SIAM J. COMPUT.
Vol. 9, No. 4, November 1980

0097-5397/80/0902-0015$01.00/0

ORTHOGONAL PACKINGS IN TWO DIMENSIONS*

BRENDA S. BAKER†, E. G. COFFMAN, JR.† AND RONALD L. RIVEST‡

Abstract. We consider problems of packing an arbitrary collection of rectangular pieces into an open-ended, rectangular bin so as to minimize the height achieved by any piece. This problem has numerous applications in operations research and studies of computer operation. We devise efficient approximation algorithms, study their limitations, and derive worst-case bounds on the performance of the packings they produce.

Key words. two-dimensional packing, bin packing, resource constrained scheduling

1. Introduction. Efficiently packing sets of rectangular figures into a given rectangular area is a problem with widespread application in operations research. Thus, one is inclined to attribute the scarcity of results on this problem, and others of its type, to inherent difficulty rather than to lack of importance. Motivated by the intractability of these problems, we define and analyze certain approximation algorithms. These algorithms are natural in the sense that they would probably be among the first to occur to anyone wishing to design simple, fast procedures for determining easily computed packings. The analysis of these algorithms leads to bounds on the performance of approximate packings relative to the best achievable.

In the remainder of this section we define the model to be studied and introduce notation. At that point we examine in more detail the applications which are served by the model, and we review the literature bearing on this and similar models. In § 2 the main results of the paper are presented and proved. Concluding remarks and a discussion of open problems are given in § 3.

As illustrated in Fig. 1, we consider an "open-ended" rectangle, R, of width w and a collection of rectangles, also called pieces, organized into a list $L=(p_1,p_2,\cdots,p_n)$. Each piece is defined by an ordered pair $p_i=(x_i,y_i), 1\leqq i\leqq n$, corresponding to the horizontal (x_i) and vertical (y_i) dimensions of the rectangle.

We are concerned with the packing or assignment of the pieces in L into R so as to minimize the *height*, h, of the packing; i.e., the maximum height, measured from the bottom edge of R, of the space occupied by any piece in the packing (see Fig. 1). In addition to the implicit requirement that the spaces occupied by distinct pieces be disjoint, we restrict attention to packings that are *orthogonal* and *oriented*. An orthogonal packing is one in which every edge of every rectangle is parallel to either the bottom edge or the vertical edges of R. An orthogonal packing is also oriented if the rectangles are regarded strictly as ordered pairs; i.e., a rectangle (x_i,y_i) must be packed in such a way that the edges of length x_i are parallel to the bottom edge of R. Thus, rotations of 90° (which preserve orthogonality) are not allowed.

Returning to applications we see that our model applies to industrial or commercial situations in which objects are to be packed on floors, shelves, truck beds, etc., where concern is limited to the objects in two prespecified dimensions. Another important application concerns systems containing a shared resource. A prime exam-

*Received by the editors July 3, 1978, and in final form March 28, 1980.
†Bell Laboratories, Murray Hill, New Jersey 07974.
‡Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139.

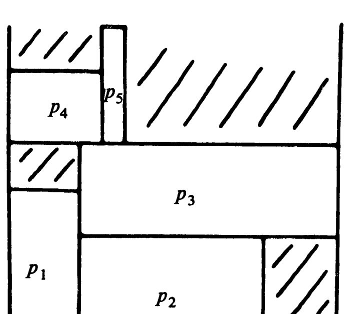
ple is the main memory resource in multiprogrammed computer systems. In such systems a number of tasks compete for a resource which they can share, but only within the limit provided by the total amount of resource available.

This application of the model was defined almost 20 years ago by E. F. Codd [1] in a study of multiprogramming systems. More recently, Garey and Graham [2] considered a related problem oriented to multiprocessor systems. In their study arbitrary numbers of processors and additional resources were considered. The analysis focused on worst-case bounds on the ratios of schedule-lengths (packing heights) for arbitrary lists; approximation algorithms were not considered. Moreover, the model of resources is basically different: Whenever an amount of the resource is available, no matter how it is configured, it can be used to satisfy any demand no greater than this amount; i.e., fragmentation of the resource is not a consideration.

