Journal of the Operations Research Society of Japan
Vol. 23, No. 1, March 1980

APPROXIMATE ALGORITHMS FOR THE MULTIPLE-CHOICE CONTINUOUS KNAPSACK PROBLEMS

Toshihide Ibaraki
Kyoto University

(Received November 2, 1978)

Abstract The multiple-choice continuous knapsack problem is defined as follows: maximize $z = \Sigma_{i=1}^{n} \Sigma_{j=1}^{m_i} c_{ij}x_{ij}$ subject to (1) $\Sigma_{i=1}^{n} \Sigma_{j=1}^{m_i} a_{ij}x_{ij} \leqslant b$, (2) $0 \leqslant x_{ij} \leqslant 1$, $i = 1, 2, \ldots, n$, $j = 1. 2, \ldots, m_i$, (3) at most one of x_{ij} ($j = 1, 2, \ldots, m_i$) is positive for each $i = 1, 2, \ldots, n$, where n, m_i are positive integers and a_{ij} are nonnegative integers. In this paper, it is proved that this problem is NP-complete (which strongly suggests the computational intractability to obtain exact optimal solutions). Then, starting with an approximate solution obtained from the LP optimal solution by rounding, two approximate algorithms are proposed and analyzed. The first one (called the breadth-1 search method) obtains an approximate solution with the (worst case) relative error less than 25%. The required computation time is $0(mN \log N)$, where $N = \Sigma_{i=1}^{n} m_i$ and $m = \max m_i$. The second one (called the breadth-K search method) obtains an approximate solution within an arbitrarily specified (worst case) error bound $\epsilon \times 100\%$ in $0(m^{\lceil 1/4(\epsilon-\epsilon^2) \rceil} N^{\lceil 1/4(\epsilon-\epsilon^2)\rceil})$ time.

1. Introduction

The *multiple-choice continuous knapsack problem* P is described as follows:

(1.1A) $\quad P$: maximize $z = \sum_{i=1}^{n} \sum_{j=1}^{m_i} c_{ij} x_{ij}$

(1.1B) $\quad$ subject to $\sum_{i=1}^{n} \sum_{j=1}^{m_i} a_{ij} x_{ij} \le b$

(1.1C) $\quad 0 \le x_{ij} \le 1$, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m_i$

(1.1D) $\quad$ At most one of $x_{i1}, x_{i2}, \ldots, x_{im_i}$ is positive for $i = 1, 2, \ldots, n$,

where n, m_i ($i = 1, 2, \ldots, n$) are positive integers, and a_{ij} ($i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m_i$) are nonnegative integers. Let

(1.2) $N=\sum_{i=1}^{n}m_i$, $m=\max\{m_i \mid i=1, 2,\dots, n\}$.

This problem was first studied in [6]. One exact algorithm based on branch-and-bound principle and two approximate algorithms (one of them, breadth-1 search method, was only outlined in [6] and its details are described in this paper) were therein proposed and tested. Their computational results are quite encouraging. For example, exact optimal solutions for problems with $n=500$ and $m_i=2$ ($i=1, 2,\dots, n$) are obtained in less than 0.2 seconds (on FACOM M190). Approximate solutions within 0.001% of the optimal values are also obtained in less than one second (on FACOM 230/60) for randomly generated problems with $n=1000$ and $m_i=2$ ($i=1, 2,\dots, n$).

Generally speaking, however, (1.1) is still a difficult problem. It is reported in [6] that some specially structured problems require long computation time to obtain exact optimal solutions. In fact, it is proved to be NP-complete in Section 2 of this paper (for the implication of NP-completeness, see [1, 8]). Therefore, it is desirable to have approximate algorithms, which run in polynomial time and yet can theoretically guarantee the quality of the obtained solutions.

This paper considers two such approximate algorithms. The first is called *breadth-1 search* algorithm, and is described in Sections 4 and 5. It obtains in $O(mN\log N)$ time an approximate solution with the ratio of its objective value and the exact optimal value greater than 3/4. This is then extended in Sections 6 and 7 to a *breadth-K search* algorithm, which is a polynomial time approximation scheme (in the sense of [4]) rather than an algorithm. It obtains an approximate solution with the ratio of its objective value and the exact optimal value greater than $1-\varepsilon$ for any given $0<\varepsilon<1/4$, if an appropriate K is chosen. The time required for this computation is $O(m^{\lceil 1/4(\varepsilon-\varepsilon^2)\rceil} N^{\lceil 1/4(\varepsilon-\varepsilon^2)\rceil})$, where $\lceil x\rceil$ denotes the smallest integer not smaller than x.

The discrete version of (1.1), i.e., P with (1.1C) replaced by

(1.3) $x_{ij}=0$ or 1, $i=1, 2,\dots, n$ and $j=1, 2,\dots, m_i$

has also been studied in various papers such as [3, 11, 12]. Some properties proved in these papers can also be extended to our continuous problems. [3, 11] discuss approximate algorithms for the discrete problem. Some discussion will be given in Section 8 on the use of such algorithms of the discrete version for our problem.

In [3, 6], it is noted that the following assumptions can be made without loss of generality.

(1.4) $$a_{i1} \geq a_{i2} \geq \dots \geq a_{im_i} \geq 0, \quad i = 1, 2, \dots, n,$$

(1.5) $$c_{i1} \geq c_{i2} \geq \dots \geq c_{im_i} > 0 \quad i = 1, 2, \dots, n,$$

(1.6) $$\frac{c_{im_i}}{a_{im_i}} > \frac{c_{im_i-1}}{a_{im_i-1}} > \dots > \frac{c_{i1}}{a_{i1}} > 0, \quad i = 1, 2, \dots, n,$$

(1.7) $$\sum_{i=1}^{n} a_{i1} > b.$$

Here the denominators of (1.6) are allowed to be 0 with the convention that

(1.8)
$$\frac{c}{0} > \frac{c'}{a'} \text{ for any } a', c', c > 0,$$
$$\frac{c}{0} \geq \frac{c'}{0} \text{ if and only if } c \geq c' \text{ for any } c, c' > 0.$$

The following property proved in [6] is also helpful to understand the structure of P.

(1.9) P has an optimal solution such that at most one variable x_{ij} satisfies $0 < x_{ij} < 1$.

2. NP-Completeness of P

The NP-completeness of P defined in (1.1) is proved from the NP-completeness of the following ordinary 0-1 knapsack problem (e.g., [1, 8]):

(2.1)
$$Q\text{: maximize } \sum_{i=1}^{n} \gamma_i x_i$$
$$\text{subject to } \sum_{i=1}^{n} \alpha_i x_i \leq \beta$$
$$x_i = 0 \text{ or } 1, \quad i = 1, 2, \dots, n,$$

where γ_i $(i = 1, 2, \dots, n)$, α_i $(i = 1, 2, \dots, n)$ and β are given positive integers. We associate the following multiple-choice continuous knapsack problem P_Q with the above Q.

(2.2)
$$P_Q\text{: maximize } \sum_{i=1}^{n} (R + \gamma_i) x_{i1} + \sum_{i=1}^{n} R x_{i2}$$
$$\text{subject to } \sum_{i=1}^{n} \alpha_i x_{i1} \leq \beta$$
$$0 \leq x_{ij} \leq 1, \quad i = 1, 2, \dots, n \text{ and } j = 1, 2$$
$$x_{i1} x_{i2} = 0, \quad i = 1, 2, \dots, n,$$

where $m_i = 2$ for $i = 1, 2, \dots, n$, and R is a positive integer.

Lemma 2.1. Let $\gamma = \max_i \gamma_i$, $\alpha = \max_i \alpha_i$ and $R > \gamma(\alpha - 1)$. Then P_Q has an optimal solution $x^0 = (x^0_{ij})$ such that $x^0_{ij} = 0$ or 1 for $i = 1, 2, \ldots, n$ and $j = 1, 2$.

Proof: Let x^0 be an optimal solution of P_Q satisfying property (1.9), i.e.,

$$0 < x^0_{i_1 1} \leq 1 \text{ and } x^0_{ij} = 0 \text{ or } 1 \text{ for every } i \neq i_1 \text{ and } j = 1, 2.$$

(The case of $0 < x^0_{i_1 2} \leq 1$ can be similarly treated.) Then $x^0_{i_1 1}$ satisfies

$$\alpha_{i_1} x^0_{i_1 1} = b - \textstyle\sum_{i \neq i_1} \alpha_i x^0_{i1} (\equiv b') \tag{2.3}$$

Since b' is an integer, $x^0_{i_1 1}$ is equal to one of the followings.

$$\frac{1}{\alpha_{i_1}}, \frac{2}{\alpha_{i_1}}, \ldots, \frac{\alpha_{i_1}}{\alpha_{i_1}} \tag{2.4}$$

Now let x' be defined by

$$x'_{i_1 1} = 0, \; x'_{i_1 2} = 1, \text{ and } x'_{ij} = x^0_{ij} \text{ for } i \neq j_1 \text{ and } j = 1, 2.$$

x' is obviously feasible, and the optimality of x^0 implies

$$(R + \gamma_{i_1}) x^0_{i_1 1} \geq R,$$

namely

$$x^0_{i_1 1} \geq \frac{R}{R+\gamma_{i_1}} \geq \frac{R}{R+\gamma} > \frac{\gamma(\alpha-1)}{\gamma(\alpha-1)+\gamma} = \frac{\alpha-1}{\alpha} \geq \frac{\alpha_{i_1} - 1}{\alpha_{i_1}} .$$

This implies $x^0_{i_1 1} = \alpha_{i_1}/\alpha_{i_1} = 1$ by (2.4). □

Lemma 2.2. Let $R > \gamma(\alpha-1)$ hold in P_Q. Then a solution $x^0 = (x^0_{ij})$ of P_Q satisfying $x_{ij} = 0$ or 1 ($i = 1, 2, \ldots, n$ and $j = 1, 2$) is optimal if and only if $(x^0_1, x^0_2, \ldots, x^0_n)$ defined by $x^0_i = x^0_{i1}$ ($i = 1, 2, \ldots, n$) is an optimal solution of Q.

Proof: Note first that $x^0_{i1} + x^0_{i2} = 1$ ($i = 1, 2, \ldots, n$) can be assumed without loss of generality since x^0_{i2} can be set to 1 without affecting the feasibility if $x^0_{i1} = 0$. Then we have

$$\begin{aligned} &\textstyle\sum_{i=1}^n (R + \gamma_i) x^0_{i1} + \sum_{i=1}^n R x^0_{i2} \\ &= nR + \textstyle\sum_{i=1}^n \gamma_i x^0_{i1} \\ &= nR + \textstyle\sum_{i=1}^n \gamma_i x^0_i \quad (\text{where } x^0_i = x^0_{i1},\; i = 1, 2, \ldots, n). \end{aligned}$$

Thus two objective functions in Q and P_Q differ only by a constant nR. Since both Q and P_Q have the same constraint, and P_Q has an optimal solution with

$x_{ij}=0$ or 1 ($i=1, 2,\ldots, n$, $j=1, 2$) by Lemma 2.1, this completes the proof. □

Theorem 2.3. The multiple-choice continuous knapsack problem P, defined in (1.1), is NP-complete. (See [1, 8] for the precise definition of the NP-completeness.)

Proof: Follows from Lemmas 2.1 and 2.2, since $P \in NP$ is obvious and the 0-1 knapsack problem Q, which is known to be NP-complete, is transformed to a special case of P (i.e., P is NP-hard). □

However, P is not NP-complete in the strong sense (defined in [4]). This is because a pseudo polynomial algorithm for P can be constructed in a manner similar to the ordinary 0-1 knapsack problem by making use of dynamic programming.

3. LP Relaxation of P and its Dual

In order to develop approximate algorithms in the subsequent sections, we introduce the LP (linear Programming) relaxation $\overline{P}$ of P and its dual problem. We list here only necessary results; see [6] for the detailed discussion.

$$(3.1)\qquad \begin{aligned} \overline{P}:\ & \text{maximize } z=\textstyle\sum_{i=1}^{n}\sum_{j=1}^{m_i} c_{ij}x_{ij} \\ & \text{subject to } \textstyle\sum_{i=1}^{n}\sum_{j=1}^{m_i} c_{ij}x_{ij} \le b \\ & \qquad \textstyle\sum_{j=1}^{m_i} x_{ij} \le 1,\ i=1, 2,\ldots, n \\ & \qquad x_{ij}\ge 0,\ i=1, 2,\ldots, n \text{ and } j=1, 2,\ldots, m_i. \end{aligned}$$

The LP dual of $\overline{P}$ is as follows.

$$(3.2)\qquad \begin{aligned} \overline{D}:\ & \text{minimize } v=\textstyle\sum_{i=1}^{n}\mu_i + b\lambda \\ & \text{subject to } \mu_i \ge c_{ij}-a_{ij}\lambda,\ i=1, 2,\ldots, n \text{ and } j=1, 2,\ldots, m_i \\ & \qquad \mu_i\ge 0,\ i=1, 2,\ldots, n \\ & \qquad \lambda \ge 0. \end{aligned}$$

For a given λ, the optimal value $\overline{v}(\lambda)$ of $\overline{D}$ is trivially given by

$$(3.3)\qquad \begin{aligned} & \overline{v}(\lambda)=\textstyle\sum_{i=1}^{n}\overline{\mu}_i(\lambda)+b\lambda \\ & \overline{\mu}_i(\lambda)=\max[0, \max\{c_{ij}-a_{ij}\lambda \mid j=1, 2,\ldots, m_i\}],\ i=1, 2,\ldots, n. \end{aligned}$$

Thus $\overline{D}$ is solved by finding a λ that minimizes $\overline{v}(\lambda)$.

From (3.3), we see that an index j_ℓ is relevant to determining $\bar{\mu}_i(\lambda)$ only if

$$c_{ij_\ell} - a_{ij_\ell}\lambda > c_{ij} - a_{ij}\lambda, \quad j \neq j_\ell \tag{3.4}$$

holds for some $\lambda \geq 0$. Such an index j_ℓ is called *dominant* and the set of dominant indices for each i is denoted by

$$J_i = \{j_1, j_2, \ldots, j_{d_i}\}, \tag{3.5}$$

where $j_1 = 1$ and $j_{d_i} = m_i$ always hold by (1.5) and (1.6). The following lemma is proved in Appendix A.

