OPERATIONS RESEARCH
Vol. 26, No. 5, September-October 1978

0030-364X/78/2605-0760 $01.25

Bin Packing with Geometric Constraints in Computer Network Design

A. K. CHANDRA

IBM T. J. Watson Research Center, Yorktown Heights, New York

D. S. HIRSCHBERG

Rice University, Houston, Texas

C. K. WONG

IBM T. J. Watson Research Center, Yorktown Heights, New York

(Received October 1976; accepted January 1978)

We consider the bin-packing problem with the constraint that the elements are in the plane, and only elements within an oriented unit square can be placed within a single bin. The elements are of given weights, and the bins have unit capacities. The problem is to minimize the number of bins used. Since the problem is obviously *NP*-hard, no algorithm is likely to solve the problem optimally in better than exponential time. We consider an obvious suboptimal algorithm and analyze its worst-case behavior. It is shown that the algorithm guarantees a solution requiring no more than 3.8 times the minimal number of bins. We can show, however, a lower bound of 3.75 in the worst case. We then generalize the problem to arbitrary convex figures and analyze a class of algorithms in this case. We also consider a generalization to multidimensional "bins," i.e., the weights of points in the plane are vectors, and the capacities of bins are unit vectors.

THE classical bin-packing problem can be stated as follows: Given n numbers between 0 and 1, pack them into "bins" such that the sum of numbers in a bin does not exceed 1 and the number of bins used is minimized. This problem has been studied thoroughly (see, e.g., [7–9, 11–14, 16]) and has applications in operations research [2, 4, 6, 10], computer operating system design and memory allocation [7–9, 11, 16]. More recently, the multi-weight bin-packing problem has also been studied by various authors [8, 16].

Now all these problems find yet another application in the area of computer network design [3]. In the design of a distributed computer system, three design problems are of major importance:

(i) Processors are to be allocated so that processing requirements from terminal stations can be satisfied.
(ii) Files and application functions are to be distributed to the various processors so that all transactions can be processed efficiently.
(iii) Data communication lines are to be installed connecting terminal stations to processors.

The three design problems are interrelated, and a successful overall design should carefully consider all three aspects. However, in order to develop tools and algorithms for designing a distributed computer system, these design problems, although interrelated, can be investigated separately. The special techniques and insights obtained from such investigations can later be combined for the solution of the overall design problem.

Reference 3 studies problem (i). It also contains extensive references to publications related to problems (ii) and (iii).

Problem (i) can be formulated in terms of a bin-packing problem as follows: Let p_i, $0<p_i<1$, $i=1, \cdots, n$, be the (normalized) processing requirements of the n stations and 1 be the (normalized) processing capacity of a processor. Then problem (i) is to group stations into clusters and allocate one processor to each cluster, such that the total processing requirement for each cluster does not exceed the processing capacity and the number of clusters is minimized. If each station has multiple requirements, such as processing requirement, file requirement, traffic requirement, they can be represented by a vector $p_i=(p_{i1}, \cdots, p_{is})$, $0<p_{ij}<1$; the capacity of a processor is $(1, 1, \cdots, 1)$. Then problem (i) is to allocate processors to clusters of stations so that in each cluster the requirements do not exceed the capacity of a processor. We then have the multi-weight bin-packing problem.

In [3] experimental and statistical studies were carried out on several simple heuristic methods. It also points out the need of some notions of geographic constraints to reflect more accurately the practical situation. For example, grouping two stations thousands of miles apart into one cluster is highly undesirable.

In this paper we propose a model for the processor allocation problem with geographic constraints in terms of a bin-packing problem with geometric constraints; i.e., stations can be grouped into a cluster only when they are "close" enough.

Formally, we assume that the stations are points in a plane with weight vectors and that clusters can be formed only when the points are within a certain neighborhood. A neighborhood is a preassigned convex figure. Thus, we have the following constrained bin-packing problem: Given n points a_i, $i=1, \cdots, n$, in a plane with associated weight vectors p_i, and a bounded convex figure G, pack the points into bins such that

(i) the points in each bin can be contained in a G-figure;
(ii) the total sum of each weight component in a bin does not exceed 1; and
(iii) the number of bins used is minimized.

Of course, the G-figures of two bins may overlap. Note that by a figure G we mean the boundary of G is included and the orientation of G may be part of the definition.

