Acta Cybernetica, Tom. 9, Fasc. 4, Szeged, 1990

On the multidimensional vector bin packing

J. Csirik* J.B.G. Frenk M. Labbé† S. Zhang

Econometrisch Instituut, Erasmus Universiteit Rotterdam, P.O. Box 1738, 3000DR Rotterdam, the Netherlands

Abstract

The multidimensional vector bin packing problem consists in packing m-dimensional items into a minimum number of m-dimensional bins with unit capacity in each of the m dimensions in such a way that the sum of each coordinate of the items received by any bin is not larger than one. We improve the lower bound of the First-Fit-Decreasing heuristic when $m \geq 5$ and odd, and prove that this heuristic is optimal when $m = 2$ if each item has at least one coordinate larger than 1/2. Finally, if this last condition holds and $m \geq 3$, we show that the problem remains NP-hard.

1 Introduction

In the multidimensional vector bin packing problem (MDVPP), we are given a list

$$L = (x_1, x_2, \dots, x_n)$$

of n items, where the items are vectors of form

$$(s_1(x_i), s_2(x_i), \dots, s_m(x_i)),\ i = 1, 2, \dots, n,$$

with $0 < s_j(x_i) \leq 1,\ j = 1, 2, \dots, m$. Then, the problem is to pack the items into a minimum number of bins, of unit capacity in each dimension, in such a way that the vector sum of the items received by any bin does not exceed $(1, 1, \dots, 1)$. Since this problem is a generalization of the classical one-dimensional bin packing problem, it is clearly NP-hard.

Garey et al.[1] analyse some heuristics to find an approximate solution to MDVPP. Specifically, they provide an exact worst-case bound for the First-Fit (FF) heuristic, but only lower and upper bounds for a variant of the First-Fit-Decreasing (FFD) method.

In this note, we improve Garey et al.'s lower bound. Then, for the special case where each item has at least one coordinate larger than 1/2, we show that FFD is optimal if $m = 2$ and that the problem remains NP-hard when $m \geq 3$.

*On leave from the Department of Computer Science, University of Szeged

†Fellow of the European Institute for Advanced Studies in Management, Brussels

2 Definitions and the lower bound for FFD

We denote the optimal number of bins for the list $L = (x_1, x_2, \ldots, x_n)$ by $OPT(L)$. For a heuristic algorithm A, we denote the number of bins used when applying A to L by $A(L)$. Let

$$R_A(k) = \max\{A(L)/OPT(L) : OPT(L) = k\}.$$

The asymptotic worst-case ratio of algorithm A is defined as

$$R_A = \limsup_{k\to\infty} R_A(k).$$

We consider the following generalization of the one-dimensional FFD algorithm.

Generalized First-Fit-Decreasing heuristic (GFFD)

Step 1 Reorder the list $L = (x_1, x_2, \ldots, x_n)$ in such a way that

$$s_{\max}(x_1) \geq s_{\max}(x_2) \geq \cdots \geq s_{\max}(x_n),$$

where $s_{\max}(x) = \max_{j=1,\ldots,m} s_j(x)$.

Step 2 Apply the FF heuristic to the ordered list (as for the one-dimensional case (cf.[1])).

Garey et al. [1] prove that for this heuristic

$$m + \frac{1}{m+2} - \frac{2}{m(m+1)(m+2)} \leq R_{GFFD} \leq m + \frac{1}{3}, \quad \text{if } m \geq 4.$$

For the special cases where $1 \leq m \leq 3$, they obtain slightly better bounds. Specifically, for $m = 1$, the exact ratio is $11/9$, and for $m = 2$ or 3, the lower bound is $m + 11/60$.

We now improve Garey et al.'s bound on R_{GFFD} for $m \geq 5$ and odd.