Lemma 3.1. J_i for a given P is computed in $O(m_i)$ time for each $i = 1, 2, \ldots, n$. Thus the total computation time for all i is $O(N)$. □

It is known [3, 6] that indices in J_i satisfy

$$0 \leq \rho_i(j_1, j_2) < \rho_i(j_2, j_3) < \ldots < \rho_i(j_{d_i-1}, j_{d_i}) < \frac{c_{im_i}}{a_{im_i}} \tag{3.6}$$

for $i = 1, 2, \ldots, n$, where

$$\rho_i(p, q) \triangleq \frac{c_{ip} - c_{iq}}{a_{ip} - a_{iq}} . \tag{3.7}$$

For notational simplicity, however, we assume in the subsequent discussion (unless otherwise stated) that

$$J_i = \{1, 2, \ldots, m_i\}, \quad i = 1, 2, \ldots, n \tag{3.8}$$

and hence

$$0 \leq \rho_i(1, 2) < \rho_i(2, 3) < \ldots < \rho_i(m_i-1, m_i) < \frac{c_{im_i}}{a_{im_i}} . \tag{3.9}$$

Now let

$$L_i = \{\rho_i(1, 2), \rho_i(2, 3), \ldots, \rho_i(m_i-1, m_i), c_{im_i}/a_{im_i}\}, \tag{3.10}$$

$$L = \bigcup_{i=1}^{n} L_i = \{\beta_1, \beta_2, \ldots, \beta_N\}. \tag{3.11}$$

where

$$\beta_1 \leq \beta_2 \leq \ldots \leq \beta_N. \tag{3.12}$$

(Some of β_k are possibly ∞; the order of these β_k's is determined by following the convention (1.8).) Optimal solutions of $\bar{D}$ and $\bar{P}$ are respectively obtained by the following algorithm [6].

The algorithm starts with $\lambda = \infty$ (for $\bar{D}$) and $x = (00\ldots0)$ (for $\bar{P}$), and sweeps list L *from right to left* in the order $\beta_N, \beta_{N-1}, \ldots, \beta_1$. Upon crossing

β_k in this process, λ is set to β_k and x is modified as follows.

(a) x_{im_i} is changed from 0 to 1 if $\beta_k = c_{im_i}/a_{im_i}$.

(b) x_{ij} is set to 0 but x_{ij-1} is set to 1 if $\beta_k = \rho_i(j-1, j)$.

Index j_i memorizes j satisfying $x_{ij} = 1$ for each i. If the new solution x does not satisfy constraint (1.1B) for $\beta_k = \beta_{\bar{k}}$ (this is checked by v'), the computation terminates.

Algorithm DUAL(P)

D1. Obtain L as shown in (3.11) and (3.12).

D2. $v' \leftarrow b$, $j_i \leftarrow \infty$ for $i = 1, 2, \ldots, n$, $k \leftarrow N$.

D3.
$$v' \leftarrow \begin{cases} v' - a_{im_i} & \text{if } \beta_k = c_{im_i}/a_{im_i} \\ v' - (a_{ij-1} - a_{ij}) & \text{if } \beta_k = \rho_i(j-1, j), \end{cases}$$
$$j_i \leftarrow \begin{cases} m_i & \text{if } \beta_k = c_{im_i}/a_{im_i} \\ j-1 & \text{if } \beta_k = \rho_i(j-1, j). \end{cases}$$

D4.† If $v' \leq 0$, go to D5; else return to D3 after letting $k \leftarrow k-1$.

D5. $\bar{k} \leftarrow k$, $\bar{\lambda} \leftarrow \beta_{\bar{k}}$,
$$\bar{e} \leftarrow \begin{cases} v' + a_{im_i} & \text{if } \beta_{\bar{k}} = c_{im_i}/a_{im_i} \\ v' + a_{ij-1} & \text{if } \beta_{\bar{k}} = \rho_i(j-1, j), \end{cases}$$

and halt. □

The computed $k = \bar{k}$ defines $\bar{\lambda} = \beta_{\bar{k}}$ that minimizes $\bar{v}(\lambda)$ of (3.3), i.e., $\bar{v}(\bar{\lambda})$ is the optimal value of $\bar{D}$. Denote j_i computed by DUAL(P) by $j_i(\bar{k})$. Let $p(i)$ denote the smallest $p \geq \bar{k}$ for each i, with $\beta_p = \rho_i(j-1, j)$ or c_{im_i}/a_{im_i}. Then $j_i(\bar{k})$ and $\bar{e}$ satisfy

$$j_i(\bar{k}) = m_i \quad \text{if } \beta_{p(i)} = c_{im_i}/a_{im_i}$$
$$(3.13) \qquad = j-1 \quad \text{if } \beta_{p(i)} = \rho_i(j-1, j)$$
$$= \infty \quad \text{if } p(i) \text{ does not exist.}$$

$$(3.14) \qquad \bar{e} = b - \textstyle\sum_{i \neq \bar{i}} a_{ij_i}(\bar{k}), \text{ where } \bar{i} \text{ satisfies } \beta_{\bar{k}} = c_{\bar{i}m_{\bar{i}}}/a_{\bar{i}m_{\bar{i}}} \text{ or } \beta_{\bar{k}} = \rho_{\bar{i}}(\bar{j}-1, \bar{j}).$$

An optimal solution $\bar{x} = (\bar{x}_{11}, \bar{x}_{12}, \ldots, \bar{x}_{nm_n})$ of $\bar{P}$ is given by

---

† By assumption (1.7), DUAL(P) always satisfies $v' \leq 0$ in D4 before k is set to 0. If DUAL(P) is applied to a problem not satisfying (1.7), k is set to 0 in D4. In this case, we can conclude that
$$x_{i1} = x_{21} = \ldots x_{n1} = 1$$
$$x_{ij} = 0 \text{ for other } i \text{ and } j$$
is an optimal solution of P as well as $\bar{P}$ (see [6]).

(i) Case of $\beta_{\bar{k}} = c_{\bar{i}m_{\bar{i}}}/a_{\bar{i}m_{\bar{i}}}$:

$$(3.15)\quad \begin{aligned} &\bar{x}_{ij_i(\bar{k})} = 1 \text{ for } i \neq \bar{i} \text{ such that } j_i(\bar{k}) < \infty \\ &\bar{x}_{\bar{i}m_{\bar{i}}} = \bar{e}/a_{\bar{i}m_{\bar{i}}} \\ &\bar{x}_{ij} = 0 \text{ for all other } i \text{ and } j \\ &\bar{z} = \textstyle\sum_{i=1}^{n}\sum_{j=1}^{m_i} c_{ij}\bar{x}_{ij} = \sum_{i\neq\bar{i}} c_{ij_i(\bar{k})} + c_{\bar{i}m_{\bar{i}}}\bar{e}/a_{\bar{i}m_{\bar{i}}} \end{aligned}$$

(ii) Case of $\beta_{\bar{k}} = \rho_{\bar{i}}(\bar{j}-1, \bar{j})$:

$$(3.16)\quad \begin{aligned} &\bar{x}_{ij_i(\bar{k})} = 1 \text{ for } i \neq \bar{i} \text{ such that } j_i(\bar{k}) < \infty \\ &\bar{x}_{\bar{i}\bar{j}} = \theta,\ \bar{x}_{\bar{i}\bar{j}-1} = 1-\theta \\ &\bar{x}_{ij} = 0 \text{ for other } i \text{ and } j \\ &\bar{z} = \textstyle\sum_{i=1}^{n}\sum_{j=1}^{m_i} c_{ij}\bar{x}_{ij} = \sum_{i\neq\bar{i}} c_{ij_i(k)} + \theta c_{\bar{i}\bar{j}} + (1-\theta)c_{\bar{i}\bar{j}-1}, \end{aligned}$$

where

$$(3.17)\quad \theta = (a_{\bar{i}\bar{j}-1} - \bar{e})/(a_{\bar{i}\bar{j}-1} - a_{\bar{i}\bar{j}}).$$

Obviously, computation of sets of dominant indices J_i (see (3.5)), execution of DUAL(P) and computation of (3.15) or (3.16) together requires $O(N\log N)$ time (computation of $J_i (i = 1, 2, \ldots, n)$ requires $O(N)$ (Lemma 3.1), D1 of DUAL (P) requires $O(N\log N)$ for sorting $\beta_1, \beta_2, \ldots, \beta_N$, and the rest of DUAL($P$) plus (3.15) or (3.16) requires $O(N)$). If $\bar{x}$ is obtained by (3.15), it satisfies condition (1.1D) and hence is also an optimal solution of P. On the other hand, if $\bar{x}$ is obtained by (3.16), (1.1D) is not satisfied for exactly one index $i = \bar{i}$ unless $\theta = 0$ holds. In case of $\theta > 0$, two variables $x_{\bar{i}\bar{j}-1}$ and $x_{\bar{i}\bar{j}}$ assume positive values.

Remark. The $O(N\log N)$ computation time given above can be reduced to $O(N)$ by employing the fast median finding algorithm in such a manner as used in [2, 11]. The details are not given here, however, since the time requirement of this portion is dominated by others in the approximate algorithms described in the subsequent sections.

4. Approximate Solutions by Rounding

Based on the LP optimal solution $\bar{x}$ obtained by (3.16) (if (3.15) is the case, P is solved), we here construct three approximate solutions of P, $x^{(1)}$, $x^{(2)}$ and $x^{(3)}$. These will be used in the approximate algorithms discussed in Section 5-7.

$x^{(1)}$ and $x^{(2)}$ are given as follows.

(4.1) $$x_{ij}^{(1)} = \bar{x}_{ij} \text{ for } i \neq \bar{i} \text{ and } j = 1, 2, \ldots, m_i$$
$$x_{\bar{i}\bar{j}}^{(1)} = 1,\ x_{\bar{i}j}^{(1)} = 0 \text{ for } j \neq \bar{j}.$$

(4.2) $$x_{ij}^{(2)} = \bar{x}_{ij} \text{ for } i \neq \bar{i} \text{ and } j = 1, 2, \ldots, m_i$$
$$x_{\bar{i}\bar{j}-1}^{(2)} = \bar{e}/a_{\bar{i}\bar{j}-1},\ x_{\bar{i}j}^{(2)} = 0 \text{ for } j \neq \bar{j} - 1.$$

The objective values of these solutions are respectively given by

(4.3) $$z^{(1)} = \sum\sum c_{ij} x_{ij}^{(1)} = \sum_{i \neq \bar{i}} c_{ij_i(\bar{k})} + c_{\bar{i}\bar{j}}$$
$$z^{(2)} = \sum\sum c_{ij} x_{ij}^{(2)} = \sum_{i \neq \bar{i}} c_{ij_i(\bar{k})} + \bar{e}(c_{\bar{i}\bar{j}-1}/a_{\bar{i}\bar{j}-1}).$$

To define the third approximate solution $x^{(3)}$, note that $\bar{i}$ and $\bar{j}-1$ given by $\beta_{\bar{k}} = \bar{\lambda} = \rho_{\bar{i}}(\bar{j}-1, \bar{j})$ satisfy

(4.4) $$c_{\bar{i}\bar{j}-1}/a_{\bar{i}\bar{j}-1} > \rho_{\bar{i}}(\bar{j}-1, \bar{j})$$

as easily proved from assumptions (1.4)-(1.6). Let k' be the smallest index satisfying $\beta_{k'} \in L$ (see (3.11)) and $\beta_{k'} \geq c_{\bar{i}\bar{j}-1}/a_{\bar{i}\bar{j}-1}$ (recall that $c_{\bar{i}\bar{j}-1}/a_{\bar{i}\bar{j}-1}$ is not in L), where $k' > \bar{k}$ by (4.4), and let

(4.5) $$A' = \sum_{i \neq \bar{i}} a_{ij_i(k')} \quad (<b),$$

where $j_i(k')$ is given by (3.13) with $\bar{k}$ replaced by k' in its definition. Then $x^{(3)}$ is defined as follows.

(i) Case of $A' + a_{\bar{i}\bar{j}-1} \geq b$: Then

(4.6) $$x_{ij_i(k')}^{(3)} = 1 \text{ for } i \neq \bar{i},\ x_{ij}^{(3)} = 0 \text{ for } i \neq \bar{i} \text{ and } j \neq j_i(k')$$
$$x_{\bar{i}\bar{j}-1}^{(3)} = (b - A')/a_{\bar{i}\bar{j}-1},\ x_{\bar{i}j}^{(3)} = 0 \text{ for } j \neq \bar{j} - 1$$
$$z^{(3)} = \sum_{i \neq \bar{i}} c_{ij_i(k')} + c_{\bar{i}\bar{j}-1}(b - A')/a_{\bar{i}\bar{j}-1}$$

(ii) Case of $A' + a_{\bar{i}\bar{j}-1} < b$: Then $x^{(3)}$ is not computed and

(4.7) $$z^{(3)} = -\infty .$$

Finally, let $x^{(T)}$ denote the best solution among $x^{(1)}$, $x^{(2)}$ and $x^{(3)}$ if (3.16) is the case, and $\bar{x}$ if (3.15) is the case. Let

(4.8) $$z^{(T)} = \begin{cases} \max[z^{(1)}, z^{(2)}, z^{(3)}] & \text{if (3.16) is the case for } P \\ \bar{z} & \text{if (3.15) is the case for } P. \end{cases}$$

$x^{(T)}$ is called the approximate solution of P obtained *by rounding*.

Lemma 4.1. If $A' + a_{\bar{i}\bar{j}-1} \geq b$ holds for A' defined in (4.5), $x^{(3)}$ obtained by (4.6) is an optimal solution of $P' \equiv P(x_{\bar{i}j} = 0$ for $j \neq \bar{j}-1)$ (i.e., P with additional restrictions $x_{\bar{i}j} = 0$ for $j \neq \bar{j}-1$).