Little else appears to have been published which bears on the packing problem we have defined. Erdös and Graham [3] have shown that orthogonal packings of squares into rectangles are not always optimum; i.e., there exist examples for which all orthogonal packings have greater height than the minimum achievable by exploiting the ability to rotate the squares. Based on earlier work of Meir and Moser [6], Kleitman and Krieger have considered the problem of finding a smallest rectangle into which a collection of squares can be packed [4], [5]. Specifically, they prove that a $\sqrt{2}\times 2/\sqrt{3}$ rectangle is always sufficient to pack a set of squares whose cumulative area is unity, and that no rectangle of smaller area can have this property.

Even when $x_i = x_j$ for all i and j, our packing problem is intractable; it can be shown that it becomes the NP-complete make-span minimization problem [7]. Hence, we are moved to consider fast heuristics and how closely the packings they produce approach optimum packings. For this purpose we define the following class of packing algorithms, to be called *bottom-up left-justified* (or simply BL) algorithms. (Recall that pieces must be packed so as to preserve oriented, orthogonal packings.) Each such algorithm packs the pieces one at a time as they are drawn in sequence from the list L. When a piece is packed into R it is first placed into the lowest possible location, and then it is left-justified at this vertical position in R. In the sequel R will also be referred to as a bin.

Fig. 1 shows a BL packing. Note that from a combinatorial point of view our problem remains essentially unchanged if we replace left-justification by right-justification and consider BR packings instead. Note also that two BL algorithms differ only in the ordering of L.



$L=(p_1, p_2, p_3, p_4, p_5)$

x_1: 3 8 11 4 1

y_1: 6 4 4 3 5

FIG 1. *Two-dimensional packing.*

2. Performance bounds for BL packings. We shall see that the basic BL algorithm, using a poorly ordered list L, can perform arbitrarily badly relative to an optimization algorithm. Thus, it is natural to inquire about the improvement possible by ordering L on the basis of some simple measure of piece size. Some obvious orderings to consider are increasing height, decreasing height, increasing width, and decreasing width. With the proper ordering the improvement can indeed be striking, as we shall see. However, with a badly chosen ordering, we can be just as poorly off as before. In particular, this will be true if we order pieces by increasing width (or decreasing height).

As a matter or notation let h_{BL} and h_{OPT} denote the respective heights of a BL and optimum packing of a list L which will always be clear by context.

THEOREM 1. *For any $M>0$, there exists a list of pieces ordered by increasing width such that $h_{\mathrm{BL}}/h_{\mathrm{OPT}}>M$.*

Proof. We shall define a class of lists which proves this result. First, let $k \geqq 2$ be given and define $r_i = \max\{m \mid i \equiv 0 \bmod k^m\}$, $i>0$. Thus, $r_i=1$ if i is a multiple of 4 but not 16, $r_i=2$ if i is a multiple of 16 but not 64, $r_i=3$ if i is a multiple of 64 but not 256, etc.; $r_i=0$ if i is not a multiple of 4.

Let the bin width be $w=k^k$ and let $s=k^{k-1}$. Rectangles are packed in the order given. Along the bottom row we pack k^k unit-width pieces, the ith of which has height $1-r_i\varepsilon$, where ε is much smaller than 1. The remaining rectangles all have unit height but widths in the order given by

$$\begin{array}{ll}
s \text{ of width } 1 & \text{(the second row)} \\
s/k \text{ of width } k & \text{(the third row)} \\
s/k^2 \text{ of width } k^2 & \text{(the fourth row)} \\
\vdots & \\
s/k^{k-1}=1 \text{ of width } k^{k-1} & \text{(the } k+1\text{)st row).}
\end{array}$$

Since $r_i=0$ for i not a multiple of k, one obtains the "notching" structure illustrated in the first row of Fig. 2; i.e., every kth piece is lower than the intervening $k-1$ pieces of unit height. Thus, the $s=k^{k-1}$ unit squares of the second row are placed on top of every kth piece of the first row. Now every kth piece of the second row corresponds to every (k^2)th piece of the first row and reaches a height at most $2-2\varepsilon$ which is less than that of the intervening $k-1$ pieces of the second row, all at height $2-\varepsilon$. Thus, the third row left-justifies pieces of width k over each of the lower pieces of the second row. Note that a width exceeding $k-1$ is necessary so that the piece width exceeds the width of spaces in the second row.