Our model is only a first attempt to attack the geographic constraint problem. Obviously, many other models are possible. We choose the present model for its simplicity.

Clearly, our bin-packing problem with geometric constraints is NP-hard. Thus, efficient algorithms for optimal solutions are very unlikely. In this paper we propose a simple heuristic and compare its worst-case performance with optimal packings.

Let $H(G, a, \mathbf{p})$ denote the number of bins used by a heuristic H for the points a and weight vectors $\mathbf{p}$. Let $M(G, a, \mathbf{p})$ be the minimum number of bins needed for $(a, \mathbf{p})$. Define $\tau_H(G) = \text{lub}_{(a,\mathbf{p})} H(G, a, \mathbf{p})/M(G, a, \mathbf{p})$, where lub means "least upper bound." The subscript H in $\tau_H(G)$ will be omitted where it is obvious. We shall derive upper and lower bounds for $\tau(G)$. The case $s=1$ will be considered first and for G to be a unit square with sides parallel to the x- and y-axes, we shall show that the heuristic uses at most 3.8 times more bins than the minimum number. Furthermore, we shall construct a family of sequences $(a, \mathbf{p})$ such that $H(G, a, \mathbf{p})/M(G, a, \mathbf{p})$ approaches 3.75. Thus, $3.75 \leqq \tau(G) \leqq 3.8$ for this specific G.

Next we shall generalize these results to arbitrary convex figure G and then to the case of $s>1$.

1. THE HEURISTIC

In this section we shall assume $s=1$ and G is a unit square with sides parallel to the x- and y-axes. We define a heuristic H_1 below.

Let x_i, y_i be the coordinates of point a_i with weight p_i. In the proposed heuristic, the points will be processed from $y=\infty$ to $y=-\infty$. For two points with the same y-coordinates, the one with smaller x-coordinate will be processed first. Therefore, we start with the point having the largest y-coordinate. In case of tie, the one with the smallest x-coordinate is the choice.

(i) For point a, we construct two G-figures such that they have one vertical edge in common and point a is located at the top of this edge. (See Figure 1.) For definiteness, the left figure includes all its four edges and the point a, and the right figure includes all its edges except a.

(ii) Remove from further consideration the area covered by all the

G-figures (including boundaries) so far. Repeat (i) except that the two figures constructed will cover only the part of plane uncovered so far.

The points chosen to construct pairs of figures will be referred to as *pivot points*, and the figures constructed will be referred to as *resultant figures*. (For example, in Figure 1 a' is another pivot point and the shaded figures are its resultant figures.) Note that the resultant figures are disjoint and might not be squares.

(iii) After all the points are covered by resultant figures, we shall pack the points in a resultant figure into "bins" such that the weight sum in each bin is at most one. The packing can be done arbitrarily as long as in the resulting bins there is at most one bin with weight sum less than or equal to ½.

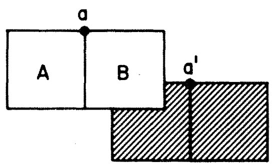


Figure 1. Two G-figures for point a are shown with one vertical edge in common and point a located at the top of this edge.

Part (i) of the above heuristic constructs two G-figures (oriented unit squares) for each pivot point. This scheme has the property that if there were no constraints on the total weights of points in a bin, the number of bins used in the heuristic would be no more than a factor of 2 times the minimum possible number of bins. This is easily seen because no two pivot points in the heuristic can be placed in one bin. The factor of 2 is tight. Suppose the heuristic were changed such that only one of these two G-figures is constructed at a pivot point, say the one in which the pivot point is at the top right-hand corner. Then the ratio of the number of bins used by the heuristic divided by the minimum number of bins would be unbounded. This applies whether or not the bins have weight constraints.

2. ANALYSIS OF THE HEURISTIC

In this section we shall show that $3.75 \leqq \tau(G) \leqq 3.8$ for H_1. (Recall that G is a unit square with sides parallel to the x- and y-axes.) To derive the upper bound, we need some definitions and lemmas.

Given a resultant figure Y, a pivot point a is said to be *near* Y if there exists a G-figure containing a and having nonempty intersection with Y.

LEMMA 1. *Let a be a pivot point with left resultant figure A and right resultant figure B. Then there are at most 4 other pivot points near A. A similar statement holds for B.*

Proof. Let A', B' be the original figures from which the resultant figures A, B are obtained. It suffices to look at the seven G-figures surrounding A' and B'. (See Figure 2.) They are labeled $S_1, \cdots, S_7$ in a counter-clockwise direction.

