

$$y'_{j_k} \ = \ y_{j_k} - 1. \tag{14}$$

As column j_0 is deficient, from (12)-(14) it follows that if $\rho(x) = 1$ (equivalently, if j_k is a unique full column with respect to x and $\rho_{j_k}(x) = 1$), then we have

$$\begin{cases} f_j(y'_j) & = f_j(y_j) < f(x) \text{ for all } j \in N \backslash \{j_0, j_k\}, \\ f_{j_0}(y'_{j_0}) & = f_{j_0}(y_{j_0} + 1) < f(x), \\ f_{j_k}(y'_{j_k}) & = f_{j_k}(y_{j_k} - 1) < f_{j_k}(y_{j_k}) = f(x). \end{cases}$$

This shows that x' is better than the current solution x. In the opposite case, we have $f(x') = f(x)$, i.e. x' is no worse than x, but x' has a lower degree than x (as $\rho_{j_k}(x') = \rho_{j_k}(x) - 1$ and $\rho_j(x') = \rho_j(x)$ for all $j \in N \setminus \{j_k\}$). $\square$

We have another criterion for optimality.

Proposition 3. *Let x be a feasible solution of* (P)*. If there is no alternating chain of white and blue cells joining a deficient column to a full column, then x is an optimal solution of* (P)*.*

Proof. Suppose to the contrary that there is a feasible solution $x' = \{x'_{ij}\}$ of (P) better than $x = \{x_{ij}\}$, i.e.

$$f(x') = \max_{j \in N} f_j(y'_j) < f(x) = \max_{j \in N} f_j(y_j) = f_{j_0}(y_{j_0}) \text{ for some } j_0 \in N, \tag{15}$$

where y_j, y'_j are defined by (2) with respect to x and x' respectively. We show that this leads to a contradiction. Indeed, from (15) we have $f_{j_0}(y'_{j_0}) < f_{j_0}(y_{j_0})$. It follows from (5) that $y'_{j_0} < y_{j_0}$. Hence, by (2) there exists one row $i_0 \in N$ such that $x'_{i_0 j_0} = 0$, $x_{i_0 j_0} = 1$ (i.e. (i_0, j_0) is a blue cell with respect to x). Moreover, as both x and x' satisfy (3) with $i = i_0$, there is one column $j_1 \in N \setminus \{j_0\}$ such that $x'_{i_0 j_1} = 1$, $x_{i_0 j_1} = 0$ (i.e. (i_0, j_1) is a white cell with respect to x). If we still have $y_{j_1} \geq y'_{j_1}$, there exists $i_1 \in M \setminus \{i_0\}$ such that $x'_{i_1 j_1} = 0$, $x_{i_1 j_1} = 1$ (i.e. (i_1, j_1) is a blue cell with respect to x), and also by (3) there must be one column $j_2 \in N \setminus \{j_1\}$ such that $x'_{i_1 j_2} = 1$, $x_{i_1 j_2} = 0$ (i.e. (i_1, j_2) is a white cell with respect to x). If $j_2 \neq j_0$, we continue this process until either of the following cases occurs.

Case A. A column $j_r \in N \setminus \{j_0, \ldots, j_{r-1}\}$ with $y_{j_r} < y'_{j_r}$ is reached. This gives $y_{j_r} + 1 \leq y'_{j_r} \leq b_{j_r}$, as y_{j_r} and y'_{j_r} are integral. Also, from (5) $f_{j_r}(y_{j_r} + 1) \leq f_{j_r}(y'_{j_r}) \leq f(x') < f(x)$, and so column j_r is deficient. Thus, in this case we obtain an alternating chain of white and blue cells of the form

$$(i_{r-1}, j_r), (i_{r-1}, j_{r-1}), \ldots, (i_0, j_1), (i_0, j_0), \quad (r \geq 1)$$

that joins the deficient column j_r to the full column j_0, contrary to the hypothesis of the proposition.

Case B. We obtain a cycle of cells of the form

$$\mathcal{C} = \{(i_s, j_s), (i_s, j_{s+1}), \ldots, (i_t, j_t), (i_t, j_s), (i_s, j_s)\},$$
$$(0 \leq s < t, t \geq 1) \text{ or}$$
$$\mathcal{C} = \{(i_s, j_{s+1}), (i_{s+1}, j_{s+1}), \ldots, (i_{t-1}, j_t), (i_s, j_t), (i_s, j_{s+1})\},$$
$$(0 \leq s < t, t \geq 2),$$

where $x'_{i_u j_u} = 0,\ x'_{i_{u-1} j_u} = 1\ (s \leq u \leq t),\ x'_{i_t j_s} = 1$ or $x'_{i_s j_t} = 0$. Setting

$$\bar{x}_{i_u j_u} = 1,\ \bar{x}_{i_{u-1} j_u} = 0\ (s \leq u \leq t),\ \bar{x}_{i_t j_s} = 0 \text{ or}$$
$$\bar{x}_{i_s j_t} = 1,\ \bar{x}_{ij} = x'_{ij},\ \forall (i,j) \notin \mathcal{C},$$

we get a new feasible solution $\bar{x}$ with $f(\bar{x}) = f(x')$ (because $\bar{y}_j = \sum_{i=1}^{m} \bar{x}_{ij} = y'_j$ for all $j \in N$).