Lemma 1 *For $m \geq 5$ and odd,*

$$R_{GFFD} \geq m + \frac{1}{m+2} - \frac{1}{m(m+1)(m+2)}.$$

Proof. We use the following "bad" list (the first part is the same as given in Graham et al. [1]). Let k be an arbitrary positive integer which is a multiple of $m(m+1)(m+2)$. The list L is composed of m regions, the items in region i occur in L before the items in region $i+1$, $1 \leq i < m$. The items in region i are denoted by

$$x_{i,1}, x_{i,2}, \ldots, x_{i,q(i)}$$

where $q(i) = (i+1)(k-1)$ for $1 \leq i < m$, and $q(m) = (m+2)k$. Furthermore, let $0 < \varepsilon < k^{-4}$. We define the item coordinates as follows.

For $1 \leq i < m$,

$$s_i(x_{i,j}) = \begin{cases} \frac{1}{i+1} + \varepsilon t_{i,j}, & \text{if } 1 \leq j \leq i(k-1), \\ \frac{1}{i+1} - i\varepsilon t'_{i,j}, & \text{if } i(k-1) < j \leq q(i), \end{cases}$$

where $t_{i,j} = k - \lfloor \frac{j-1}{i} \rfloor$, $t'_{i,j} = j + 1 - i(k-1)$ and

$$s_l(x_{i,j}) = \varepsilon/m^2 \text{ for } 1 \le l \le m, l \ne i \text{ and } 1 \le j \le q(i).$$

For the items of the last region,

$$s_l(x_{m,j}) = \varepsilon/(2 \cdot m^2) \text{ for } 1 \le l \le m-1 \text{ and } 1 \le j \le q(m),$$

and

$$s_m(x_{m,j}) = \begin{cases} \frac{1}{m+1} + \frac{2}{m(m+1)(m+3)} - \frac{\varepsilon}{m+2}, & \text{if } 1 \le j \le k\frac{m-1}{2}, \\ \frac{1}{m+2} + \frac{1}{(m+1)(m+2)(m+3)} - \frac{\varepsilon}{m+2}, & \text{if } k\frac{m-1}{2} < j \le k\frac{m+1}{2}, \\ \frac{1}{m+3} + \frac{2}{m(m+1)(m+3)^2} - \frac{\varepsilon}{m+2}, & \text{if } k\frac{m+1}{2} < j \le q(m). \end{cases}$$

So, specifically, items in the first region have the following sizes:
$s(x_{1,1}) = (1/2 + k \cdot \varepsilon, \varepsilon/m^2, \varepsilon/m^2, \ldots, \varepsilon/m^2)$,
$s(x_{1,2}) = (1/2 + (k-1) \cdot \varepsilon, \varepsilon/m^2, \varepsilon/m^2, \ldots, \varepsilon/m^2)$,

$$\vdots$$

$s(x_{1,k-1}) = (1/2 + 2 \cdot \varepsilon, \varepsilon/m^2, \varepsilon/m^2, \ldots, \varepsilon/m^2)$,

$s(x_{1,k}) = (1/2 - 2 \cdot \varepsilon, \varepsilon/m^2, \varepsilon/m^2, \ldots, \varepsilon/m^2)$,
$s(x_{1,k+1}) = (1/2 - 3 \cdot \varepsilon, \varepsilon/m^2, \varepsilon/m^2, \ldots, \varepsilon/m^2)$,

$$\vdots$$

$s(x_{1,2(k-1)}) = (1/2 - k \cdot \varepsilon, \varepsilon/m^2, \varepsilon/m^2, \ldots, \varepsilon/m^2)$.