By the definition of resultant figures, no pivot points near A can occur in S_3, S_2, S_1 and B'. Since each G-figure can contain at most one pivot point, it follows that there are at most 4 pivot points near A. Note that the shape as well as the boundary of the G-figure plays an important role here.

A similar argument applies to B.

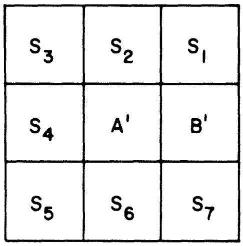


Figure 2. The seven G-figures, S_1–S_7, are shown surrounding A' and B' in a counter clockwise direction.

DEFINITION. *We say a resultant G-figure is "light" if the sum of weights of points in it is less than or equal to 1; otherwise, it is a "heavy" figure. A pivot point is "light" if both its resultant G-figures are light.*

LEMMA 2. *Given a, $\mathbf{p}$, let n_1 be the number of light resultant G-figures and n_2 the number of heavy resultant G-figures. Let m be the total number of bins used for heavy figures. Write $H = H_1(G, a, \mathbf{p})$ and $\theta = M(G, a, \mathbf{p})$ for short. Then*

$$\text{(i)}\quad m \geqq 2n_2 \tag{1}$$

$$\text{(ii)}\quad H = n_1 + m \tag{2}$$

$$\text{(iii)}\quad \theta \geqq (n_1 + n_2)/2 \tag{3}$$

$$\text{(iv)}\quad \theta \geqq m/2 + \max\{(n_1 - n_2)/2 - 4n_2, 0\}. \tag{4}$$

Proof.

(i) It is obvious.

(ii) It follows from the fact that a light G-figure needs at most one bin by definition of the heuristic.

(iii) Only notice that $(n_1 + n_2)/2$ is exactly the number of pivot points generated by the heuristic and no two such points would be included in the same G-figure (hence in the same bin) in any algorithm. Thus (3) follows.

(iv) In an optimal algorithm, to cover points in all heavy figures one needs at least $m/2$ bins. Also, there are at least $\max\{(n_1-n_2)/2, 0\}$ light pivot points. By Lemma 1 at most $4n_2$ of these light pivot points are near the n_2 heavy figures. Thus, at least $\max\{(n_1-n_2)/2-4n_2, 0\}$ light pivot points are not near any heavy figure. But each of these needs a bin to cover it in an optimal algorithm. Thus,

$$\theta \geqq m/2+\max\{\max((n_1-n_2)/2, 0)-4n_2, 0\},$$

which is (4).

THEOREM 1. *For heuristic H_1 and oriented unit squares G, $\tau(G) \leqq 3.8$.*

Proof. We shall show that the ratio H/θ will achieve the maximum value 3.8 over all real values of n_1, n_2, m.

We shall minimize θ/H subject to (1) to (4). Equation (4) can be rewritten as

$$\theta \geqq m/2+(n_1-9n_2)/2, \quad \text{for} \quad n_1 \geqq 9n_2 \tag{5}$$

$$\theta \geqq m/2, \quad \text{for} \quad n_1 \leqq 9n_2. \tag{6}$$

Substituting $n_1=H-m$ in (3), (5) and (6), we have

$$\theta \geqq H/2-m/2+n_2/2 \tag{7}$$

$$\theta \geqq H/2-9n_2/2 \quad \text{for} \quad m+9n_2 \leqq H \tag{8}$$

$$\theta \geqq m/2, \quad \text{for} \quad m+9n_2 \geqq H. \tag{9}$$

Solving the linear inequalities (1), (7), (8), we see that θ/H has minimum value 5/19. The same result holds if we solve the linear inequalities (1), (7), (9). Consequently, $\tau(G) \leqq 19/5=3.8$.