Proof: By (4.5) and (4.6), i is easily shown that

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} a_{ij} x_{ij}^{(3)} = b \tag{4.9}$$

holds and hence $x^{(3)}$ is a feasible solution of P' since it satisfies (1.1C) and (1.1D) as well. Next apply DUAL(P') to obtain an optimal solution of $\bar{P}'$. L for P' is obtained from L for P, by deleting $L_{\bar{i}}$ (see (3.10)) and inserting $c_{\bar{i}\bar{j}-1}/a_{\bar{i}\bar{j}-1}$. By condition $A' + a_{\bar{i}\bar{j}-1} \geq b$, DUAL($P'$) terminates at $\beta_{\bar{k}} = c_{\bar{i}\bar{j}-1}/a_{\bar{i}\bar{j}-1}$ with $\bar{e} = b - A'$ (>0). Therefore $\bar{x}$ computed for P' by (3.15) is equal to $x^{(3)}$, i.e., $x^{(3)}$ is an LP optimal solution. This shows that $x^{(3)}$ is an optimal solution of P'. □

Lemma 4.2. If $A' + a_{\bar{i}\bar{j}-1} < b$ holds for A' of (4.5), then $z^{(T)}$ obtained by (4.8) for P satisfies

$$z^{(T)}/\bar{z} > 3/4,$$

where $\bar{z}$ is the LP optimal value of $\bar{P}$.

Proof: Assume that (3.16) holds since otherwise $z^{(T)}/\bar{z} = 1$. Let $\bar{k}$ and $\bar{\lambda} = \rho_{\bar{i}}(\bar{j}-1, \bar{j})$ be obtained by DUAL(P), and let

$$C = \sum_{i \neq \bar{i}} c_{ij_i(\bar{k})} \tag{4.10}$$

(i.e., the first portion in $\bar{z}$ of (3.16)). We first show that

$$C \geq (a_{\bar{i}\bar{j}-1} - \bar{e})\rho_{\bar{i}}(\bar{j}-1, \bar{j}), \tag{4.11}$$

where $\bar{e}$ is given by (3.14). (Note that $a_{\bar{i}\bar{j}} < \bar{e} \leq a_{\bar{i}\bar{j}-1}$ holds by the termination condition of DUAL(P).) Now note that the LP optimal solution $\bar{x}$ obtained by (3.16) satisfies (1.1D) (and hence $\bar{x}$ is an optimal solution of P) if $\theta=0$ holds. In other words, $\tilde{x} = (\tilde{x}_{ij})$ defined by

$$\begin{aligned} &\tilde{x}_{ij} = \bar{x}_{ij} \text{ for } i \neq \bar{i} \text{ and } j = 1, 2, \ldots, m_i \\ &\tilde{x}_{\bar{i}\bar{j}-1} = 1, \ \tilde{x}_{\bar{i}j} = 0 \text{ for } j \neq \bar{j}-1 \end{aligned} \tag{4.12}$$

is an optimal solution of $\tilde{P}$ which is P with the right hand side b replaced by $\tilde{b} = b + (a_{\bar{i}\bar{j}-1} - \bar{e})$ (i.e., b is increased by $(a_{\bar{i}\bar{j}-1} - \bar{e})$ to make $\theta = 0$ in (3.17)). Thus

$$\tilde{z} = C + c_{\bar{i}\bar{j}-1} \tag{4.13}$$

is the optimal value of $\tilde{P}$. Since $\tilde{x}$ is also an LP optimal solution of $\tilde{P}$, we have

$$\bar{z} \leq \tilde{z} - \lambda(a_{\bar{i}\bar{j}-1} - \bar{e}). \tag{4.14}$$

This follows from the LP duality theory because $\bar{P}$ is obtained from the LP relaxation of $\tilde{P}$ by decreasing the right hand side by $(a_{\bar{i}\bar{j}-1} - \bar{e})$ and λ is the value of the dual variable λ corresponding to the constraint (1.1B) for both $\bar{P}$ and $\tilde{P}$.

The condition $A' + a_{\bar{i}\bar{j}-1} < b$ guarantees the property $x_{\bar{i}\bar{j}-1} = 1$ in the LP optimal solution of $P' = P(x_{\bar{i}j} = 0,\ j \neq \bar{j}-1)$ (execute DUAL(P') as in the proof of Lemma 4.1). Thus the LP optimal value $\bar{z}'$ of P' satisfies

$$\bar{z}' \geq c_{\bar{i}\bar{j}-1} \tag{4.15}$$

Since $\bar{z}' \leq \bar{z}$ holds, combining (4.13), (4.14) and (4.15) yields (4.11).

Now note that $z^{(T)} = \max[z^{(1)},\ z^{(2)}]$ holds by condition $A' + a_{\bar{i}\bar{j}-1} < b$. Consider $z^{(1)}$, $z^{(2)}$ and $\bar{z}$ as functions of $\bar{e}$ (see (4.3) and (3.16), where θ is determined by $\bar{e}$ as shown in (3.17)), and denote them by $z^{(1)}(\bar{e})$, $z^{(2)}(\bar{e})$ and $\bar{z}(\bar{e})$. By direct calculation, it can be shown that

(a) $z^{(1)}(\bar{e})/\bar{z}(\bar{e})$ is nonincreasing in $\bar{e}$,

(b) $z^{(2)}(\bar{e})/\bar{z}(\bar{e})$ is nondecreasing in $\bar{e}$, and

(c) $(z^{(1)}(\bar{e})/\bar{z}(\bar{e})) = (z^{(2)}(\bar{e})/\bar{z}(\bar{e}))$ if and only if

$$\bar{e} = a_{\bar{i}\bar{j}-1} c_{\bar{i}\bar{j}} / c_{\bar{i}\bar{j}-1} \tag{4.16}$$

holds. This proves that $z^{(T)}/\bar{z} = \max[z^{(1)}/\bar{z},\ z^{(2)}/\bar{z}]$ takes its minimum when $z^{(1)} = z^{(2)}$ holds, i.e., $\bar{e}$ satisfies (4.16).

Finally we show $z^{(T)}/\bar{z} > 3/4$. By (4.16) and (3.17), we have

$$\theta = \rho_{\bar{i}}(\bar{j}-1,\ \bar{j}) a_{\bar{i}\bar{j}-1} / c_{\bar{i}\bar{j}-1}. \tag{4.17}$$

Then

$$\begin{aligned} \frac{z^{(T)}}{\bar{z}} &\geq \frac{z^{(1)}(a_{\bar{i}\bar{j}-1} c_{\bar{i}\bar{j}} / c_{\bar{i}\bar{j}-1})}{\bar{z}(a_{\bar{i}\bar{j}-1} c_{\bar{i}\bar{j}} / c_{\bar{i}\bar{j}-1})} \\ &= \frac{C + c_{\bar{i}\bar{j}}}{C + \theta c_{\bar{i}\bar{j}} + (1-\theta) c_{\bar{i}\bar{j}-1}} \quad \text{(where } \theta \text{ is given by (4.17))} \\ &\geq \frac{(a_{\bar{i}\bar{j}-1} - \bar{e}) \rho_{\bar{i}}(\bar{j}-1,\ \bar{j}) + c_{\bar{i}\bar{j}}}{(a_{\bar{i}\bar{j}-1} - \bar{e}) \rho_{\bar{i}}(\bar{j}-1,\ \bar{j}) + \theta c_{\bar{i}\bar{j}} + (1-\theta) c_{\bar{i}\bar{j}-1}} \quad \text{(by (4.11))} \\ &= \frac{1 - \alpha(1 - \alpha + \beta)}{1 - \beta}, \end{aligned} \tag{4.18}$$

where

$$\alpha = \frac{c_{\bar{i}\bar{j}}}{c_{\bar{i}\bar{j}-1}},\quad \beta = \frac{a_{\bar{i}\bar{j}}}{a_{\bar{i}\bar{j}-1}},\ \text{and } 1 \geq \alpha > \beta > 0. \tag{4.19}$$

The last formula of (4.18) takes its minimum $(\beta+3)/4$ when $\alpha = (\beta+1)/2$ holds. $(\beta+3)/4$ is greater than $3/4$ because $0 < \beta < 1$. □

In closing this section, we show that $x^{(T)}$ is computed in $O(N)$ time after DUAL(P) is executed. $x^{(1)}$ and $x^{(2)}$ are trivially obtained in $O(N)$ time. To obtain A' of (4.5), DUAL(P) is slightly modified. First

(4.20) $$c_{ij-1}/a_{ij-1},\ i=1, 2,\dots, n \text{ and } j=2, 3,\dots, m_i$$

are added to list $L=\{\beta_1, \beta_2,\dots, \beta_N\}$ and sorted together with $\beta_1, \beta_2,\dots, \beta_N$. Whenever $\beta_{k-1} < c_{\bar{i}\bar{j}-1}/a_{\bar{i}\bar{j}-1}$ is detected in step D3 of DUAL(P),

(4.21) $$A' = b - v' - a_{\bar{i}\bar{j}_{\bar{i}}} = \sum_{i\neq\bar{i}} a_{ij_i}$$

is computed and stored with a pointer to $c_{\bar{i}\bar{j}-1}/a_{\bar{i}\bar{j}-1}$. This modification does not change the order of time required by DUAL(P). The computation of $x^{(3)}$ by (4.6) is then done in $O(N)$ time.

5. Approximate Algorithm by Breadth-1 Search

The approximate algorithm by breadth-1 search first obtains the LP optimal solution $\bar{x}$ of P by applying DUAL(P). If $\bar{x}$ is feasible in P (i.e., (3.15) holds or $\theta=0$ in (3.16)), then P is solved. Otherwise (i.e., (3.16) with $\theta>0$), $x^{(T)}$ is calculated and a new problem $P(x_{\bar{i}\bar{j}-1}=0)$ is generated, where $\bar{\lambda}=\rho_{\bar{i}}(\bar{j}-1, \bar{j})$. The same procedure is then repeated for $P(x_{\bar{i}\bar{j}-1}=0)$ until the LP optimal solution of a tested problem Q is feasible in Q. This process eventually terminates since the LP optimal solution of Q is feasible in Q if all N variables x_{ij} are fixed to 0. The best feasible solution obtained in this process is denoted by $x^{(B)}$ and its objective value by $z^{(B)}$.

Algorithm APPRXB1(P)

B1. $Q_1 \leftarrow P \quad z \leftarrow -\infty \quad h \leftarrow 1.$

B2.† Obtain the LP optimal solution $\bar{x}_h$ and its value $\bar{z}_h$ of Q_h (i.e., DUAL (Q_h) is executed). If $\bar{x}_h$ is feasible in Q_h, then

$$z^{(B)} \leftarrow \max[z,\ \bar{z}_h]$$

and halt. Otherwise calculate $x_h^{(T)}$ and its value $z_h^{(T)}$ of Q_h and let

$$z \leftarrow \max[z,\ z_h^{(T)}].$$

B3. $Q_{h+1} \leftarrow Q_h(x_{\bar{i}_h\bar{j}_h-1}=0)$, where $\bar{i}_h$ and $\bar{j}_h-1$ are determined by $\beta_{\bar{k}}=\bar{\lambda}=\rho_{\bar{i}_h}(\bar{j}_h-1, \bar{j}_h)$ obtained in DUAL(Q_h). Return to A2 after letting $h\leftarrow h+1$. □

---

† The LP optimal value $\bar{z}_1$ obtained for $Q_1=P$ gives an upper bound on the exact optimal value z^0 of P, i.e.,

$$z^{(B)} \le z^0 \le \bar{z}_1.$$

Theorem 5.1. $z^{(B)}$ obtained by APPRXB1(P) satisfies

$$z^{(B)}/z^0 > 3/4,$$

where z^0 is the exact optimal value of P.

Proof: Assume that APPRXB1(P) has solved Q_h, $h=1, 2, \ldots, H$, i.e., the LP optimal solution of Q_H is feasible in Q_H. Note that for each h satisfying $1 \le h \le H-1$, the optimal value of Q_h is equal either to the optimal value of $Q_h(x_{\bar{i}_h \bar{j}_h-1}=0)(=Q_{h+1})$ or to the optimal value of $Q_h(x_{\bar{i}_h j}=0,\ j \ne \bar{j}_h-1)$ (denoted by Q'_{h+1}). Now assume that condition $A'+a_{\bar{i}\bar{j}-1} \ge b$ of (4.6) holds for Q_1, Q_2, $\ldots$, Q_g ($0 \le g \le H-1$ because Q_H is solved by DUAL(Q_H)). Then optimal solutions of Q'_2, $Q'_3, \ldots, Q'_{g+1}$ are respectively obtained as $x^{(3)}$ by Lemma 4.1. Therefore if one of Q'_2, $Q'_3, \ldots, Q'_{g+1}$ contains an optimal solution of P, P is solved by APPRXB1(P), i.e., $z^{(B)}/z^0=1$. On the other hand, if none of Q'_2, $Q'_3, \ldots, Q'_{g+1}$ contains an optimal solution of P, Q_{g+1} contains an optimal solution of P. Denoting the optimal value of Q_{g+1} by z^0_{g+1}, we have

$$z^{(B)}/z^0 \ge z^{(T)}_{g+1}/z^0 = z^{(T)}_{g+1}/z^0_{g+1} \ge z^{(T)}_{g+1}/\bar{z}_{g+1} > 3/4,$$

where the last inequality follows from Lemma 4.2. □

By carefully analyzing the above proof, we notice that the execution of APPRXB1 may be cut off at B2 as soon as $A'+a_{\bar{i}\bar{j}-1} < b$ holds for the current Q_h. This does not change the worst case bound of Theorem 5.1. Our APPRXB1, however, seems to give better performance in the sense of the average quality of solutions, without increasing the order of computation time.

Theorem 5.2. Let $z^{(B)}$ and z^0 be defined as in Theorem 5.1. The bound obtained in Theorem 5.1 is sharp in the sense that there exists a problem P (defined by (1.1)) such that

$$z^{(B)}/z^0 < (3/4)+\delta \tag{5.1}$$

holds for any given $\delta > 0$.