Accordingly, in the i-th region $(2 < i < m)$ (the "big" coordinates are in the i-th position):
$s(x_{i,1}) = s(x_{i,2}) = \ldots = s(x_{i,i}) =$

$$= (\varepsilon/m^2, \ldots, \varepsilon/m^2, \frac{1}{i+1} + k\varepsilon, \varepsilon/m^2, \ldots, \varepsilon/m^2),$$

$s(x_{i,i+1}) = s(x_{i,i+2}) = \ldots = s(x_{i,2i}) =$

$$= (\varepsilon/m^2, \ldots, \varepsilon/m^2, \frac{1}{i+1} + (k-1)\varepsilon, \varepsilon/m^2, \ldots, \varepsilon/m^2),$$

$$\vdots$$

$s(x_{i,(k-2)i+1}) = s(x_{i,(k-2)i+2}) = \ldots = s(x_{i,(k-1)i}) =$

$$= (\varepsilon/m^2, \ldots, \varepsilon/m^2, \frac{1}{i+1} + 2\varepsilon, \varepsilon/m^2, \ldots, \varepsilon/m^2),$$

$s(x_{i,(k-1)i+1}) = (\varepsilon/m^2, \ldots, \varepsilon/m^2, \frac{1}{i+1} - 2 \cdot i \cdot \varepsilon, \varepsilon/m^2, \ldots, \varepsilon/m^2)$,

$$s(x_{i,(k-1)i+2}) = (\varepsilon/m^2, \ldots, \varepsilon/m^2, \tfrac{1}{i+1} - 3 \cdot i \cdot \varepsilon, \varepsilon/m^2, \ldots, \varepsilon/m^2).$$

$$\vdots$$

$$s(x_{i,(k-1)(i+1)}) = (\varepsilon/m^2, \ldots, \varepsilon/m^2, \tfrac{1}{i+1} - k \cdot i \cdot \varepsilon, \varepsilon/m^2, \ldots, \varepsilon/m^2).$$

When applying GFFD to our list L, we may also partition the set of bins used into m subsets, each bin of subset i containing only items from region i of list L. For $1 \le i \le m-1$, we have $(k-1)$ bins in subset i, the l-th bin of them contains exactly i items with $s_i = \frac{1}{i+1} + (k+1-l)\varepsilon$ and one item with $s_i = \frac{1}{i+1} - i(k+1-l)\varepsilon$. With these items, the bin is full in the i-th dimension. So, we use $(m-1)(k-1)$ bins for the items in the $(m-1)$ first regions of L, and we can not pack items from the later regions in these bins, even if their i-th coordinate is just ε/m^2.
Concerning the items in the m-th region, their "large" coordinate is the last one and can be of three different types. It is easy to check that all of these items will be packed in some bin together with other items, the "large" coordinate of which being of the same type. Hence, we use $k \cdot \frac{m-1}{2}/m$ bins for the items with large coordinate of the first type, $k/(m+1)$ bins for the items of the second type and $k \cdot \frac{m+3}{2}/(m+2)$ for the third type. Consequently, the total number of bins used when applying GFFD to L is

$$(k-1)(m-1) + k \cdot (1 + \frac{1}{m+2} - \frac{1}{m(m+1)(m+2)})$$

On the other hand, the optimal packing uses at most k bins. To see this, we provide the following packing with k bins. Each bin contains $m+2$ items from the last region, i.e. $(m-1)/2$ items of the first type, one item of the second type and $(m+3)/2$ items of the third type. With these items, the sum of item sizes at each bin is

$$(\frac{m+2}{2m^2}\varepsilon, \frac{m+2}{2m^2}\varepsilon, \ldots, \frac{m+2}{2m^2}\varepsilon, 1-\varepsilon).$$

Moreover, each bin contains items from each of the other regions. Specifically:

- The first bin contains the first i items from each region i $(i < m)$, these are the items with size $s_i = 1/(i+1) + k \cdot \varepsilon$.
- The last bin contains the $((k-1)i+1)$-th item of each region $i < m$.
- Each remaining bin contains $i+1$ items from each subset i, i.e. i items with size $s_i = 1/(i+1) - t \cdot \varepsilon$ and one item with $s_i = 1/(i+1) + i \cdot (t-1) \cdot \varepsilon$ (for an appropriate t).