If $\bar{x}$ still differs from x, the above process will be repeated with x' replaced by $\bar{x}$. As the number of components of $\bar{x}$ different from the corresponding components of x decreases by at least four units when Case B occurs, after a finite number of repetitions we must have $\hat{x} = x$ and, at the same time, $f(\hat{x}) = f(x')$, i.e. $f(x) = f(\hat{x}) = f(x')$, which contradicts (15). □

One question now to elucidate is whether there exists an alternating chain of white and blue cells joining a deficient column to a full column, as mentioned in Propositions 2 and 3. In answer to this question, we consider the following procedure for rows and columns labeling.

The rows and columns labeling procedure. First of all, we assign label 0 to each column j which is full ($f_j(y_j) = f(x)$). If column j is labeled, we assign label j to each row i which has not yet been labeled and has $x_{ij} = 1$ ((i, j) is a blue cell). Then, if row i is labeled, we assign label i to each column j which has not yet been labeled and has $a_{ij} - x_{ij} = 1$ (this is equivalent to $a_{ij} = 1$, $x_{ij} = 0$, i.e. (i, j) is a white cell) and so on. The above procedure must stop after at most $m + n$ labelings.

Proposition 4. *An alternating chain of white and blue cells joining a deficient column to a full column exists if and only if there is at least one deficient column that is labeled.*

Proof. Suppose there exists a chain of the form (11) joining deficient column j_0 to full column j_k. We claim that j_0 will be labeled using the above labeling procedure. Indeed, if column j_0 is not labeled row i_0 cannot be labeled either, as (i_0, j_0) is a white cell. Then j_1 cannot be labeled either as (i_0, j_1) is a blue cell, and so on. In the end, j_k cannot be labeled, contrary to the fact that full column j_k was first assigned label 0.

Turning to the proof of sufficiency, suppose that a deficient column, say j_0, is assigned label i_0 ((i_0, j_0) is a white cell) and row i_0 is assigned label $j_1 \neq j_0$ ((i_0, j_1) is a blue cell). Let column j_1 be assigned a label not equal to 0, for instance, $i_1 \neq i_0$ ((i_1, j_1) is a white cell), and row i_1 be assigned label $j_2 \neq j_0, j_1$ ((i_1, j_2) is a blue cell). If column j_2 is assigned a label not equal to 0, we continue searching. As the number of columns is finite (equal to n), eventually we must determine a column $j_k \neq j_t, t = 0, 1, \ldots, k-1$, assigned label 0, i.e. j_k is a full column, and the required chain is

$$\mathcal{S} = \{(i_0, j_0), (i_0, j_1), \ldots, (i_{k-1}, j_{k-1}), (i_{k-1}, j_k)\}, \quad (k \geq 1),$$

where (i_t, j_t), $t = 0, 1, \ldots, k-1$, are white cells, while (i_t, j_{t+1}), $t = 0, 1, \ldots, k-1$, are blue cells. □

3. The polynomial time algorithm for (P)

From the above results we are now in a position to derive an algorithm for solving (P). The algorithm consists of the following steps.

Step 1 (Initialization). Find an initial feasible solution x^1 of (P) (see Remark 1). Set $k = 1$ and go to Step 2.

Step 2 (Test for optimality). Determine the full columns and the deficient columns with respect to x^k. If no deficient column exists, x^k is an optimal solution of (P) (by Proposition 1). Otherwise, perform the rows and columns labeling as described in Section 2. If there is no deficient column that is labeled then x^k is also optimal (by Propositions 3 and 4). If a deficient column is labeled, a chain of the form (11) is discovered, joining a deficient column to a full column (by Proposition 4). Go to Step 3.

Step 3 (Solution improvement). By applying Transformation A in the chain found in Step 2, obtain a new feasible solution x' which is better or has a lower degree than x^k (by Proposition 2). Set $x^{k+1} = x'$ and $k \leftarrow k + 1$, then return to Step 2.