THEOREM 2. *For heuristic H_1 and oriented unit squares G, $\tau(G) \geqq 3.75$.*

Proof. We shall construct a family of examples, such that the lower bound of H/θ approaches 3.75. Take 5 unit squares with sides parallel to the x- and y-axes as in Figure 3. X represents a set of 8 arbitrarily close points with weight 2ϵ each, where $\epsilon>0$ is an arbitrarily small number. Each of A, B, C, D, represents a set of 2 arbitrarily close distinct points with weight $\frac{1}{2}-\epsilon$ each. Each of E, F, G, H represents a set of 2 points with weight ϵ each. Furthermore, A, E can be covered by a unit square, so can B, F; C, G; D, H respectively. But F, G, and G, H cannot be covered by unit squares. If the heuristic is applied to this set of points and weights, one of the points in X will be a pivot point with S_1 its left resultant figure. As a result of packing within this figure, in the worst case one may end up with 8 bins with weight $\frac{1}{2}+\epsilon$ each, namely, one point from X and one point from A, B, C, or D constitute one bin. By appropriately arranging the positions of points in E, F, G, H, each point may need one bin with a total of 8 bins. Thus $H=16$. On the other hand, one can clearly group A, E in one bin, and similarly for the pairs B, F; C, G and D, H. For X, another bin is needed. Thus, $\theta \leqq 5$ and $H/\theta \geqq 3.2$.

To improve this lower bound, we take an exact copy of the configuration in Figure 3 except in the weight: ϵ is now replaced by $\epsilon_1 = 16\epsilon$. Put this new configuration to the upper right-hand side of the old configuration such that the left-most point in F coincides with X. Eliminate this point from the combined figure (but keep the original X). Therefore, the new F consists of the set X of 8 points with weight 2ϵ each and another point with weight ϵ_1. Repeat the same process by adding a new configuration with new weights ($\epsilon_{i+1} = 16\epsilon_i$) to the upper right-hand side of the current figure.

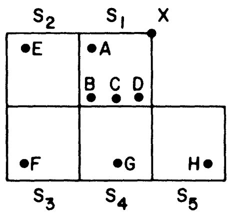


Figure 3. A typical configuration of five squares used in the proof of Theorem 2.

If we now consider a typical configuration of five squares as in Figure 3, the heuristic may yield 8 bins for the square S_1, 2 bins for each of S_2, S_4, S_5 but 1 bin for F since one of the points in F is being taken care of by a lower left-hand square. The total is 15. On the other hand, one can put A, E; B, F; C, G; D, H in 4 bins and X will be taken care of by an upper right-hand square. Thus in the limit $H/\theta \geqq 3.75$.

The heuristic H_1 used a particularly simple algorithm to pack points in a resultant figure into bins (step (iii) of H_1). If we use an optimal packing at this step, the modified heuristic still cannot guarantee using less than 3 times the minimum number of bins for the overall problem. This can be seen by modifying the construction in the proof of Theorem 2 as follows: A, B, C, D represent one point each with weight $1-2\epsilon$; E, F, G, H represent a set of 2 points each with weight ϵ; and X represents 2 points with weight 3ϵ each. This configuration is repeated with weights adjusted accordingly as in the proof of the theorem.

3. GENERALIZATION TO ARBITRARY FIGURES

In this section we still assume $s=1$ but generalize the previous results to more general geometric figures. Let G be an arbitrary bounded convex figure. The heuristic is still the same but with different parameters, namely, points will be processed from top to bottom and from left to right. If a point is picked as a pivot point, α G-figures will be generated covering at least the pivot point. After removing the area already covered, we repeat the process. We require that one bin be used even if a resultant G-figure is empty. Eliminating this restriction, however, would not change our upper bound, nor our lower bound for $\alpha=1$.

Now in addition to α, two parameters are needed: β, the maximum number of pivot points near a resultant G-figure other than the one that generates the G-figure, and γ, the maximum number of pivot points coverable by a G-figure (a pivot point A is "near" a resultant G-figure if there is a point B in the resultant G-figure such that it is possible to place a G-figure overlapping both A and B). In the previous case, $\alpha=2$, $\beta=4$, $\gamma=1$.

We have similar results as before. Proofs are given in the appendix.

LEMMA 2$'$.

(i) $m \geqq 2n_2$

(ii) $H=n_1+m$

(iii) $\theta \geqq (n_1+n_2)/\alpha\gamma$

(iv) $\theta \geqq m/2+\max\{(n_1-n_2)/\alpha\gamma-[(\alpha\beta+\alpha-2)/\alpha\gamma]n_2, 0\}$.

THEOREM 1$'$. $\tau(G) \leqq 2+\alpha\gamma-\gamma/(\beta+1)$.

The dominating term is $\alpha\gamma$. Since the case $\alpha=1$, $\gamma=1$ simultaneously is impossible, $\alpha=2$, $\gamma=1$ or $\alpha=1$, $\gamma=2$ are the minimal values. Our previous case belongs to the former and we shall exhibit a case for the latter at the end of the section.