Note that for all bins and in each dimension, the capacity used by "large" coordinates is never bigger than $1-\varepsilon$ so that we leave place enough for "small" coordinates.
In conclusion, for our list L,

$$\frac{\mathrm{GFFD}(L)}{\mathrm{OPT}(L)} \ge \frac{1}{k}[(k-1)(m-1) + k(1 + \frac{1}{m+2} - \frac{1}{m(m+1)(m+2)})]$$

$$= m + \frac{1}{m+2} - \frac{1}{m(m+1)(m+2)} - \frac{m-1}{k},$$

which can be made arbitrarily close to

$$m + \frac{1}{m+2} - \frac{1}{m(m+1)(m+2)}$$

by choosing k large.

□

3 The special case

We study here the MDVPP for lists $L = (x_1, x_2, \ldots, x_n)$ such that

$$s_{\max}(x_i) > 1/2 \quad \text{for } i = 1, 2, \ldots, n. \tag{1}$$

A. *When $m = 2$, we prove that GFFD is optimal.* The idea behind this result is that, in such a case, each bin can contain at most two items, a situation which also occurs in the classical bin packing when all items have a size larger than 1/3 and this case is known to be polynomial.

Lemma 2 *If m=2, then for all lists for which (1) holds,* GFFD*(L)*=OPT*(L)*.

Proof. Assume that the elements in L are ranked by decreasing value of their largest coordinate. Furthermore, let $L = L_1 \cup L_2$ with $L_1 = (x_1, x_2, \ldots, x_p)$ and $L_2 = (y_1, y_2, \ldots, y_q)$ such that $s_{\max}(x_i) = s_1(x_i)$, $i = 1, 2, \ldots, p$ and $s_{\max}(y_i) = s_2(y_i)$, $i = 1, 2, \ldots, q$. Without loss of generality, we may assume that $L_1 \cap L_2 = \emptyset$.

Now, define a bipartitate graph $G = (L_1, L_2, E)$ where $(x_i, y_j) \in E$ iff $s_k(x_i) + s_k(y_j) \leq 1$, $k = 1, 2$. Since each bin can contain at most two elements (one from L_1 and the other one from L_2), the optimal packing of L corresponds to maximum matching M^* of G. Hence, to prove that GFFD is optimal, we shall show that the matching M_{GFFD} corresponding to the GFFD solution is optimal in G, i.e. there exists no augmenting path with respect to M_{GFFD} in G (see e.g. Papadimitriou and Steiglitz [3]). At the end of this proof, such a heuristic solution and its corresponding matching are illustrated by an example.

To begin with, remark that if $x_i \in L_1$ and $y_j \in L_2$ are both free vertices for M_{GFFD}, then $(x_i, y_j) \notin E$, for otherwise they would have been put together in a bin when applying GFFD.

Now, assume, by contradiction, that there exists an alternating path with respect to M_{GFFD}. From the above remark, we know that such a path contains more that one edge, i.e. at least three edges. Let

$$P = \{x_{i_1}, y_{j_1}, x_{i_2}, y_{j_2}, \ldots, y_{j_{l-1}}, x_{i_l}, y_{j_l}\}$$

be a *minimal alternating path* with respect to M_{GFFD}. Hence, x_{i_1} and y_{j_l} are free vertices, $(x_{i_k}, y_{j_k}) \in E \setminus M_{\text{GFFD}}$ for $k = 1, 2, \ldots, l$ and $(x_{i_{k+1}}, y_{j_k}) \in M_{\text{GFFD}}$ for $k = 1, 2, \ldots, l-1$. (see Figure 1 in which the edges of M_{GFFD} are indicated in waved lines). Furthermore, we denote by $x_{i_h} \succ x_{i_k}$ the fact that x_{i_h} has been considered before x_{i_k} when applying GFFD (i.e. $s_1(x_{i_h}) \geq s_1(x_{i_k})$). The same notation applies for items in L_2.

Claim. $x_{i_{k+1}} \succ x_{i_k}$ and $y_{i_{k+1}} \succ y_{i_k}$ for $k = 1, 2, \ldots, l-1$.