Proof: Consider P with $n=2$, $m_1=m_2=2$ and the following coefficients:

$$\begin{aligned} &a_{11}=\alpha+1,\ a_{12}=1,\ a_{21}=((\alpha+2)/2\beta)+((\alpha+1)/2),\ a_{22}=(\alpha+2)/2\beta \\ &b=(a_{11}/2)+a_{21}=((\alpha+2)/2\beta)+(\alpha+1) \\ &c_{11}=2\beta,\ c_{12}=\beta,\ c_{21}=\frac{\beta}{2\alpha}(\alpha+1)+1,\ c_{22}=1, \end{aligned} \tag{5.2}$$

where α and β are parameters satisfying $\alpha>>\beta>>1$. First note that list L of (3.11) is given by

$$L=\{\beta_1,\ \beta_2,\ \beta_3,\ \beta_4\}=\{\rho_1(1,\ 2),\ \rho_2(1,\ 2),\ c_{22}/a_{22},\ c_{12}/a_{12}\}, \tag{5.3}$$

where

$$\rho_1(1,\ 2) = \rho_2(1,\ 2) = \beta/\alpha < c_{22}/a_{22} = 2\beta/(\alpha+2) < c_{12}/a_{12} = \beta.$$

Applying DUAL(Q_1), where $Q_1 = P$, we obtain

(5.4)
$$\begin{aligned} &\bar{k} = 1,\ \bar{\lambda} = \beta_{\bar{k}} = \rho_{\bar{i}}(\bar{j}-1,\ \bar{j}) = \rho_1(1,\ 2) = \beta/\alpha,\ \bar{e} = b - a_{21} = (\alpha+1)/2 \\ &\bar{x}_1 = (\bar{x}_{11},\ \bar{x}_{12},\ \bar{x}_{21},\ \bar{x}_{22}) = ((\alpha-1)/2\alpha,\ (\alpha+1)/2\alpha,\ 1,\ 0) \\ &\bar{z}_1 = 2\beta + 1. \end{aligned}$$

Since $\bar{x}_1$ is not feasible, $x^{(1)}$, $x^{(2)}$ and $x^{(3)}$ are computed for Q_1:

$$\begin{aligned} &x^{(1)} = (0,\ 1,\ 1,\ 0),\ z^{(1)} = (3\beta/2) + (\beta/2\alpha) + 1, \\ &x^{(2)} = (1/2,\ 0,\ 1,\ 0),\ z^{(2)} = (3\beta/2) + (\beta/2\alpha) + 1, \\ &x^{(3)} \text{ is not computed, and } z^{(3)} = -\infty, \\ &z_1^{(T)} = (3\beta/2) + (\beta/2\alpha) + 1. \end{aligned}$$

The result for $x^{(3)}$ follows since $k' = 4$ (because $c_{11}/a_{11} = 2\beta/(\alpha+1)$ is located between $\beta_3 = 2\beta/(\alpha+2)$ and $\beta_4 = \beta$ in L), $A' = 0$ and $A' + a_{\bar{i}\bar{j}-1} = a_{11} = \alpha+1 < b = ((\alpha+2)/2\beta) + (\alpha+1)$ hold in (4.5)-(4.7).

According to APPRXB1(P), $Q_2 = P(x_{\bar{i}\bar{j}-1} = 0) = P(x_{11} = 0)$ is next examined. However, Q_2 does not satisfy (1.7) because

$$\begin{aligned} a_{12} + a_{21} &= ((\alpha+2)/2\beta) + ((\alpha+1)/2) + 1 \\ &< b = ((\alpha+2)/2\beta) + (\alpha+1), \end{aligned}$$

and hence $x_{12} = x_{21} = 1$, $x_{22} = 0$ is an optimal solution as noted in the footnote given to D4 of DUAL(P). The optimal value is $z_2^0 = c_{12} + c_{21} = (\beta(\alpha+1)/2\alpha) + 1 + \beta = z^{(1)}$. This is stored as $z_2^{(T)}$. APPRXB1(P) then terminates. $z^{(B)}$ is given by

(5.5)
$$\begin{aligned} z^{(B)} &= \max[z_1^{(T)},\ z_2^{(T)}] \\ &= (3\beta/2) + (\beta/2\alpha) + 1. \end{aligned}$$

On the other hand, it is easy to show that

(5.6)
$$x^0 = (1,\ 0,\ 0,\ 1),\ z^0 = 2\beta + 1 (= \bar{z}_1)$$

are an optimal solution of P and its value, i.e.,

$$z^{(B)}/z^0 = [3\beta/2) + (\beta/2\alpha) + 1]/(2\beta+1) > 3/4.$$

The left hand side approaches to 3/4 when both α and β tend to infinity in such a way that $\beta/\alpha \to 0$ holds. □

Although the problem P in the above proof has $n = 2$ and $m_1 = m_2 = 2$, it can be easily generalized to a problem with arbitrary n and $m_i \geq 2$ ($i = 1, 2, \ldots, n$) by introducing coefficients such that all the new $\rho_i(j-1, j)$ and c_{im_i}/a_{im_i} are sufficiently smaller than $\rho_1(1, 2)$ of (5.3). Then these quantities are located to the left of $\rho_1(1, 2)$ in list L, and the computation of APPRXB1(P) is not substantially affected by this modification.

Now we estimate the computation time required to carry out APPRXB1(P). For each $Q_{h+1} = Q_h(x_{\bar{i}_h \bar{j}_h - 1} = 0)$ encountered in B2, sets of dominant indices J_i ($i = 1, 2, \ldots, n$) (see (3.5)) are first computed. Since J_i's for $i \neq \bar{i}_h$ do not change from the previous J_i for Q_h, only $J_{\bar{i}_h}$ is actually updated in $O(m_{\bar{i}_h})$ time (Lemma 3.1). Let L_i^h and L^h denote L_i and L for Q_h respectively (see (3.10), (3.11)). As a result of changing $J_{\bar{i}_h}$, $L_{\bar{i}_h}^h$ in L^h is replaced with $L_{\bar{i}_h}^{h+1}$ to obtain L^{h+1}. To perform this process efficiently, a master list $\tilde{L}$ is formed before solving $Q_1 = P$ in B2. $\tilde{L}$ consists of all the elements of the form

$$\text{(5.7)} \qquad \begin{array}{l} \rho_i(j_1, j_2), \ i = 1, 2, \ldots, n \text{ and } 1 \leq j_1 \leq j_2 \leq m_i \\ \dfrac{c_{ij}}{a_{ij}}, \quad i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, m_i \end{array}$$

and these elements are sorted in the nondecreasing order. Since $\tilde{L}$ contains

$$\textstyle\sum_{i=1}^n m_i(m_i - 1)/2 + \sum_{i=1}^n m_i \leq mN + N$$

elements, its requires $O(mN\log N)$ time (note that $mN \leq N^2$). However L^1 (list L for Q_1) can be formed from $\tilde{L}$ in $O(mN)$ time if appropriate links are added to $\tilde{L}$. Furthermore, each update from L^h to L^{h+1} is done in $O(m_i)$ time because at most $O(m_i)$ elements are involved each time in replacing $L_{\bar{i}_h}^h$ with $L_{\bar{i}_h}^{h+1}$. Thus the total time required for computing L^h for $h = 1, 2, \ldots, H$ in APPRXB1(P) is $O(mN)$ since $H \leq N$ holds as noted prior to APPRXB1(P), plus the extra $O(mN\log N)$ time required for sorting $\tilde{L}$.

In order to obtain $\bar{x}_h$ or $x_h^{(T)}$ for each Q_h, DUAL(Q_h) is executed for $h = 1, 2, \ldots, H$. DUAL(Q_1) searches $\bar{\lambda}$ on L^1 starting from the right end of L^1 to its left. To solve Q_h for $h \geq 2$, however, it is not necessary to repeat the search over again, by the next lemma.

Lemma 5.3. Let $\bar{\lambda}^h$ denote $\bar{\lambda}$ obtained as a result of executing DUAL(Q_h), and let $L_{\bar{i}_h}^h$ and $L_{\bar{i}_h}^{h+1}$ be defined as above. Then any element β which is in exactly one of $L_{\bar{i}_h}^h$ and $L_{\bar{i}_h}^{h+1}$ (i.e., those which are removed or inserted when L^{h+1} is obtained from L^h) satisfies

$$\beta \leq \bar{\lambda}^h$$

(Proof is given in Appendix B.)

Namely, L^2 is formed from L^1 without changing the elements located to the right of $\bar{\lambda}^1 = \beta_{\bar{k}}1$. Thus the search for the new $\bar{\lambda}^2$ can be resumed on L^2 from the element, which was next to the right of $\beta_{\bar{k}}1$ in L^1, to its left. This can be repeated for $h = 2, 3, \ldots, H$. When APPRXB1(P) halts (i.e., Q_H is solved), we have list L^H and its length is $O(N)$. Thus the total time required for search $\bar{\lambda}^1, \bar{\lambda}^2, \ldots, \bar{\lambda}^H$ is $O(N)$. Although we omit the details, computation of $\bar{x}_h$ and $x_h^{(T)}$ for all $h = 1, 2, \ldots, H$ can also be done in $O(N)$ time since we need only to compute how $\bar{x}_h$ and $x_h^{(T)}$ differ from $\bar{x}_{h-1}$ and x_{h-1}^T.

Consequently we have the next theorem.

Theorem 5.4. APPRXB1(P) can be implemented so that the required time is $O(mN\log N)$ for sorting $\tilde{L}$ and $O(mN)$ for the rest of computation. □

As noted in Introduction, the average computational results reported in [6] are extremely good, and breadth-1 search seems to be sufficient for practical purposes. Breadth-K search discussed in the following sections may be mainly of theoretical importance.

6. Approximate Algorithm by Breadth-K Search

During computation of breadth-K search algorithm, which will be described below, partial problems $P_\ell = P(D_\ell, E_\ell, F_\ell)$ are generated, where D_ℓ, E_ℓ and F_ℓ are some subsets of the set of all index pairs $I = \{(i, j) \mid i = 1, 2, \ldots, n, j = 1, 2, \ldots, m_i\}$. D_ℓ, E_ℓ and F_ℓ have the following interpretation:

(6.1)
$(i, j) \in D_\ell$ implies that $x_{ij} = 1$ holds in P_ℓ:
$(i, j) \in E_\ell$ implies that $x_{ij} = 0$ holds in P_ℓ:
$(i, j) \in F_\ell$ implies that $0 \le x_{ij} \le 1$ holds in P_ℓ and $x_{ij'} = 0$ hold for all $j' \ne j$.

Furthermore,

(6.2) $F_\ell = \phi$ or $\{(i, j)\}$ (a singleton)

is assumed by property (1.9). Problem $P_\ell = P(D_\ell, E_\ell, F_\ell)$ is actually constructed from P as follows:

(i) Delete all pairs $(i, j')(j' = 1, 2, \ldots, m_i)$ from I if $(i, j) \in D_\ell$ for some j, delete (i, j) if $(i, j) \in E_\ell$, and delete $(i, j')(j' \ne j)$ if $(i, j) \in F_\ell$. Let the resulting set of index pairs be G_ℓ.

(ii) $P_\ell = P(D_\ell, E_\ell, F_\ell)$ is defined by

$$\text{maximize } z_\ell = \textstyle\sum_{(i,j)\in D_\ell} c_{ij} + \sum_{(i,j)\in G_\ell} c_{ij}x_{ij},$$

(6.3) subject to $\sum_{(i,j)\in G_\ell} a_{ij}x_{ij} \le b_\ell$

$0 \le x_{ij} \le 1$, for $(i, j) \in G_\ell$,

at most one of $x_{ij}((i, j)\in G_\ell)$ is positive for each i,

where

(6.4) $b_\ell = b - \sum_{(i,j)\in D_\ell} a_{ij}$.

Breadth-K search algorithm, where K is a given positive integer, is a variant of branch-and-bound algorithms. During computation, $\mathscr{A}$ stores all the active partial problems, i.e., P_ℓ's which have been generated but not tested yet. The computation halts when $\mathscr{A}$ becomes empty. The best feasible solution of P obtained by then is the computed approximate solution $x^{(A)}$ of P with its value $z^{(A)}$. z is used to maintain the best objective value during computation.

Algorithm APPRXBK(P, K)

Remark. K is a positive integer.

A1. $\mathscr{A} \leftarrow \{P\}$, $x \leftarrow \phi$ and $z \leftarrow -\infty$.

A2. If $\mathscr{A} = \phi$, let $x^{(A)} \leftarrow x$, $z^{(A)} \leftarrow z$ and halt. If $\mathscr{A} \ne \phi$, select a partial problem $P_\ell = P(D_\ell, E_\ell, F_\ell) \in \mathscr{A}$.

A3. Obtain $\bar{z}_\ell$(LP optimal value), the approximate solution $x_\ell^{(B)}$ and its value $z_\ell^{(B)}$ of P_ℓ by executing APPRXB1(P_ℓ), and let $x \leftarrow x_\ell^{(B)}$ if $z_\ell^{(B)} > z$ and $z \leftarrow \max[z, z_\ell^{(B)}]$. Go to A4, (1) if P_ℓ is solved by APPRXB1(P_ℓ) (i.e., $\bar{z}_\ell = z_\ell^{(B)}$), (2) if $\bar{z}_\ell \le z$, or (3) if $|D_\ell \cup F_\ell| = K-1$, where $|X|$ denotes the cardinality of set X. Otherwise go to A5.

A4. Let $\mathscr{A} \leftarrow \mathscr{A} - \{P_\ell\}$ and return to A2.

A5. Select a variable $x_{i'j'}$ satisfying

(6.5) $c_{i'j'} = \max\{c_{ij} \mid (i, j) \in G_\ell\}$

(G_ℓ is defined in (i) above). Two cases are possible.

(a) If $|F_\ell| = 0$, generate three partial problems as follows:

$$P_{\ell_1} = P(D_\ell \cup \{(i', j')\}, E_\ell, F_\ell)$$

(6.6) $$P_{\ell_2} = P(D_\ell, E_\ell \cup \{(i', j')\}, F_\ell)$$

$$P_{\ell_3} = P(D_\ell, E_\ell, \{(i', j')\}).$$

Return to A2 after letting $\mathscr{A} \leftarrow \mathscr{A} \cup \{P_{\ell_1}, P_{\ell_2}, P_{\ell_3}\} - \{P_\ell\}$.