THEOREM 2$'$. $\tau(G) \geqq \max\{2+\alpha-(\alpha+5)/(\beta+2), \alpha\gamma\}$.

We shall next present some examples for illustration.

(1) Let G be a unit square with sides parallel to the x- and y-axes. Let $\alpha=1$ and let the mid-point of the top horizontal edge be a pivot point. Then $\beta=14$, $\gamma=2$. $2.625 \leqq \tau(G) \leqq 3.87$.

(2) Let G be a unit circle. Let $\alpha=1$ and let the center of the circle be a pivot point. Then $\beta=18$, $\gamma=5$. $5 \leqq \tau(G) \leqq 6.74$.

(3) Let G be an oriented unit square. Let $\alpha=1$ and let the top left corner of G be the pivot point. Then $\gamma=\infty$ and $\tau(G)=\infty$.

(4) Let G be a unit circle. Let $\alpha=2$. The centers of the two circles are at the same y-coordinate and separated by one unit on the x-coordinate. The pivot point is the intersection of these two circles that has the larger y-coordinate. Then it can be shown that $\gamma=2$. Hence $4 \leqq \tau(G) \leqq 6$.

4. GENERALIZATION TO THE MULTI-DIMENSIONAL CASE

In this section we generalize the results of Sections 1 and 2 to the multi-dimensional case (the dimension s is an arbitrary integer $\geqq 1$), but retain the requirement that G-figures be oriented unit squares, i.e., unit squares with sides parallel to the x- and y-axes.

The heuristic H_s is the same as that in Section 1 except that (iii) is changed to (iii′) below.

(iii′) After all the points are covered by resultant figures, pack the points in a resultant figure arbitrarily into "bins" provided that:

(a) if the weight vectors of the points in a bin are $p_i = (p_{i1}, \cdots, p_{is})$, $1 \leqq i \leqq k$, then the weight vector of the bin is defined as $b = (b_1, \cdots, b_s) = (\sum_i p_{i1}, \cdots, \sum_i p_{is})$ and we require that it satisfy the constraint $b_j \leqq 1$, $1 \leqq j \leqq s$,

(b) there are no two bins with weight vectors $b = (b_1, \cdots, b_s)$ and $b' = (b_1', \cdots, b_s')$ such that $b_j + b_j' \leqq 1$, $1 \leqq j \leqq s$.

The algorithm (iii′) for multi-dimensional bin packing (without geometric constraints) is known to use bins no more than $s+1$ times the minimum possible number of bins [7]. On the other hand, no algorithm is known that is guaranteed to use less than s times the minimum possible.

We can show that the above heuristic for multi-dimensional bin packing with geometric constraints is guaranteed to use no more than $s+2.8$ times the minimum possible number of bins. We have the following theorem. (See the appendix for proof.)

THEOREM 3. *For heuristic H_s and oriented unit squares G,*

$$s + 2.75 \leqq \tau(G) \leqq s + 2.8.$$

5. CONCLUDING REMARKS

In this paper we propose and analyze a simple heuristic for the bin-packing problem with geometric constraints. An interesting problem remaining is to find better approximation algorithms for this problem. It seems that the emphasis should be placed on the allocation of points to figures rather than the bin-packing aspect since, as demonstrated at the end of Section 2, even optimal packing does not help much. Another area of interest is to consider other models that may reflect more accurately the geographic constraints in processor allocation problems.

APPENDIX

LEMMA 2′.

Proof. To prove (iv), only note that the number of light pivot points is bounded by max $([n_1 - (\alpha-1)n_2]/\alpha, 0)$ and thus

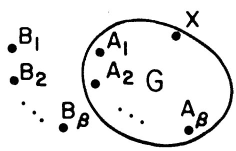
$$\theta \geqq m/2 + \max\{[\max([n_1 - (\alpha-1)n_2]/\alpha, 0) - \beta n_2]/\gamma, 0\}.$$

THEOREM 1′.

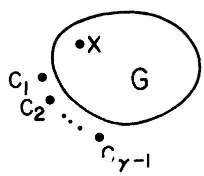
Proof. Similar to Theorem 1.

THEOREM 2′.