A similar pattern applies to the heights reached in the third row: The height reached by every kth piece is determined by that of every (k^3)th piece of the first row. Thus, in the third row every kth piece achieves a height at most $3-3\varepsilon$ while the remaining pieces all achieve a height of $3-2\varepsilon$. It follows that the fourth row left-justifies pieces of width k^2 over each of the lower pieces of the third row. In general, then, the spaces in the $(j+1)$st row ($j \geqq 1$) all have width $k^j(k-1)$, the s/k^j pieces have width k^j, and every kth piece reaches a height less than the others. Thus, when the next s/k^{j+1} pieces of width k^{j+1} are added, they are placed on top of every

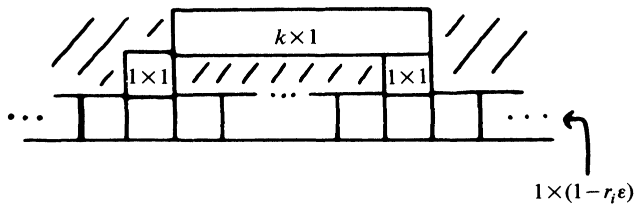


FIG 2. *The increasing widths example.*

kth piece in the $(j+1)$st row and each abuts the piece to the left in the $(j+1)$st row. Overall, the pattern looks like Fig. 2, drawn for $k=4$. (Distinctions of $O(\varepsilon)$ are not all represented, because of scaling.)

Since there are $k+1$ rows, we see that $h_{\mathrm{BL}}=k+1-O(k\varepsilon)$. But a different packing can be found which packs rows 2 through $k+1$ into one row of height 1. (Note that the sum of widths of all pieces in rows 2 through $k+1$ is $k^{k-1}\times 1+k^{k-2}\times k+\cdots+1\times k^{k-1}=k^k=w$.) Therefore, an optimum packing has a height not exceeding 2. Hence, we obtain a BL packing at least $k/2$ times higher than an optimum packing. Since k is arbitrary, the result follows. □

A dramatic improvement in the performance of BL packings is obtained when the list of rectangles is ordered by decreasing width. In fact, the ratio of BL to optimum packing height is guaranteed to be no worse than 3 when L is in decreasing order by width. For the case of squares, where decreasing width is equivalent to decreasing height, the bound is further reduced to 2. First, we shall show that the bounds of 3 and 2 can be approached as closely as desired; thus, these bounds are best possible.

THEOREM 2. *For any $\delta>0$ there exists a list L of rectangles ordered by decreasing width such that the* BL *packing gives a height h_{BL} for which*

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}}>3-\delta. \tag{1}$$

If the pieces are restricted to squares then an L can be found such that for any $\delta>0$

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}}>2-\delta. \tag{2}$$

Proof. We shall prove the second result first, since the first result is but a slight modification.

The list proving (2) corresponds to the "checkerboard" packing in Fig. 3. The pieces are all either unit squares or approximately 2×2 squares. In particular, the larger squares are disposed on the bottom of the bin with the dimensions stepping down by ε from piece to piece. Hence the assignment of pieces in the second row must be made from right to left according to the bottom-up rule. Since two unit squares exceed the dimension of any larger square and squares are left-justified, only one unit square is placed on each large square. Except for the first and last pieces, this type of assignment repeats on the second row since the "holes" in the second row all have width less than 1 and squares to the right are lower than squares to the left. In general,

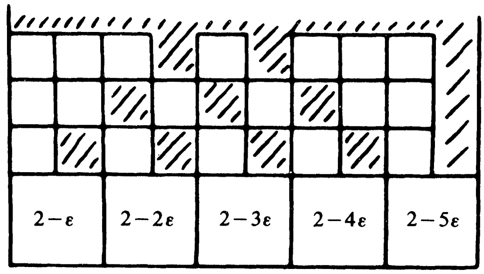


FIG 3. *The checkerboard example.*

the ith row of unit squares alternates holes and pieces except for at most the initial and final $i-1$ pieces of the row. Note that an optimum packing can be found which, except for possibly the last row, is within $O(\varepsilon)$ of being fully occupied.