Proof. By induction.

a) $k = 1$. From the definition of P, $(x_{i_1}, y_{j_1}) \in E \setminus M_{\text{GFFD}}$ and $(y_{j_1}, x_{i_2}) \in M_{\text{GFFD}}$, i.e. GFFD put y_{j_1} and x_{i_2} in the same bin. Since x_{i_1} fits also with y_{j_1} in a bin, this means that $x_{i_2} \succ x_{i_1}$.

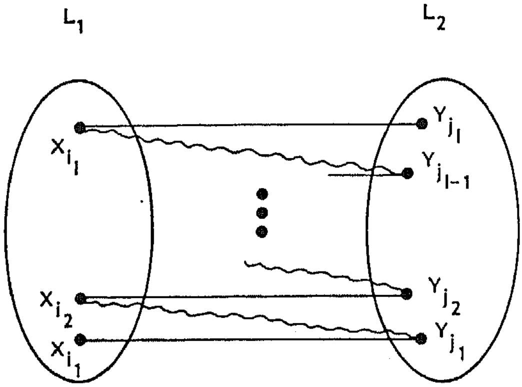

Figure 1: The minimal alternating path

Next, if $y_{j_2} \prec y_{j_1}$, we have that

$$s_1(x_{i_1}) + s_1(y_{j_2}) \le s_1(x_{i_2}) + s_1(y_{j_2}),$$

since $x_{i_2} \succ x_{i_1}$, and the right size is bounded above by 1, since $(x_{i_2}, y_{j_2}) \in E$; and

$$s_2(x_{i_1}) + s_2(y_{j_2}) \le s_2(x_{i_1}) + s_2(y_{j_1}),$$

since $y_{j_2} \prec y_{j_1}$, and the right size is bounded above by 1, since $(x_{i_1}, y_{j_1}) \in E$.

Hence $y_{j_2} \prec y_{j_1}$ implies that $(x_{i_1}, y_{j_2}) \in E$, which is impossible. Indeed, if $(x_{i_1}, y_{j_2}) \in E \setminus M_{\mathrm{GFFD}}$, then P is not minimal. If $(x_{i_1}, y_{j_2}) \in M_{\mathrm{GFFD}}$, then P contains a cycle. Consequently, $y_{j_2} \succ y_{j_1}$.

b) Assume $x_{i_{k-1}} \succ \ldots \succ x_{i_2} \succ x_{i_1}$ and $y_{i_{k-1}} \succ \ldots \succ y_{i_2} \succ y_{i_1}$. Since $y_{j_{k-1}}$ is not free and $y_{j_{k-1}} \succ y_{j_{k-2}}$ has been matched before $y_{j_{k-2}}$ when using GFFD. Because both $y_{j_{k-1}}$ and $y_{j_{k-2}}$ could have been matched with $x_{i_{k-1}}$, the item x_{i_k}, with which $y_{j_{k-1}}$ has been matched must be such that $x_{i_k} \succ x_{i_{k-1}}$.

Now, as in a), if $y_{j_k} \prec y_{j_{k-1}}$, we have that

$$s_1(x_{i_{k-1}}) + s_1(y_{j_k}) \le s_1(x_{i_k}) + s_1(y_{j_k}),$$

since $x_{i_k} \succ x_{i_{k-1}}$, and the right size is bounded above by 1, since $(x_{i_k}, y_{j_k}) \in E$; and

$$s_2(x_{i_{k-1}}) + s_2(y_{j_k}) \le s_2(x_{i_{k-1}}) + s_2(y_{j_{k-1}}),$$

since $y_{j_k} \prec y_{j_{k-1}}$, and the right size is bounded above by 1, since $(x_{i_{k-1}}, y_{j_{k-1}}) \in E$.

Hence $y_{j_k} \prec y_{j_{k-1}}$ implies that $(x_{i_{k-1}}, y_{j_k}) \in E$.