Proposition 5. *The above algorithm terminates after a finite number of steps.*

Proof. After each improvement in Step 3, either a better feasible solution or a solution with a lower degree than the previous one is obtained. Since the objective function of the problem can take on only a finite number of values (see (7), (8)) while the degree of each feasible solution of the problem is positive and bounded by $m \times n$, the algorithm cannot be infinite. □

Complexity of the algorithm. In order to bound the running time of the algorithm, we evaluate the number of arithmetic operations needed in each step of the algorithm in the worst case.

Step 1. An initial feasible solution and the corresponding values y_j^1 $(j \in N)$ can be computed in $O(m \times n)$ arithmetic operations.

Step 2. As shown in (12)-(14), the time needed to update y_j^k $(j \in N)$ and $f(x^k)$ is at most $O(m + n)$. Determining the full columns and the deficient columns requires $O(n)$ arithmetic operations. Row and column labeling can be performed in $O(m \times n)$ arithmetic operations. The operation of searching for an alternating chain of white and blue cells joining a deficient column to a full column requires $O(m + n)$ arithmetic operations (using labels assigned to the rows and columns). In all, Step 2 requires $O(m \times n)$ arithmetic operations in the worst case.

Step 3. The solution improvement in a chain obtained in Step 2 requires $O(m+n)$ arithmetic operations, because there are at most $(m + n)$ cells in such a chain.

Steps 2 and 3 are repeated several times. After each repetition either the objective value or the degree of the current feasible solution is reduced. Since the objective function (1) can take on at most $m \times n$ different values and the degree of a feasible solution is bounded by $m \times n$, the number of repetitions is bounded by $O(m^2 \times n^2)$. Consequently, the algorithm requires $O((m \times n)(m^2 \times n^2))$ or $O(m^3 n^3)$ arithmetic operations.

As for the time needed to compute the function values f_j, after having x^1 we need $2n$ evaluations of f_j to obtain $f_j(y_j^1)$ and $f_j(y_j^1 + 1)$ for all $j \in N$. In order to perform Step 2 we have to compute only two new function values $f_{j_0}(y_{j_0}+2)$ and $f_{j_k}(y_{j_k}-1)$. Since Step 2 is repeated at most $O(m^2n^2)$ times, the total number of evaluations of f_j is about $O(m^2n^2)$.

We have thus established the following result.

Proposition 6. *Problem (P) can be solved in $O(m^3n^3)$ arithmetic operations and $O(m^2n^2)$ evaluations of the function f_j.*

Assuming that an evaluation of the function f_j can be done in a unit time, the running time of the algorithm is $O(m^3n^3)$. When $m \approx n$ it is $O(n^6)$.

Example. Solve problem (P) with $m = 4$, $n = 5$, $p_1 = 2$, $p_2 = 3$, $p_3 = 3$, $p_4 = 2$ and

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

The functions $f_j(.)$ are given by

$$f_1(y) = -1.2 + 0.6y,\ f_2(y) = 1.8 + 0.2y,\ f_3(y) = -0.5 + 0.8y,$$

$$f_4(y) = 1.5 + 0.4y,\ f_5(y) = 1.3 + 0.7y.$$

Summing up the elements of A in each row and each column yields

$$a_1 = a_2 = a_3 = 4,\ a_4 = 3;\ b_1 = b_2 = b_3 = b_4 = b_5 = 3 \text{ and } p = 10.$$

Step1. Since $0 \le y_j \le b_j = 3$ for all $j = 1, \ldots, 5$, we have

$$\begin{cases} k & = & 0 & 1 & 2 & 3 \\ f_1(k) & = & -1.2 & -0.6 & 0.0 & 0.6 \\ f_2(k) & = & 1.8 & 2.0 & 2.2 & 2.4 \\ f_3(k) & = & -0.5 & 0.3 & 1.1 & 1.9 \\ f_4(k) & = & 1.5 & 1.9 & 2.3 & 2.7 \\ f_5(k) & = & 1.3 & 2.0 & 2.7 & 3.4 \end{cases}$$

At the completion of Step 1 of the algorithm, we obtain an initial feasible solution of (P)

$$x^1 = \begin{pmatrix} 1 & 1 & \times & 0 & 0 \\ 1 & \times & 1 & 1 & 0 \\ \times & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & \times & \times \end{pmatrix} \begin{matrix} 2 \\ \\ 2 \\ 2 \end{matrix}$$
$$\begin{matrix} & 0 & & 1 & 1 \end{matrix}$$

(black cells are marked by $\times$, the last column indicates the labels assigned to the rows and the last row the labels assigned to the columns).