Proof. We can repeat essentially the construction in the first part of the proof of Theorem 2 and obtain the lower bound $2+\alpha-(\alpha+5)/(\beta+2)$. In Figure 4(a), let X be a point with weight ϵ. Each of A_i is a set of 3 points, of which 2 points have weight $(1-\epsilon)/2$ each and the third has



(a)



(b)

Figure 4. Illustration of proof of Theorem 2.

weight ϵ. Each of B_i is a point with weight ϵ. They are positioned such that one of the points in X is a pivot point in the heuristic and $\{A_i\}$, $1\leqq i\leqq\beta$, are all contained in one of the resultant G-figures generated. Furthermore, B_i are positioned such that for each i, A_i, B_i can be covered by a G-figure and each B_i is a pivot point.

In the worst case the heuristic will use 2β bins for the points A_i, an additional $\alpha-1$ bins for the pivot X, and $\alpha\beta$ bins for the pivots B_i. Thus, $H=2\beta+\alpha\beta+\alpha-1$. On the other hand, $\theta\leqq\beta+2$. Thus $\tau(G)\geqq\alpha+2-(\alpha+5)/(\beta+2)$.

On the other hand, as in Figure 4(b), suppose X, C_1, C_2, $\cdots$, and $C_{\gamma-1}$ are all points with weight ϵ each. If appropriately positioned, each of these sets can contain a pivot point, yet all can be covered by one G-figure. Thus $H=\alpha\gamma$ and $\theta=1$. It follows that $\tau(G)\geqq\alpha\gamma$.

THEOREM 3.

Upper bound. Let n_1, n_2 and m be defined as in Lemma 2. Let $H=H_s(G, a, \boldsymbol{p})$ and $\theta=M(G, a, \boldsymbol{p})$. As before,

$$m=2n_2 \tag{1}$$

$$H=n_1+m \tag{2}$$

$$\theta\geqq(n_1+n_2)/2 \tag{3}$$

$$\theta\geqq m/(s+1)+\max\{(n_1-n_2)/2-4n_2, 0\}. \tag{10}$$

Equation (10) follows from the result in [7] that if step (iii′) in heuristic H_s packs the points into, say, k bins, an optimal packing requires at least $k/(s+1)$ bins. As in the proof of Theorem 1, we can solve (1), (2), (3), (10) to obtain $H/\theta\leqq s+14/5=s+2.8$.

Lower bound. This is obtained by generalizing the construction used in the proof of Theorem 2 to $s\geqq 2$. For any $\delta>0$ we construct the following example, for which $H/\theta\geqq s+2.75-\delta$.

Let k be a natural number such that

$$k\geqq(4s-4\delta+3)/16\delta, \tag{11}$$

and let ϵ' be a positive real number such that $\epsilon'\leqq s/[(4s^2+5s)^{k+1}]$.
There are $8ks+15k+4s+5$ points

$A_{ij}, B_{ij}, C_{ij}, D_{ij}$	$1\leqq i\leqq s+1$,	$1\leqq j\leqq k$
E_{ij}, G_{ij}, H_{ij}	$i=1, 2,$	$1\leqq j\leqq k$
F_{ij}	$1\leqq i\leqq 4s+5$,	$0\leqq j\leqq k$

located at the following coordinates

A_{ij} is at $(1.8j-0.9, 1.3j-0.2)$	
B_{ij} is at $(1.8j-0.9, 1.3j-0.4)$	
C_{ij} is at $(1.8j-0.7, 1.3j-0.4)$	
D_{ij} is at $(1.8j-0.6, 1.3j-0.4)$	
E_{ij} is at $(1.8j-1.8+0.1i, 1.3j-0.2)$	
F_{ij} is at $(1.8j, 1.3j)$	$1\leqq i\leqq 4s+4$
F_{ij} is at $(1.8j+0.1, 1.3j)$	$i=4s+5$
G_{ij} is at $(1.8j-0.7+0.1i, 1.3j-1.3)$	
H_{ij} is at $(1.8j+0.4+0.1i, 1.3j-1.3)$.	

The weight vectors of the points are as follows (ϵ denotes $\epsilon'(4s^2+5s)^j$):

$A_{ij}, B_{ij}, C_{ij}, D_{ij}$ have weights $((1-\epsilon)/2, 0, \cdots, 0)$ for $i=1, 2$.

$A_{ij}, B_{ij}, C_{ij}, D_{ij}$ have weights $(\epsilon/s, \epsilon/s, \cdots, 1-\epsilon, \cdots, \epsilon/s)$ for $3 \leqq i \leqq s+1$, where all elements are ϵ/s except the $i-1$ which is $1-\epsilon$.