Then, if $(x_{i_{k-1}}, y_{j_1}) \in E \setminus M_{\mathrm{GFFD}}$, P is not minimal and if $(x_{i_{k-1}}, y_{j_k}) \in M_{\mathrm{GFFD}}$, M_{GFFD} contains two edge incident to $x_{i_{p-1}}$, which is impossible. Hence, $y_{j_k} \succ y_{j_{k-1}}$.

This completes the proof of the claim.

Finally, we know that the free item y_{j_l} of P which is also in L_2 is such that $y_{j_l} \succ y_{j_{l-1}}$. This is a contradiction, since, when applying GFFD, we considered it before $y_{j_{l-1}}$ and we did not matched it with x_{i_l}, though it was possible.

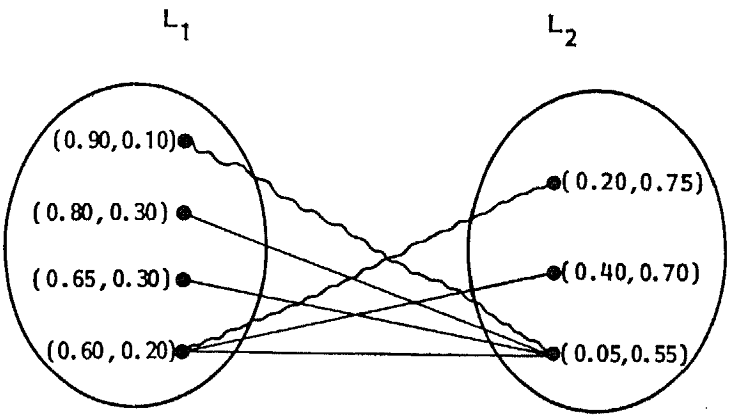

Figure 2: The graph and its maximum matching M_{GFFD} corresponding to the example.

Example: Consider the following list

$$L = \big((0.90, 0.10), (0.80, 0.30), (0.20, 0.75), (0.40, 0.70), (0.65, 0.30), (0.60, 0.20), (0.05, 0.55)\big).$$

When applying heuristic GFFD, we get GFFD(L)=5 and the corresponding bins are

$B_1 = \{(0.90, 0.10)(0.05, 0.55)\}$,
$B_2 = \{(0.80, 0.30)\}$,
$B_3 = \{(0.20, 0.75)(0.60, 0.20)\}$,
$B_4 = \{(0.40, 0.70)\}$,
$B_5 = \{(0.65, 0.30)\}$.

The graph associated with L is presented in Figure 2 where the edges of the maximum matching M_{GFFD} are indicated in waved lines.

B. *When the dimension of the items in a list L is at least three,* MDVPP remains unfortunately NP-hard even if each item has at least one coordinate larger than 1/2.

To see this, we first define the decision version of the 3-dimensional vector bin packing problem for which condition (1) holds (we call it 3-DVPP with $s_{\max}$ large.)

3-DVPP with $s_{\max}$ large (P1)

INSTANCE: A finite set L of 3-dimensional nonnegative integer vectors

$$a_i = (s_1(a_i), s_2(a_i), s_3(a_i)), \quad i = 1, 2, \ldots, n.$$

A positive integer bin capacity B such that $\max(s_1(a_i), s_2(a_i), s_3(a_i)) > B/2$ for $i = 1, 2, \ldots, n$, and a positive integer K.

QUESTION: Is there a partition of L into disjoint sets $L_1, L_2, \ldots, L_K$ such that $\sum_{a_i \in L_h} s_j(a_i) \leq B$ for $j = 1, 2, 3$ and $h = 1, 2, ..., K$?

Lemma 3 *3-DVPP with $s_{\max}$ large is NP-complete.*

Proof. Clearly, this problem belongs to NP. To prove it is NP-complete we show that *NUMERICAL 3-DIMENSIONAL MATCHING* (which is NP-complete, (cf.[2], p.224)) reduces to a special case of our problem where $n = 3m$ and $K = m$.