(b) If $|F_\ell| = 1$, generate only two partial problems P_{ℓ_1} and P_{ℓ_2}. Return to A2 after letting $\mathscr{A} \leftarrow \mathscr{A} \cup \{P_{\ell_1}, P_{\ell_2}\} - \{P_\ell\}$. □

Now let

(6.7) $$(i_1, j_1), (i_2, j_2), \ldots, (i_N, j_N)$$

be the priority order in the selection of x_{ij}, in (6.5), i.e., $c_{i_1j_1} \geq c_{i_2j_2} \geq \ldots \geq c_{i_Nj_N}$. Let

(6.8) $$I_k = \{(i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k)\}.$$

If no P_ℓ generated in APPRXBK(P, K) is solved by APPRXB1(P_ℓ) or satisfies $\bar{z}_\ell \leq z$ (i.e., neither (1) nor (2) occurs in A3), all partial problems $P_\ell = P(D_\ell, E_\ell, F_\ell)$ satisfying

(6.9)
$$\begin{aligned} &D_\ell \cup E_\ell \cup F_\ell = I_k \text{ for some } k > 0 \\ &|D_\ell \cup F_\ell| \leq K - 1 \\ &|F_\ell| \leq 1 \\ &1 \leq k \leq N \end{aligned}$$

are generated in APPRXBK(P, K). This is easily proved from the generation mechanism of APPRXBK.

Next note that at least one partial problem generated from P_ℓ in A5 contains an optimal solution of P_ℓ. Furthermore, if $\bar{z}_\ell \leq z$ holds in A3, P_ℓ does not contain a solution which has a greater objective value than z. Therefore, if condition (3) $|D_\ell \cup F_\ell| = K - 1$ is deleted from A3, the resulting algorithm is simply an ordinary branch-and-bound algorithm (e.g., [5, 9, 10]) using the LP values as upper bounds. Although such a branch-and-bound algorithm obtains an exact optimal solution, the computation time is likely to grow exponentially with respect to N. Thus the condition $|D_\ell \cup F_\ell| = K - 1$ in A3 is a key trick which keeps the computation time within a polynomial order, while the selection of variables by (6.5) is essential to keep $z^{(A)}$ within a certain bound.

If K is set to 1 in APPRXBK(P, K), it halts after testing only the original problem P. Thus breadth-K search with K=1 is the same as breadth-1 search discussed in Section 5. Fig. 1 shows typical search trees of generated partial problems when K is set to 2 and 3. In the figure, each node represents a partial problem; the top node represents the original problem P. A node $P_{\ell'}$ is placed below P_ℓ with a connecting edge if $P_{\ell'}$ is generated from P_ℓ in A5.

The next theorem summarizes the results for the computational complexity. The quality of $z^{(A)}$ will be discussed in the next section.

Theorem 6.1. The computation time required for APPRXBK(P, K) is as follows.

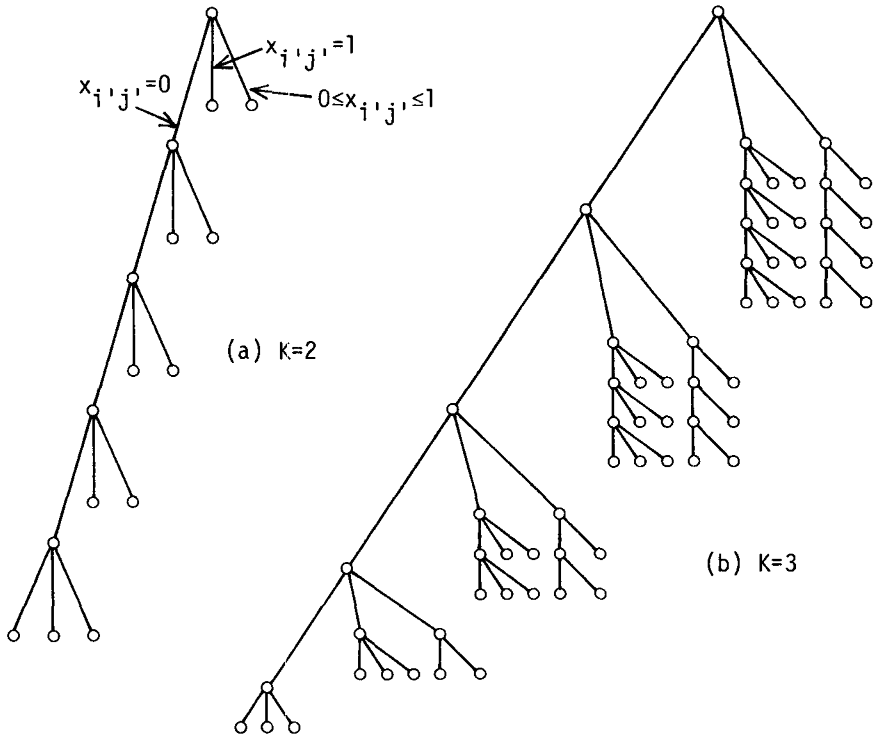

Fig. 1 Typical search trees generated by breadth-K search. (If a node is decomposed into three nodes according to (6.6), the left one represents P_{ℓ_1}, the middle one represents P_{ℓ_2}, and the right one represents P_{ℓ_3}. If a node is decomposed into two, the left one represents P_{ℓ_1}, and the right one represents P_{ℓ_2}.)

$O(mN\log N)$ for $K=1$,

$O(mKN^K)$ for $K>1$.

Proof: The result for $K=1$ was proved in Theorem 5.4. In order to consider the case of $K>1$, first notice that the number of partial problems generated by APPRXBK(P, K) is $O(KN^{K-1})$. This is because the number of the generated partial problem P_ℓ is bounded above by

$$3\times(\text{the number of } P_\ell \text{ generated at A5 as } P_{\ell_1} \text{ of (6.6)})$$

since at most three partial problems are generated at each decomposition. The latter number is not greater than the number of subsets D_ℓ, F_ℓ with $|D_\ell \cup F_\ell| \le K-1$ and $|F_\ell| \le 1$ by (6.9), i.e., $K\cdot\sum_{k=1}^{K-1}\binom{N}{k} \le KN^{K-1}$. For each generated P_ℓ, APPRXB1(P_ℓ) is executed in A3, requiring $O(mN)$ time (Theorem 5.4) if the time for sorting $\tilde{L}$ is not included. Since $\tilde{L}$ is sorted only at the beginning of APPRXBK(P, K), the total time is given by

$$\max[O(mN\log N),\ O(mN)\times O(KN^{K-1})] = O(mKN^K). \quad \square$$

7. Analysis of Breadth-K Search

After a series of lemmas, we derive in this section a worst case bound of $z^{(A)}$ calculated by APPRXBK(P, K). It will also be shown that the obtained bound is sharp.

Lemma 7.1. If P has an optimal solution with at most (K-1) positive variables, then $z^{(A)}$ obtained by APPRXBK(P, K) satisfies $z^{(A)}=z^0$ (the exact optimal value of P). On the other hand, if P has no such optimal solution, APPRXBK(P, K) generates a partial problem $P_\ell = P(D_\ell, E_\ell, F_\ell)$ such that $|D_\ell \cup F_\ell| = K-1$ and $z_\ell^0 = z^0$, where z_ℓ^0 denotes the optimal value of P_ℓ.

Proof: First assume that P has an optimal solution $x^0=(x_{ij}^0)$ with at most K-1 positive variables. Define

$$(7.1)\qquad \begin{aligned} &(i,\ j)\in D \quad \text{if } x_{ij}^0 = 1 \\ &(i,\ j)\in F \quad \text{if } 0<x_{ij}^0<1, \end{aligned}$$

where $|F|\le 1$ and $|D\cup F| \le K-1$. Assume that a $P_\ell = P(D, E, F)$ with some E is generated in APPRXBK(P, K), since $z^A=z^0$ has already been attained if an ancestor $P_{\ell'}$ of P_ℓ in the search tree (as shown in Fig. 1) is terminated by condition (1) or (2) in A3. If $|F|=0$, $z_\ell^{(B)}$ obtained for P_ℓ must satisfy $z_\ell^{(B)}\ge z^0$ (i.e., $z^{(B)}=z^0$) since all variables x_{ij} with $(i,\ j)\in D$ are already fixed to 1 in P_ℓ and $x_\ell^{(B)}$ is feasible in P_ℓ. If $F=\{(i',\ j')\}$, however, assume that Q_H is the last problem tested in B2 of APPRXB1(P_ℓ). Q_H is obtained from P_ℓ by fixing certain variables to 0 but $x_{i'j'}$ is not set to 0, because each $\bar{i}_h$ obtained in B3 (i.e., $\bar{\lambda}_h = \rho_{\bar{i}_h}(\bar{j}_h - 1, \bar{j}_h)$) is not equal to i' (note that $(i', j') \in F$ implies that list L contains only $c_{i'j'}/a_{i'j'}$ for $i = i'$). Thus the LP optimal value $\bar{z}_H$ of Q_H satisfies

$$\bar{z}_H \geq z^0,$$

but $\bar{z}_H$ is equal to the optimal value z_H of Q_H, i.e., $\bar{z}_H = z_H \leq z^0$ since the LP optimal solution of Q_H is feasible in P by definition of Q_H. In either case, therefore, z^0 is obtained as $z_\ell^{(B)}$ for P_ℓ.

Next assume that any optimal solution of P has at least K positive variables. Take any optimal solution x^0 and let $x^0_{i_1'j_1'}, x^0_{i_2'j_2'}, \ldots, x^0_{i_{K-1}'j_{K-1}'}$ be the first K-1 positive variables selected in the priority order (6.7). define D_ℓ and F_ℓ by

$$(7.2)\qquad \begin{array}{l} (i_s', j_s') \in D_\ell \text{ if } x^0_{i_s'j_s'} = 1 \\ (i_s', j_s') \in F_\ell \text{ if } 0 < x^0_{i_s'j_s'} < 1 \end{array}$$

for $s = 1, 2, \ldots, K-1$. We assume $|F_\ell| \leq 1$ by property (1.9). Then APPRXBK(P, K) generates $P_\ell = (D_\ell, E_\ell, F_\ell)$ with

$$(7.3)\qquad E_\ell = \{(i, j) \mid x^0_{ij} = 0 \text{ and } (i, j) \text{ has a higher priority than } (i_{K-1}', j_{K-1}') \text{ in } (6.7)\},$$

as easily proved from (6.9). The optimal value z_ℓ^0 of P_ℓ is obviously equal to z^0. □

Lemma 7.2. Let $P_\ell = P(D_\ell, E_\ell, F_\ell)$ be a partial problem with $|D_\ell| = K-1$ and $|F_\ell| = 0$ which is generated by APPRXBK(P, K). Then

$$(7.4)\qquad z_\ell^{(B)}/\bar{z}_\ell > (4K-1)/4K.$$

Proof: When APPRXB1(P_ℓ) is executed in A3, assume that condition $A' + a_{\bar{i}\bar{j}-1} \geq b$ of (4.6) holds for $Q_1, Q_2, \ldots, Q_g$ but not for Q_{g+1}. As shown in the proof of Theorem 5.1,

$$z_\ell^{(B)}/\bar{z}_\ell \geq z_{g+1}^{(T)}/\bar{z}_\ell$$

then follows, where $z_{g+1}^{(T)}$ is $z^{(T)}$ (see (4.8)) for Q_{g+1}. In this case, however, we have

$$(7.5)\qquad z_{g+1}^{(T)}/\bar{z}_\ell \geq \frac{(K-1)c_{\bar{i}\bar{j}-1}+(a_{\bar{i}\bar{j}-1}-\bar{e})\rho_{\bar{i}}(\bar{j}-1, \bar{j})+c_{\bar{i}\bar{j}}}{(K-1)c_{\bar{i}\bar{j}-1}+(a_{\bar{i}\bar{j}-1}-\bar{e})\rho_{\bar{i}}(\bar{j}-1, \bar{j})+\theta c_{\bar{i}\bar{j}}+(1-\theta)c_{\bar{i}\bar{j}-1}},$$

where $\theta = \rho_{\bar{i}}(\bar{j}-1, \bar{j})a_{\bar{i}\bar{j}-1}/c_{\bar{i}\bar{j}-1}$. This is derived by using the same argument as in the proof of Lemma 4.2, but considering that K-1 variables x_{ij} $((i, j) \in D_\ell)$

are already fixed to 1 and that $c_{ij} \geq c_{\bar{i}\bar{j}-1}$ holds for $(i, j) \in D_\ell$ by selection rule (6.5) in A5. Letting $\alpha = c_{\bar{i}\bar{j}}/c_{\bar{i}\bar{j}-1}$ and $\beta = a_{\bar{i}\bar{j}}/a_{\bar{i}\bar{j}-1}$, the right hand side of (7.5) can be rewritten as

$$((K-1) + \frac{1-\alpha(1+\beta-\alpha)}{1-\beta})/K, \tag{7.6}$$

where α, β satisfy $1 \geq \alpha > \beta > 0$. It is now direct to see that (7.4) takes its minimum when $\alpha = (\beta+1)/2$ holds, and its value is

$$(4K-1+\beta)/4K > (4K-1)/4K \quad (\text{since } \beta > 0). \quad \square$$

Lemma 7.3. Assume that P has an optimal solution x^0 with more than $(K-1)$ positive variables $x^0_{i'_1j'_1}, x^0_{i'_2j'_2}, \ldots, x^0_{i'_Tj'_T}$ $(T \geq K)$ arranged in the priority order (6.7). Furthermore assume that one variable $x^0_{i'_sj'_s}$ with $1 \leq s \leq K-1$ satisfies $0 < x^0_{i'_sj'_s} < 1$. Then

$$z^{(A)}/z^0 > \frac{1}{2} + \frac{1}{2}\sqrt{\frac{K-1}{K}} \tag{7.7}$$

holds for $K \geq 2$.