E_{ij}, G_{ij}, H_{ij}, have weights $(\epsilon/(2s), \cdots, \epsilon/(2s))$ for $i=1, 2$.

F_{ij} have weights $(\epsilon, \cdots, \epsilon)$ for $1 \leqq i \leqq 4s+5$.

The heuristic applied to this set of points has pivot points E_{1j}, F_{1j}, G_{1j}, H_{1j} for all j. The bins are packed as follows. One bin is used for each E_{ij}, $F_{4s+5,j}$, G_{ij} and H_{ij}. In addition, one bin is used for each of the following pairs: A_{ij} and F_{ij}, B_{ij} and $F_{i+s+1,j}$, C_{ij} and $F_{i+2s+2,j}$, D_{ij} and $F_{i+3s+3,j}$. Finally, one bin is used for all $F_{i,0}$, $1 \leqq i \leqq 4s+4$, for a total of $H=4sk+11k+2$ bins. The optimal way to pack the same set is to use for each j, one bin for all points A_{ij}, E_{ij}, $1 \leqq i \leqq s+1$, one bin for C_{ij}, G_{ij}, one for D_{ij}, H_{ij}, and one for all points B_{ij}, $F_{i',j-1}$, $1 \leqq i \leqq s+1$, $1 \leqq i' \leqq 4s+5$. Finally, one bin is used for all $F_{i,k}$, $1 \leqq i \leqq 4s+5$, for a total of $\theta=4k+1$ bins. Thus $H/\theta = (4sk+11k+2)/(4k+1) \geqq s+2.75-\delta$ from (11).

REFERENCES

1. A. V. Aho, J. E. Hopcroft, and J., D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass., 1974.
2. A. R. Brown, *Optimum Packing and Depletion*, American Elsevier, New York, 1971.
3. S. K. Chang and D. T. Tang, "Processor Allocation in a Distributed Computer System," *Proceedings of the IEEE/NBS* 1975 *Symposium on Computer Communications: Trends and Applications*, pp. 47–54 Gaithersburg, Md., 1975.
4. R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*, Addison-Wesley, Reading, Mass., 1967.
5. S. A. Cook, "The Complexity of Theorem Proving Procedures," *Proceedings 3rd Annual ACM Symposium on the Theory of Computing*, pp. 151–158 Shaker Heights, Ohio, 1971.
6. S. Eilon and N. Christofides, "The Loading Problem," *Management Sci.* **17,** 259–268 (1971).
7. M. R. Garey and R. L. Graham, "Bounds for Multiprocessor Scheduling with Resource Constraints," *SIAM J. Comput.* **4,** 187–200 (1975).
8. M. R. Garey, R. L. Graham, D. S. Johnson, and A. C.-C. Yao, "Resource Constrained Scheduling as Generalized Bih Packing," *J. Combinatorial Theory (A)* **21,** 257–298 (1976).
9. M. R. Garey, R. L. Graham, and J. D. Ullman, "Worst-Case Analysis of Memory Allocation Algorithm ," *Proceedings 4th Annual ACM Symposium on the Theory of Computing*, pp. 143–150, Denver, Colo., 1972.
10. P. C. Gilmore and R. E. Gomory, "A Linear Programming Approach to the Cutting Stock Problem II," *Opns Res.* **11,** 863–888, (1963).
11. R. L. Graham, "Bounds on Multiprocessing Anomalies and Related Packing

Algorithms," *AFIPS Conference Proceedings, Vol. 40*, pp. 205–217, *Spring Joint Computer Conference*, 1972.

12. D. S. Johnson, Near-Optimal Bin Packing Algorithms," Doctoral thesis, M.I.T., 1973.
13. D. S. Johnson, "Fast Algorithms for Bin Packing," *J. Comput. System Sci.* **8,** 272–314 (1974).
14. D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, "Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms," *SIAM J. Comput.* **3,** 299–325 (1974).
15. R. M. Karp, "Reducibility among Combinatorial Problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher (eds.), pp. 85–104, Plenum Press, New York, 1972.
16. A. C.-C. Yao, "A Study of Concrete Computational Complexity," Technical Report UIUCDCS-R-75-716, Department of Computer Science, University of Illinois at Urbana-Champaign, May 1975.