Step 2. Summing up the elements in each column of x^1 yields

$$y_1^1 = y_2^1 = 3, \quad y_3^1 = y_4^1 = 2, \quad y_5^1 = 0,$$

$$f^1 = \max\{0.6,\ 2.4,\ 1.1,\ 2.3,\ 1.3\} = 2.4.$$

Column 2 is full. Columns 3, 5 are deficient. Column 2 is first labeled with a 0. We search column 2 in x^1 for a 1 (blue cell) and find it in rows 1, 3, 4, so these rows are labeled with a 2 (subscript of column 2). We now search labeled row 1 for a 0 (white cell) and find it in columns 4 and 5 (not yet labeled), so these columns are labeled with a 1 (subscript of row 1). At this point, deficient column 5 is labeled with a 1 (row 1), row 1 is labeled with a 2 (column 2). Column 2 is full. Thus, we obtain the chain of cells: (1,5)-(1,2) joining deficient column 5 to full column 2.

Step 3. Changing x^1 in the chain just found in Step 2 gives a new feasible solution

$$x^2 = \begin{pmatrix} 1 & 0 & \times & 0 & 1 \\ 1 & \times & 1 & 1 & 0 \\ \times & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & \times & \times \end{pmatrix} \begin{matrix} 5 \\ 4 \\ 4 \\ 2 \end{matrix}$$
$$\begin{matrix} 1 & 4 & 0 & 2 \end{matrix}$$

First return to Step 2. Summing up the elements in each column of x^2 yields

$$y_1^2 = 3, \quad y_2^2 = y_3^2 = y_4^2 = 2, \quad y_5^2 = 1 \text{ and}$$
$$f^2 = \max\{0.6,\ 2.2,\ 1.1,\ 2.3,\ 2.0\} = 2.3.$$

Column 4 is full. Column 3 is deficient. The labeling procedure now gives the chain of cells: $(4,3)-(4,2)-(1,2)-(1,5)-(2,5)-(2,4)$ joining deficient column 3 to full column 4.

Step 3. Changing x^2 in this new chain gives a new feasible solution

$$x^3 = \begin{pmatrix} 1 & 1 & \times & 0 & 0 \\ 1 & \times & 1 & 0 & 1 \\ \times & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & \times & \times \end{pmatrix}.$$

Second return to Step 2. Summing up the elements in each column of x^3 yields

$$y_1^3 = 3, \quad y_2^3 = 2, \quad y_3^3 = 3, \quad y_4^3 = y_5^3 = 1 \text{ and}$$
$$f^3 = \max\{0.6,\ 2.2,\ 1.9,\ 1.9,\ 2.0\} = 2.2.$$

Now column 2 is full, but there is no deficient column, so x^3 (with 0 instead of $\times$) is an optimal solution. The optimal function value is $f^* = f^3 = 2.2$.

Remark 2. Direct computation shows that one of the optimal continuous solutions of (P) $(0 \le x_{ij} \le 1)$ is

$$x_{opt} = \begin{pmatrix} 1 & 0.42 & 0 & 0.46 & 0.12 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

with the optimal value

$$\begin{aligned} f_{opt} &= \max\{f_1(3), f_2(1.42), f_3(3), f_4(1.46), f_5(1.12)\} \\ &= \max\{0.6,\ 2.084,\ 1.9,\ 2.084,\ 2.084\} \\ &= 2.084 \ < \ f^* \ = \ 2.2. \end{aligned}$$

Remark 3. The above algorithm can also be extended to the case where f_j is either increasing or decreasing in $[0, m]$. The details are left to the reader.

Acknowledgement

The authors wish to thank the anonymous referee for many suggestions that have helped to substantially improve the paper.

References

[1] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, 1993.
[2] V. V. T. Dung and T. V. Thieu, *A finite algorithm for a class of nonlinear optimization problems*, VNU. Journal of Science Nat. Sci. **15** (1) (1999), 6-15.
[3] S. Khuller, Y. J. Sussman and W. Gasarch, *Advanced Algorithms*, Lectures CMSC 858K, 1997.
[4] M. Minoux, *A polynomial algorithm for minimum quadratic cost flow problems*, European J. Oper. Res. **18** (1984), 377-387.
[5] M. Minoux, *Solving integer minimum cost flows with separable convex cost objective polynomially*, Math. Proc. Study **26** (1986), 237-239.
[6] N. D. Nghia and V. V. T. Dung, *A polynomial-time algorithm for solving a class of discrete optimization problems*, Journal of Computer Science and Cybernetics **15** (1) (1999), 8-13 (in Vietnamese).
[7] H. Tuy, *On a class of minimax problems*, Kibernetika **2** (1971), 115-118 (in Russian).
[8] N. H. Xuong. *Mathematiques Discretes et Informatiques*, Masson, Paris 1992.

Hanoi Institute of Mathematics
P.O. Box 631, Boho, Hanoi, Vietnam

Departement of Mathemathics
Faculty of Science, University of Hue
Hue, Vietnam