Proof: Let $P_\ell = P(D_\ell, E_\ell, F_\ell)$ be the partial problem generated in APPRXBK (P, K) such that

$$D_\ell = \{(i'_t, j'_t) \mid t = 1, 2, \ldots, K-1, t \neq s\}$$

$$F_\ell = \{(i_s, j_s)\}$$

and E_ℓ satisfies (7.3). (Such P_ℓ is generated as shown in Lemma 7.1.) P_ℓ satisfies $z^0 = z^0_\ell$, where z^0_ℓ is the optimal value of P_ℓ. When APPRXB1(P_ℓ) is executed at A3, $Q_1 = P_\ell$ and $Q_{h+1} = Q_h(x_{\bar{i}_h\bar{j}_h-1} = 0)$ $(h = 1, 2, \ldots, H-1)$ are generated. Let $\bar{h}$ be the smallest h satisfying

$$x^0_{\bar{i}_{\bar{h}}\bar{j}_{\bar{h}}-1} = 1 \text{ (i.e., } (\bar{i}_{\bar{h}}, \bar{j}_{\bar{h}}-1) = (i'_t, j'_t) \text{ for some } K \leq t \leq T). \tag{7.8}$$

Then

$$Q_{\bar{h}} = P(D_\ell, E_\ell \cup E', F_\ell),\ E' = \{(\bar{i}_1, \bar{j}_1-1), \ldots, (\bar{i}_{\bar{h}-1}, \bar{j}_{\bar{h}-1}-1)\}. \tag{7.9}$$

If there is no such $\bar{h}$, the LP optimal solution of Q_H is also an optimal solution of Q_H and hence $z^{(A)}/z^0 \geq z^{(B)}_\ell/z^0 \geq \bar{z}_\ell/z^0 = z^0/z^0 = 1$ follows. Thus assume that $\bar{h}$ exists. Obviously $Q_{\bar{h}}$ contains an optimal solution x^0 of P, and its $\bar{\lambda}$ value (denoted $\bar{\lambda}_{\bar{h}}$) satisfies

$$\bar{\lambda}_{\bar{h}} = \rho_{\bar{i}_{\bar{h}}}(\bar{j}_{\bar{h}}-1, \bar{j}_{\bar{h}}). \tag{7.10}$$

Hereafter in this proof, $\bar{x}_{\bar{h}}$ (LP optimal solution of $Q_{\bar{h}}$), $\bar{z}_{\bar{h}}$ (the value of $\bar{x}_{\bar{h}}$), $x^{(T)}_{\bar{h}}$ ($x^{(T)}$ obtained for $Q_{\bar{h}}$), $\bar{\lambda}_{\bar{h}}$, $\bar{i}_{\bar{h}}$ and $\bar{j}_{\bar{h}}$ are respectively denoted by $\bar{x}$, $\bar{z}$,

$x^{(T)}$, $\bar{\lambda}$, $\bar{i}$ and $\bar{j}$ for notational simplicity. Two cases are possible.

(a) There exists a variable $x_{\hat{i}\hat{j}}$ $(\hat{i} \neq \bar{i})$ satisfying $(\hat{i}, \hat{j}) \notin D_\ell$, $\bar{x}_{\hat{i}\hat{j}} = 1$ and $c_{\hat{i}\hat{j}} \geq c_{\bar{i}\bar{j}-1}$ (this includes the case of $(\hat{i}, \hat{j}) = (i'_s, j'_s) \in F_\ell$). Then by an argument similar to the proofs of Lemma 4.2 and Lemma 7.2, we can derive

$$(7.11) \qquad z^{(T)}/\bar{z} \geq \frac{\sum_{(i,j)\in D_\ell} c_{ij} + c_{\hat{i}\hat{j}} + c_{\bar{i}\bar{j}}}{\sum_{(i,j)\in D_\ell} c_{ij} + c_{\hat{i}\hat{j}} + \theta c_{\bar{i}\bar{j}} + (1-\theta) c_{\bar{i}\bar{j}-1}},$$

where $\theta = \rho_{\bar{i}}(\bar{j}-1, \bar{j}) a_{\bar{i}\bar{j}-1}/c_{\bar{i}\bar{j}-1}$. Here $\sum_{(i,j)\in D_\ell} c_{ij} + c_{\hat{i}\hat{j}}$ are introduced because all $x_{ij}((i, j) \in D_\ell)$ and $x_{\hat{i}\hat{j}}$ are fixed to 1. The term $(a_{\bar{i}\bar{j}-1} - \bar{e})\rho_{\bar{i}}(\bar{j}-1, \bar{j})$ which is in (4.18) and (7.5) (it comes from C discussed in (4.10) and (4.11)) disappears in (7.8) because $c_{\hat{i}\hat{j}}$ may be included in C of (4.10). Since $c_{\hat{i}\hat{j}} \geq c_{\bar{i}\bar{j}-1}$ and $c_{ij} \geq c_{\bar{i}\bar{j}-1}$ for $(i,j) \in D_\ell$, the right hand side of (7.11) is further bounded below by

$$(7.12) \qquad \frac{(K-1)c_{\bar{i}\bar{j}-1} + c_{\bar{i}\bar{j}}}{(K-1)c_{\bar{i}\bar{j}-1} + \theta c_{\bar{i}\bar{j}} + (1-\theta) c_{\bar{i}\bar{j}-1}} = \frac{(K-1) + \alpha}{(K-1) + (2\alpha - \beta - \alpha^2)/(1-\beta)},$$

where $\alpha = c_{\bar{i}\bar{j}}/c_{\bar{i}\bar{j}-1}$ and $\beta = a_{\bar{i}\bar{j}}/a_{\bar{i}\bar{j}-1}$. α and β satisfy $1 \geq \alpha > \beta > 0$. By direct calculation, it can be shown that (7.12) attains its minimum when $\beta \to 0$ and

$$(7.13) \qquad \alpha = -(K-1) + \sqrt{(K-1)^2 + (K-1)}.$$

Substituting these into (7.12), we obtain

$$z_\ell^{(T)}/\bar{z}_\ell > \frac{1}{2} + \frac{1}{2}\sqrt{\frac{K-1}{K}}.$$

This proves (7.7) because $\bar{z}_\ell \leq \bar{z}$ and $z^{(A)} \geq z_\ell^{(T)}$.

(b) All variables x_{ij} with $\bar{x}_{ij} = 1$ $(i \neq \bar{i})$ and $(i, j) \in D_\ell$ satisfy

$$(7.14) \qquad c_{ij} < c_{\bar{i}\bar{j}-1}.$$

In particular this implies $\bar{x}_{i'_s j'_s} = 0$ $((i'_s, i'_s) \in F_\ell)$ since (7.8) (7.10) are not satisfied if $i'_s = \bar{i}$ and $j'_s = \bar{j}-1$ (i.e., $\lambda_{\bar{h}} = c_{i'_s j'_s}/a_{i'_s j'_s}$) (hence we have $\bar{x}_{i'_s j'_s} = 0$ or 1), but (7.14) is not satisfied if $\bar{x}_{i'_s j'_s} = 1$ by $c_{i'_s j'_s} \geq c_{\bar{i}\bar{j}-1}$. In other words, $c_{i'_s j'_s}/a_{i'_s j'_s}$ is located to the left of $\rho_{\bar{i}}(\bar{j}-1, \bar{j})$ (see (7.10)) in list L for $Q_{\bar{h}}$. Now let

$$(7.15) \qquad E = \{(i, j) \mid (i, j) \notin D_\ell \cup F_\ell \cup \{(\bar{i}, \bar{j}-1)\} \text{ and } (i, j) \text{ has higher priority than } (\bar{i}, \bar{j}-1) \text{ in the sense of (6.7)}\}$$

and consider the following partial problem

(7.16) $P'_\ell = P(D_\ell \cup \{(\bar{i}, \bar{j}-1)\}, E \cup E', F_\ell)$

(E' is given in (7.9)). In other words, P'_ℓ is $Q_{\bar{h}}$ with a restriction $x_{\bar{i}\bar{j}-1} = 1$ added. Note also that $E \supset E_\ell$ holds. By (7.8), P'_ℓ contains an optimal solution of P. We now show that $\bar{x}'_{i'_s j'_s} = 0$ still holds in the LP optimal solution $\bar{x}'$ of P'_ℓ. First note that

(7.17) $\sum_{i \neq \bar{i}} \sum_{j=1}^{m_i} a_{ij}\bar{x}_{ij} + a_{\bar{i}\bar{j}-1} \geq b_\ell$

holds for $\bar{x}$ (b_ℓ is defined in (6.4)) since the computation of $\bar{x}$ for $Q_{\bar{h}}$ has terminated at $\rho_{\bar{i}}(\bar{j}-1, \bar{j})$ by assumption. This means that the computation of DUAL(P'_ℓ) terminates at some β_k which is located to the right of $\rho_{\bar{i}}(\bar{j}-1, \bar{j})$ in list L, because condition $x_{\bar{i}\bar{j}-1} = 1$ is further added. Thus $\bar{x}'_{i'_s j'_s} = 0$ follows since $c_{i'_s j'_s}/a_{i'_s j'_s}$ is located to the left of $\rho_{\bar{i}}(\bar{j}-1, \bar{j})$ in L.

Next construct the following problem

(7.18) $P''_\ell = P(D_\ell \cup \{(\bar{i}, \bar{j}-1)\}, E, F_\ell)$

by deleting E' from $E \cup E'$. P''_ℓ also contains an optimal solution of P. The LP optimal solution $\bar{x}''$ of P''_ℓ also satisfies $\bar{x}''_{i_s j_s} = 0$ since P''_ℓ is less constrained than P'_ℓ and hence $\bar{\lambda}' \leq \bar{\lambda}''$ holds, where $\bar{\lambda}''(\bar{\lambda}')$ is $\bar{\lambda}$ obtained by DUAL for $P''_\ell(P'_\ell)$.

Althouth P''_ℓ is not generated in APPRXBK(P, K) since $|D_\ell \cup \{(\bar{i}, \bar{j}-1)\} \cup F_\ell| = K$, the next problem is generated.

(7.19) $P^+_\ell = P(D_\ell \cup \{(\bar{i}, \bar{j}-1)\}, E \cup F_\ell, \phi)$.

Now compare P''_ℓ and P^+_ℓ. Since $\bar{x}''_{i'_s j'_s} = 0$ holds in the LP optimal solution of P''_ℓ, forcing $x_{i'_s j'_s}$ to 0 in P^+_ℓ does not introduce any additional restriction. Therefore we have

(7.20) $\bar{z}'' = \bar{z}^+$,

where $\bar{z}''(\bar{z}^+)$ is the LP optimal value of $P''_\ell(P^+_\ell)$.

Consequently, it follows that

$$
\begin{aligned}
z^{(A)}/z^0 &\geq z^{(A)}/\bar{z}'' \quad \text{(since } P'' \text{ contains an optimal solution of } P\text{)} \\
&= z^{(A)}/\bar{z}^+ \quad \text{(by (7.20))} \\
&\geq (z^+)^{(B)}/\bar{z}^+ \quad ((z^+)^{(B)} \text{ is the value obtained by APPRXB1}(P^+_\ell)) \\
&> (4K-1)/4K \quad \text{(by Lemma 7.2)} \\
&> \frac{1}{2} + \frac{1}{2}\sqrt{\frac{K-1}{K}} .
\end{aligned}
$$

This completes the proof. □

Theorem 7.4. The objective value $z^{(A)}$ obtained by APPRXBK(P,K) satisfies

(7.21) $$z^{(A)}/z^0 > \frac{1}{2}+\frac{1}{2}\sqrt{\frac{K-1}{K}} \quad \text{for } K \geq 2,$$

where z^0 is the optimal value of P.

Proof: Assume that P has an optimal solution x^0 with positive variables

$$x^0_{i'_1j'_1},\ x^0_{i'_2j'_2},\ldots,\ x^0_{i'_Tj'_T}$$

arranged in the priority order of (6.7). If $T \leq K-1$, $z^A = z^0$ holds as shown in Lemma 7.1. Thus assume $T > K-1$. Next if $x^0_{i'_1j'_1} = x^0_{i'_2j'_2} = \ldots = x^0_{i'_{K-1}j'_{K-1}} = 1$, consider partial problem $P_\ell = (D_\ell, E_\ell, F_\ell)$ defined by

$$D_\ell = \{(i'_t, j'_t) \mid t = 1, 2, \ldots, K-1\}$$

$$F_\ell = \phi$$

$$E_\ell = \{(i, j) \mid (i, j) \notin D_\ell \text{ and } (i, j) \text{ has a higher priority than } (i'_{K-1}, j'_{K-1}) \text{ in (6.7)}\}.$$

P_ℓ contains an optimal solution x^0 of P, and furthermore P_ℓ is generated in APPRXBK(P, K) since $|D_\ell \cup F_\ell| = K-1$. Then by Lemma 7.2, we have

$$\begin{aligned} z^{(A)}/z^0 &\geq z_\ell^{(B)}/z_\ell^0 \quad (z_\ell^0 \text{ is the optimal value of } P_\ell) \\ &\geq z_\ell^{(B)}/\bar{z}_\ell \quad (\bar{z}_\ell \text{ is the LP optimal value of } P_\ell) \\ &> (4K-1)/4K > \frac{1}{2}+\frac{1}{2}\sqrt{\frac{K-1}{K}}\ . \end{aligned}$$

Finally if $T > K-1$ and $x^0_{i'_sj'_s} < 1$ for some $1 \leq s \leq K-1$, we obtain $z^{(A)}/z^0 > \frac{1}{2}+\frac{1}{2}\sqrt{\frac{K-1}{K}}$ by lemma 7.3. □

By combining Theorem 6.1 and Theorem 7.4, the next theorem is immediate.

Theorem 7.5. For any ε with $0 < \varepsilon < \frac{1}{4}$, an approximate solution $x^{(A)}$ of P (see (1.1)) satisfying

(7.22) $$z^{(A)}/z^0 > 1-\varepsilon$$

is obtained in $O(m\lceil \frac{1}{4(\varepsilon-\varepsilon^2)} \rceil N^{\lceil \frac{1}{4(\varepsilon-\varepsilon^2)} \rceil})$ time, where z^0 is the optimal value of P.

Proof: First note that

(7.23) $$\frac{1}{2}+\frac{1}{2}\sqrt{\frac{K-1}{K}} \geq 1-\varepsilon \iff K \geq \frac{1}{4(\varepsilon-\varepsilon^2)}\ .$$

Thus breadth-$\lceil \frac{1}{4(\varepsilon-\varepsilon^2)} \rceil$ search gives an approximate solution $x^{(A)}$ satisfying $z^{(A)}/z^0 > 1-\varepsilon$ as shown in Theorem 7.4. The result for computation time then follows from Theorem 6.1. □

In concluding this section, we show that the bound obtained in Theorem 7.4 is sharp.