NUMERICAL 3-DIMENSIONAL MATCHING (P2)
INSTANCE: Three disjoint sets X, Y and Z, each containing m elements, a nonnegative integer size $c(a)$ for each element $a \in X \cup Y \cup Z$, and a nonnegative integer bound B.
QUESTION: Can $X \cup Y \cup Z$ be partitioned inte m disjoint sets $L_1, L_2, \ldots, L_m$ such that each L_i contains exactly one element from each set X, Y and Z such that for $i = 1, 2, \ldots, m$:

$$\sum_{a \in L_i} c(a) = B?$$

We construct the instance of (P1) based on the instance of (P2) in the following way.

- For $a \in X$, define $s_1(a) = 2B/3, s_2(a) = 0$, and $s_3(a) = c(a)/2$.
- For $a \in Y$, define $s_1(a) = 0, s_2(a) = 2B/3$, and $s_3(a) = c(a)/2$.
- For $a \in Z$, define $s_1(a) = 0, s_2(a) = 0$, and $s_3(a) = B/2 + c(a)/2$.

Now, consider a nontrivial instance of (P2), i.e. where $\max_{a \in X \cup Y \cup Z} c(a) \leq B$ and $\sum_{a \in X \cup Y \cup Z} c(a) = mB$. Hence, $c(a)/2 \leq B/2 < B$ for $a \in X \cup Y \cup Z$ and the reduced instance of (P2) is indeed an instance of (P1).

Assume now that the answer to the reduced instance is yes. Then, since each item has at least one coordinate larger than $B/2$, each set $L_1, L_2, \ldots, L_m$ contains at most three items, i.e. at most one from X, one from Y and one from Z. Further, $n = 3m$ implies that each set $L_i, i = 1, 2, \ldots m$ contains exactly three items, say $x_i \in X, y_i \in Y$ and $z_i \in Z$. Furthermore, we know that

$$s_3(x_i) + s_3(y_i) + s_3(z_i) \leq B \quad \text{for } i = 1, 2, \ldots, m. \tag{2}$$

However,

$$\sum_{a \in X \cup Y \cup Z} s_3(a) = \sum_{a \in X} c(a)/2 + \sum_{a \in Y} c(a)/2 + \sum_{a \in Z} (B/2 + c(a)/2)$$

$$= \frac{1}{2} \sum_{a \in X \cup Y \cup Z} c(a) + \frac{B}{2} \mid Z \mid = mB/2 + mB/2 = mB.$$

In consequence, (2) must be satisfied as an equality, i.e.

$$B = s_3(x_i) + s_3(y_i) + s_3(z_i) = c(x_i)/2 + c(y_i)/2 + B/2 + c(z_i)/2.$$

Hence, $c(x_i) + c(y_i) + c(z_i) = B$, for $i = 1, 2, \ldots, m$, and the partition $L_1, L_2, \ldots, L_m$ also provides a yes answer to (P2).

Conversely, if a partition $L_1, L_2, \ldots, L_m$ provides a yes answer to (P2), it follows directly from the definition of $s_i(a)$, for $i = 1, 2, 3$ and $a \in X \cup Y \cup Z$ that this partition also provides a yes answer to (P1) with $n = 3m$ and $K = m$. □

From Lemma 3, we can immediately conclude that MDVPP with at least one coordinate larger that 1/2 is NP-hard for any $m \geq 3$.

References

[1] Garey, M.R., Graham, R.L., Johnson, D.S.: Resource Constrained Scheduling as Generalized Bin Packing, *J. Combinatorial Theory (A)*, 21(1976), 257-298.

[2] Garey, M.R., Johnson, D.S.: *Computers and Intractability: a Guide to the theory of NP-Completeness*, Freeman, New York, 1979.

[3] Papadimitriu, C.H., Steiglitz, K.: *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, New Jersey, 1982.

(Received April 15,1990)