Theorem 7.6. For any given $\delta > 0$ and a positive integer $K \geq 2$, there is a problem P such that the approximate solution $x^{(A)}$ obtained by APPRXBK(P, K) satisfies

$$z^{(A)}/z^0 < \frac{1}{2} + \frac{1}{2}\sqrt{\frac{K-1}{K}} + \varepsilon$$

Proof: Consider the following problem P with $n = K$, $m_i = 2$, $i = 1, 2, \ldots, K-2$, $m_{K-1} = 1$ and $m_K = 2$.

$$\begin{aligned} P:\ & a_{i1} = 1,\ i = 1, 2, \ldots, K-2,\ a_{K-1,1} = (1-\beta)/(1-\alpha),\ a_{K1} = 1 \\ & a_{i2} = 1/3,\ i = 1, 2, \ldots, K-2,\ a_{K2} = \beta \\ & c_{i1} = 1,\ i = 1, 2, \ldots, K \\ & c_{i2} = 1/(1+\sqrt{2}),\ i = 1, 2, \ldots, K-2,\ c_{K2} = \alpha \\ & b = (K-2) + \alpha + (1-\beta)/(1-\alpha) \end{aligned} \tag{7.24}$$

where

$$\alpha = -(K-1) + \sqrt{(K-1)^2 + (K-1)} = 1/(1+\sqrt{K/K-1}) \text{ (see (7.13))} \tag{7.25}$$

and $\beta > 0$ is an arbitrarily small positive number. α satisfies

$$1/2 > \alpha \geq 1/(1+\sqrt{2}). \tag{7.26}$$

Parameters relevant to performing APPRXBK(P, K) are arranged below in the non-decreasing order.

(7.27)

$\rho_K(1,2)$, $\dfrac{c_{K-1,1}}{a_{K-1,1}}$	$\rho_i(1,2)$ (i=1,2,...,K-2)	c_{K1}/a_{K1}, c_{i1}/a_{i1} (i=1,2,...,K-2)	c_{i2}/a_{i2} (i=1,2,...,K-2)	c_{K2}/a_{K2}
$\frac{1-\alpha}{1-\beta}$	$\frac{3}{2}(2-\sqrt{2})$	1	$\frac{3}{1+\sqrt{2}}$	α/β

In addition, the following tie breaking rules are adopted.

(i) When DUAL(P) is carried out on list L (which is (7.27) with c_{i1}/a_{i1} (i=1, 2,..., K-1) deleted) from the right, $c_{K-1,1}/a_{K-1,1}$ is selected before $\rho_K(1, 2)$ though these have the same value.

(ii) (i', j') in (6.5) of APPRXBK(P, K) are selected in the following priority order, though some c_{ij}'s assume the same value.

$$(1,1),\ (2,1),\ \ldots,\ (K,1),\ (K,2),\ (1,2),\ (2,2),\ \ldots,\ (K-1,2). \tag{7.28}$$

Then the following LP optimal solution $\bar{x}$ and its value $\bar{z}$ of P are obtained.

$$\begin{aligned} & \bar{x}_{i1} = 1,\ i = 1, 2, \ldots, K-1,\ \bar{x}_{i2} = 0,\ i = 1, 2, \ldots, K-2 \\ & \bar{x}_{K1} = 1 - \theta,\ \bar{x}_{K2} = \theta \text{ and } \theta = (1-\alpha)/(1-\beta) \\ & \bar{z} = (K-1) + (2\alpha - \alpha^2 - \beta)/(1-\beta). \end{aligned} \tag{7.29}$$

On the other hand, an optimal solution x^0 and its value z^0 is given by

$$(7.30)\quad \begin{aligned} & x^0_{i1} = 1,\ i = 1, 2, \dots, K-2, K, \quad x^0_{i2} = 0,\ i = 1, 2, \dots, K-2, K \\ & x^0_{K-1,1} = (2\alpha - \alpha^2 - \beta)/(1 - \beta) \\ & z^0 = (K-1) + (2\alpha - \alpha^2 - \beta)/(1-\beta) \end{aligned}$$

(this is obviously optimal because $z^0 = \bar{z}$).

Next consider partial problems P_ℓ generated in APPRXBK(P, K). First note that two problems

$$(7.31)\quad \begin{aligned} & P_{\ell_1} = P(\{(1, 1), (2, 1), \dots, (K-1, 1)\}, \phi, \phi) \\ & P_{\ell_2} = P(\{(1, 1), (2, 1), \dots, (K-2, 1)\}, \phi, \{(K-1, 1)\}) \end{aligned}$$

are generated by the above rule (ii). These have the following approximate solution $x^{(B)}$ and its value $z^{(B)}$.

$$(7.32)\quad \begin{aligned} & x^{(B)}_{i1} = 1,\ i = 1, 2, \dots, K-1, \quad x^{(B)}_{i2} = 0,\ i = 1, 2, \dots, K-2 \\ & \begin{cases} x^{(B)}_{K1} = \alpha \\ x^{(B)}_{K2} = 0 \end{cases} \quad \text{or} \quad \begin{cases} x^{(B)}_{K1} = 0 \\ x^{(B)}_{K2} = 1 \end{cases} \\ & z^{(B)} = (K-1) + \alpha\ . \end{aligned}$$

Note also any ancestor of P_{ℓ_1} or P_{ℓ_2} has the same $z^{(B)}$. In all partial problems P_ℓ except for those mentioned above, at least one $x_{i'1}$ $(1 \le i' \le K-1)$ is fixed to 0. Therefore the objective value $z^{(B)}_\ell$ of such P_ℓ is bounded above by

$$(7.33)\quad \begin{aligned} & (K-1) + \frac{1}{1+\sqrt{2}}\ \text{(assuming } x_{i1} = 1\ (i \ne i'),\ x_{i'2} = 1 \text{ and } x_{ij} = 0 \\ & \qquad\qquad\qquad\qquad\qquad\qquad \text{for all other } (i, j)) \\ & \le (K-1) + \alpha \quad \text{(by (7.26)).} \end{aligned}$$

Consequently, we have

$$z^{(A)}/z^0 = ((K-1) + \alpha)/((K-1) + (2\alpha - \alpha^2 - \beta)/(1-\beta))$$

$$\xrightarrow[\beta \to 0]{} \frac{(K-1)+\alpha}{(K-1)+2\alpha-\alpha^2} = \frac{1}{2} + \frac{1}{2}\sqrt{\frac{K-1}{K}} \quad \text{(by (7.25)).} \quad \square$$

8. Discussion

Although the multiple-choice continuous knapsack problem is NP-complete as shown in Theorem 2.3, it seems to be a rather tractable one compared with other NP-complete problems. Exact optimal solutions can be obtained quite efficiently by a branch-and-bound algorithm as reported in [6], and good approximate solutions can be obtained in polynomial time as shown in this paper.

As noted previously, the discrete 0-1 version of our problem has been studied in several papers. In particular, [3, 11] contain polynomially bounded approximate algorithms. Such algorithms may also be used as approximate algorithms for our continuous version. For a given error bound $\varepsilon > 0$, the algorithm of [3] requires $O(N^{\lceil \frac{1}{\varepsilon}-1 \rceil}\log n)$ time, while the algorithm of [11] requires $O(\frac{nN}{\varepsilon})$ time. The latter is substantially faster than the former as well as than our breadth-K approach, in the sense of the worst case time bound. Judging from our limited experience, however, we believe that the breadth-K search method is at least competitive with the algorithm of [11] in the sense of the average computation time for the following reasons. (1) The number H of the problems Q_h solved in breadth-1 search applied to a partial problem P_ℓ (generated in breadth-K search algorithm) is very small (typical values of H observed in the experiment of [6] are less than 5 even for problems of $N \geq 1000$). (2) The number of the generated partial problems P_ℓ should also be very small by the fact that surprisingly accurate approximate value $z_\ell^{(B)}$ and the upper bound $\bar{z}_\ell$ are usually obtained for each P_ℓ (typical errors from the exact optimal value are less than 0.001% as reported in [6]); hence most P_ℓ are terminated by (1) or (2) of Step A3.

Recently, approximate algorithms for various combinatorial optimization problems receive intensive attention (e.g., [4, 7, 13, 14]). The idea of breadth-K search seems to be applicable to many other problems as well, with suitable modifications incorporated.

Acknowledgment

The author wishes to thank Professor H. Mine and Professor T. Hasegawa of Kyoto University for their helpful comments.

References

[1] Aho, A. V., Hopcroft, J. E., and Ullman, J. D.: *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading Mass., 1974.

[2] Balas, E., and Zemel, E.: Solving Large Zero-One Knapsack Problems. Management Science Research Report No. 408, Carnegie-Mellon University, 1977.

[3] Chandra, A. K., Hirschberg, D. S., and Wong, C. K.: Approximate Algorithms for the Knapsack Problem and Its Generalization. Research Report RC 5616, IBM T. J. Watson Research Center, 1975.

[4] Garey, M. R., and Johnson, D. S.: "Strong" NP-Completeness Results: Motivation, Examples and Implications. Technical Report, Bell Laboratories,

Murray Hill, 1976.

[5] Ibaraki, T.: Theoretical Comparison of Search Strategies in Branch-and-Bound Algorithms. *International J. of Computer and Information Sciences*, Vol. 5 (1976), 315-344.

[6] Ibaraki, T., Teranaka, K., Iwase, J., and Hasegawa, T.: The Multiple-Choice Knapsack Problem. *J. of Operations Research Society of Japan*, Vol. 21 (1978), 59-93.

[7] Johnson, D.: Approximate Algorithms for Combinatorial Problems. Proc. Fifth ACM Symposium on Theory of Computing (1973), 38-49.

[8] Karp, R. M.: Reducibility among Combinatorial Problems. In *Complexity of Computer Computations*, Miller, R. E., and Thatcher, J. W. (eds.), Plenum Press, New York, 1972.

[9] Kohler, W. H., and Steiglitz, K.: Characterization and Theoretical Comparison of Branch-and-Bound Algorithms for Permutation Problems. *J. of ACM*, No. 21 (1974), 140-156.

[10] Lawler, E. L., and Wood, D. E.: Branch-and-Bound Methods: A Survey. *Operations Research*, Vol. 14 (1966), 699-719.

[11] Lawler, E. L.: Fast Approximate Algorithms for Knapsack Problems. Memorandum UCB/ERL M77/45, University of California, Berkeley, 1977.

[12] Nauss, R. M.: The 0-1 Knapsack Problem with Multiple Choice Constraints. *European J. of Operational Research*, Vol. 2 (1978), 125-131.

[13] Sahni, S.: General Techniques for Combinatorial Approximation. *Operations Research*, Vol. 25 (1977), 920-936.

[14] Sahni, S.: Approximation Algorithms for the 0/1 Knapsack Problem. *J. of ACM*, Vol. 22 (1975), 115-124.

Appendix A. An $O(m_i)$ algorithm to compute the set of dominant indices J_i.

Lemma 3.1 is proved here by actually constructing an $O(m_i)$ algorithm to compute the set of dominant indices J_i for each i $(1 \leq i \leq n)$.

For a subset $J = \{k_1, k_2, \ldots, k_\ell\}$ $(k_1 < k_2 < \ldots < k_\ell)$ of $\{1, 2, \ldots, m_i\}$, it is shown in [6] that $k_\ell \notin J_i$ if

(A1) $$\rho_i(k_{\ell-1}, k_\ell) \geq \rho_i(k_\ell, k_{\ell+1})$$

holds, where ρ_i is defined in (3.7). Furthermore, $J_i = \{j_1, j_2, \ldots, j_{d_i}\}$ is the set of dominant indices if it satisfies

(A2) $$\rho_i(j_1, j_2) < \rho_i(j_2, j_3) < \ldots < \rho_i(j_{d_i-1}, j_{d_i}).$$

Thus J_i can be obtained by calculating the maximum subset J which does not contain a k_ℓ satisfying (A1). This is done by the following algorithm.

The algorithm starts with $J = \{1, 2, \ldots, m_i\}$ and, at each iteration, eliminates from J an element k_ℓ satisfying (A1). Let k_ℓ always denote the ℓ-th element from the left in J (i.e., the ℓ-th smallest element).

Algorithm DOM(P, i)

J1. $J(\triangleq\{k_1, k_2, \ldots, k_\ell\}) \leftarrow \{1, 2, \ldots, m_i\}$, $\ell \leftarrow 2$.

J2. If $\rho_i(k_{\ell-1}, k_\ell) < \rho_i(k_\ell, k_{\ell+1})$, go to J3; else go to J4 after deleting k_ℓ from J.

J3. $\ell \leftarrow \ell + 1$. If $\ell = |J|$ halt; else return to J2.

J4. Let $\ell \leftarrow \ell - 1$ if $\ell - 1 \geq 2$; do not change ℓ if $\ell - 1 = 1$. If $\ell = |J|$ (this is possible only when ℓ does not change), halt; else return to J2. □

Upon termination in J3 or J4, $J_i = J$ holds. The validity of the algorithm may be obvious from the comments given above. The time complexity is considered below.

Since each step J1-J4 requires a constant time, the complexity is measured by counting how many times J3 or J4 is visited. Let

(A3) $$p = |J| - \ell.$$

Initially p is $m_i - 2$. p descreases by one at J3 always, and at J4 if $\ell - 1 = 1$ holds. If $\ell \leftarrow \ell - 1$ is executed in J4, p does not change but $|J|$ decreases by one. Consequently either p or $|J|$ decreases by one whenever J3 or J4 is visted. Since $p = m_i - 2$ and $|J| = m_i$ initially, and $p \geq 0$ and $|J| \geq 2$ must hold, the above argument proves that J3 and J4 are executed $2m_i - 4$ times in total. Thus the time complexity of DOM(p, i) is $O(m_i)$.

Appendix B. Proof of Lemma 5.3.

For notational simplicity, assume that list

(B1) $$L_i = \{\rho_i(j_1, j_2), \rho_i(j_2, j_3), \ldots, \rho(j_{d-1}, j_d), c_{ij_d}/a_{ij_d}\}$$
$$\triangleq \{\beta_1, \beta_2, \ldots, \beta_d\} \text{ where } \beta_1 < \beta_2 < \ldots < \beta_d$$

is given for a set of dominant indices

(B2) $$J_i = \{j_1, j_2, \ldots, j_d\} \text{ of } J = \{1, 2, \ldots, m_i\}.$$

Assume further that

(B3) $$\bar{\lambda} = \rho_i(j_\ell, j_{\ell+1}), \text{ where } 1 \leq \ell \leq d-1,$$

holds and j_ℓ is deleted from J to obtain $J' = \{1, 2, \ldots, j_{\ell-1}, j_{\ell+1}, \ldots, m_i\}$.

Let L_i' be obtained from J_i' (the set of dominant indices of J') in the same manner as L_i for J_i. We show that any β which is in exactly one of L_i and L_i' satisfies

(B4) $\beta \le \bar{\lambda}$.

Before proceeding to the proof, construct a graph G_i to visualize the computation process of J_i (and hence L_i). It starts with the following set of nodes labeled with pairs of indices,

$$(1, 2), (2, 3), \ldots, (m_i - 1, m_i).$$

All nodes are defined to be uncovered. The following operation MERGE is then repeated as long as possible.

MERGE: Take any two uncovered nodes (p, q) and (q, r) $(p < q < r)$ such that $\rho_i(p, q) \ge \rho_i(q, r)$, and form a new node (p, r). (p, r) is placed above (p, q) and (q, r), and edges $((p, r), (p, q))$ and $((p, r), (q, r))$ are drawn. (p, q) and (q, r) are now covered and (p, r) is uncovered.

Fig. B1(a) illustrates an example of G_i for $J_i = \{1, 2, \ldots, 12\}$. The obtained set of uncovered nodes is $\{(1, 3), (3, 7), (7, 12)\}$. The following properties may be obvious from the definition of G_i.

(a) Let the resulting set of uncovered nodes in G_i be $\{(j_1, j_2), (j_2, j_3), \ldots, (j_{d-1}, j_d)\}$. Then $J_i = \{j_1, j_2, \ldots, j_d\}$ and $L_i = \{\rho_i(j_1, j_2), \ldots, \rho_i(j_{d-1}, j_d), c_{ij_d}/a_{ij_d}\}$ hold, where $\rho_i(j_1, j_2) < \rho_i(j_2, j_3) < \ldots < \rho_i(j_{d-1}, j_d) < c_{ij_d}/a_{ij_d}$.

(b) G_i is not unique, but any G_i has the same set of uncovered nodes.

(c) Any index $j_\ell (\ne 1, m_i) \in J_i$ appears in two uncovered nodes (p, j_ℓ) and (j_ℓ, q), where $p < j_\ell$ and $j_\ell < q$, while $j_\ell = 1$ or $j_\ell = m_i$ always appears in exactly one uncovered node

(d) Let (p, r) be generated from (p, q) and (q, r) by MERGE. Then

$$\rho_i(p, q) \ge \rho_i(p, r) \ge \rho_i(q, r)$$

holds as directly proved from the following properties

(B5)
$$\frac{b}{a} > \frac{d}{c} \ (a, b, c, d > 0) \text{ implies } \frac{b}{a} > \frac{b+d}{a+c} > \frac{d}{c}$$
$$\frac{b}{a} = \frac{d}{c} \ (a, b, c, d > 0) \text{ implies } \frac{b}{a} = \frac{b+d}{a+c} = \frac{d}{c} .$$

Now assume that $j_\ell (\ne m_i) \in J_i$ is deleted from J to obtain J'. G_i' (i.e., G_i for J') is obtained by

(1) deleting from G_i all the nodes containing j_ℓ and also all the edges connected to them,

(2) adding uncovered node $(j_{\ell-1}, j_{\ell+1})$ (if $j_\ell = 1$, this is not necessary),

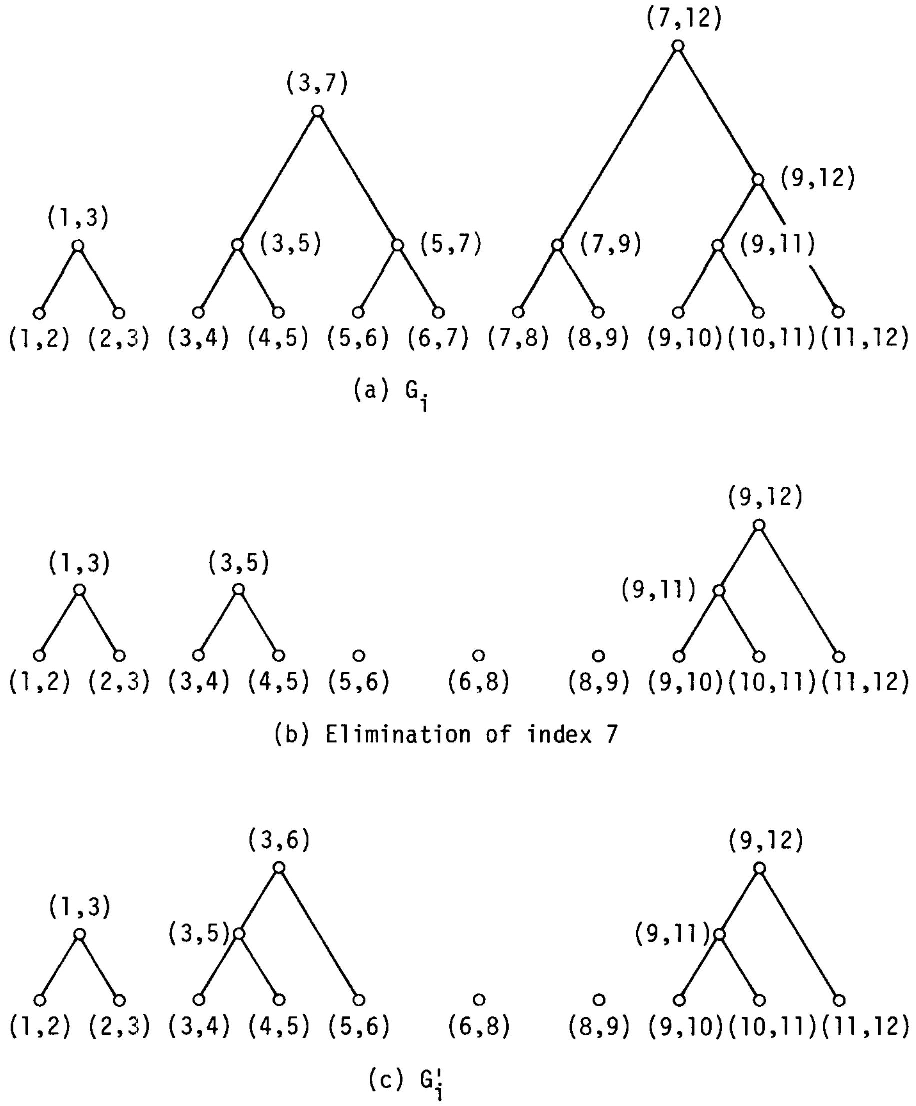

Fig. B1. Example of G_i and the computation of G_i'.

and

(3) applying MERGE as long as possible to the resulting graph.

Fig. 1B(b)(c) exemplifies this process.

Among all possible G_i (see comment (b) above), we assume here that G_i is obtained as follows.

(i) First G_i^1 and G_i^2 for $J^1 = \{1, 2, \ldots, j_{\ell-1}\}$ and $J^2 = \{j_{\ell+1}, \ldots, m_i\}$ are respectively obtained ($J^1 = \phi$ in case of $j_\ell = 1$).

(ii) G_i is then obtained from the union of G_i^1 and G_i^2 with two nodes $(j_{\ell-1}, j_\ell)$ and $(j_\ell, j_{\ell+1})$ added.

Under these assumptions, it is easy to see that the graph obtained after (1) above is exactly the union of G_i^1 and G_i^2. Thus $G_i^!$ is obtained by adding one node $(j_{\ell-1}, j_{\ell+1})$ to $G_i^1 \cup G_i^2$ instead of adding two nodes $(j_{\ell-1}, j_\ell)$ and $(j_\ell, j_{\ell+1})$.

Now we are ready to prove Lemma 5.3. First note that the elements which are in L_i but not in $L_i^!$ are only $\rho_i(j_{\ell-1}, j_\ell)$ and $\rho_i(j_\ell, j_{\ell+1})$. These satisfy

(B6) $$\rho_i(j_{\ell-1}, j_\ell) < \rho_i(j_\ell, j_{\ell+1}) = \bar{\lambda}.$$

Next we consider the elements which are in $L_i^!$ but not in L_i, i.e., these uncovered nodes newly generated in steps (1)-(3) above. Now let $(t, j_{\ell+1})$ be the node in G_i with the smallest index t satisfying $t > j_\ell$. The position of $(t, j_{\ell+1})$ is schematically illustrated in Fig. B2. After applying (1)-(3), two cases are possible.

(A) Node $(t, j_{\ell+1})$ remains uncovered (note that $(t, j_{\ell+1})$ is an uncovered node in $G_i^1 \cup G_i^2$ after (1) is executed): Then any new uncovered node (u, v) is located to the left of $(t, j_{\ell+1})$ in $G_i^!$. This means

$$\rho_i(u, v) < \rho_i(t, j_{\ell+1})$$

by definition of $G_i^!$. However

(B7) $$\rho_i(t, j_{\ell+1}) \le \rho_i(j_\ell, j_{\ell+1}) = \bar{\lambda}$$

holds by property (d) mentioned above.

(B) Node $(t, j_{\ell+1})$ becomes covered: Let $(s, j_{\ell+1})$ be the node in $G_i^!$ with the smallest index s. The generation process of $(s, j_{\ell+1})$ is illustrated in Fig. B2 by broken edges. Now note that

(B8) $$\rho_i(j_\ell - 1, j_\ell) < \rho_i(j_\ell - 1, j_\ell + 1)$$

is directly derived from

$$\rho_i(j_\ell - 1, j_\ell) \le \rho_i(j_{\ell-1}, j_\ell) < \rho_i(j_\ell, j_{\ell+1}) \le \rho_i(j_\ell, j_\ell + 1)$$

(by properties (a) and (d)),

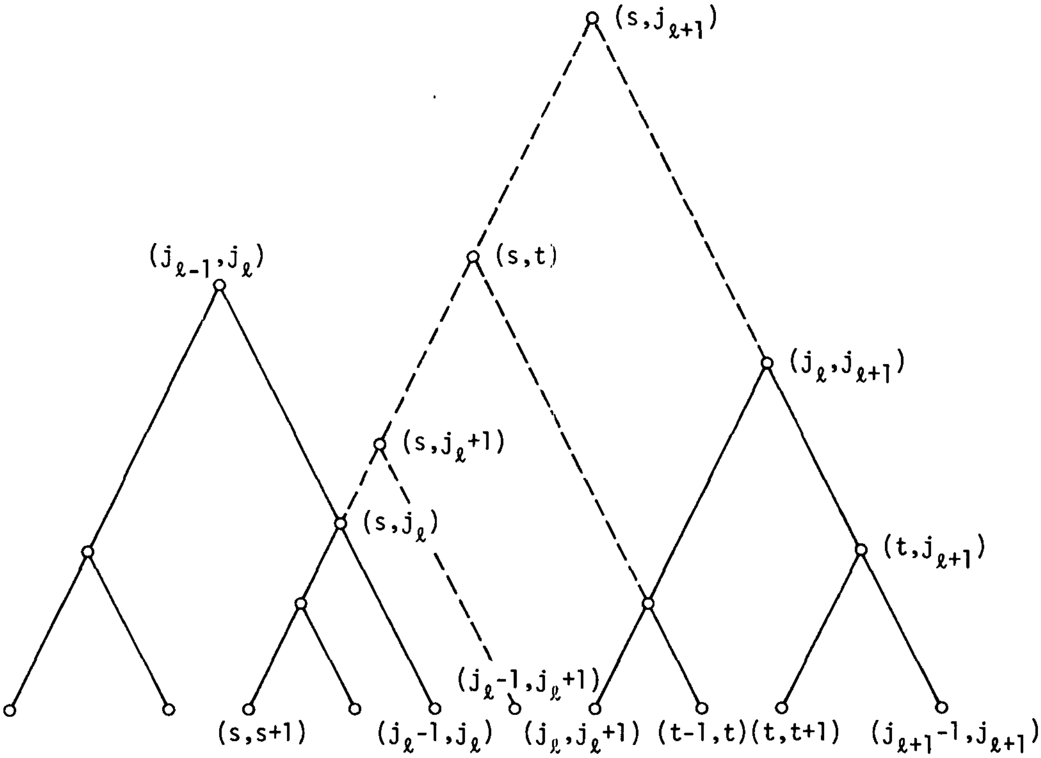

Fig. B2. Position of $(t, j_{\ell+1})$ and generation of $(s, j_{\ell+1})$ (broken edges indicate the newly generated part).

and property (B5). (B8) and the fact that $(s, j_{\ell+1})$ is a node in G_i' imply that (s, j_ℓ) is a node in G_i, as easily proved by the similarity between the construction of G_i (i.e., (i) and (ii)) and the construction of G_i' (i.e., (1)-(3)). Then we have

$$\rho_i(s, j_\ell) < \rho_i(j_\ell, j_{\ell+1})$$

by applying properties (a) and (d). However this means

(B9) $$\rho_i(s, j_\ell) < \rho_i(s, j_{\ell+1}) < \rho_i(j_\ell, j_{\ell+1}) = \bar{\lambda}$$

by (B5). In addition, there is no new uncovered node in G_i' to the right of $(s, j_{\ell+1})$. This is because operation MERGE cannot be applied between $(s, j_{\ell+1})$ and an uncovered node (u, v) to the right of $(s, j_{\ell+1})$ in $G_i^1 \cup G_i^2$ since

$$\rho_i(s, j_{\ell+1}) < \rho_i(j_\ell, j_{\ell+1}) < \rho_i(u, v)$$

holds. This also proves that $(s, j_{\ell+1})$ is an uncovered node in G_i'. Finally any new uncovered node (u, v) which is located to the left of $(s, j_{\ell+1})$ in G_i' satisfies

(B10) $$\rho_i(u, v) < \rho_i(s, j_{\ell+1}) < \bar{\lambda}$$

(see property (a)). From (B7) (B9) and (B10), Lemma 5.3 is proved. □

Toshihide IBARAKI: Department of
Applied Mathematics and Physics,
Faculty of Engineering,
Kyoto University,
Kyoto 606, Japan