The edge effects inhibiting the waste of half the space in the BL packing consist of

1. the row of larger pieces on the bottom, and
2. the triangular-shaped solidly packed collections of squares on the left and right of the packing.

Holding piece sizes constant, the influence of the first edge effect is reduced by increasing the height of the packing, while the second is attenuated by widening the bin. Let k be two greater than the number of rows of unit squares. If the width of the bin is selected to be k^2, then the area of the bottom row and side edge effects is $O(k^2)$. Thus, ignoring $O(\varepsilon)$ terms, we can find a list such that

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}}=\frac{k^3}{k^3/2+O(k^2)}.$$

In the limit $k\to\infty$, we have the bound of 2.

For the case of rectangles, it is only necessary to augment the list for Fig. 3 by adding as a new, last piece a rectangle of unit width and a height which equals the height of the optimum packing corresponding to the new list. Omitting the details, the BL packing will correspond to Fig. 3 with the new piece placed on top. It is easy to verify that the height ratio can now be made to approach 3 as closely as desired. □

THEOREM 3. *Let L be a list of rectangles ordered by decreasing widths. Then*

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}}\leqq 3. \tag{3}$$

This bound is best possible in the sense of Theorem 2.

Proof. Let h^* denote the height of the lower edge of a tallest piece whose upper edge is at height h_{BL}. If y denotes the height of this piece, then $h_{\mathrm{BL}}=y+h^*$. Let A denote the region of the bin up to height h^*.

Suppose we can show that A is at least half occupied. Then we have $h_{\mathrm{OPT}}\geqq \max\{y,h^*/2\}$; hence, $y>h^*/2$ implies

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}}\leqq\frac{y+h^*}{y}<\frac{y+2y}{y}=3,$$

and if $y \leqq h^*/2$, we have

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}} \leqq \frac{h^*/2+h^*}{h^*/2} = 3.$$

The result will thus be proved. It remains to show that A is at least half occupied.

Any horizontal cut or line through A can be partitioned into alternating segments corresponding to cuts through unoccupied and occupied areas of the BL packing. We shall show that the sum of the occupied segments is at least the sum of the unoccupied segments. For convenience we may restrict ourselves to lines which do not coincide with the (upper or lower) edges of any piece. Since the set of such lines is of measure zero, ignoring them will not influence our claim that A is at least half occupied.

Initially, consider the partition of a given line just prior to when the first rectangle, say q, is assigned with a lower edge at a height exceeding the height, h, of the line. The piece q need not be in A; its existence is guaranteed by the fact that there is a piece packed above A. We claim that at that point in the assignment sequence the line is "half occupied."

First, bottom-up packing implies that all lines must cut through at least one piece. Second, all lines must cut through a piece abutting the left bin edge. For suppose not; then the left-most piece, say q', cut by the line must abut another piece, say q'', to the left and entirely below the line. Thus, the length, x, of the unoccupied, initial segment of the line must be at least the width of q''. But since q'' was packed prior to q, the width of q must be less than that of q'' and hence less than x. Since at the point in time we are considering, no piece has been assigned entirely above the line, the space vertically above the initial segment must be completely unoccupied. Thus, we have the contradiction that q would have fit into the space above q'' in such a way that its lower edge is at a height less than h.

Now consider any segment, S, of the line which cuts through an unoccupied space. Let p be the piece bordering S on the left. Since when q is assigned, it is placed above the line, q must be wider than the length of S. (Once again, at the time q is assigned its height could not prevent its placement in a sufficiently wide unoccupied space cut by the line.) But q is packed later than p; consequently, p is at least as wide as q. It follows that for each segment representing unoccupied space along the line there is a longer segment representing occupied space immediately to its left. Clearly, for any given line, the sum of the segment lengths corresponding to unoccupied space must be monotonically nonincreasing as the packing sequence progresses. Therefore, the line continues to be at least half occupied. Finally, "integration" over the height of A verifies that A is at least half full. □

COROLLARY 1. *If L in the statement of Theorem 3 consists only of squares, then*

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}} \leqq 2.$$

This bound is best possible in the sense of Theorem 2.

Proof. First, define A' as the area extending from height y to height $h_{\mathrm{BL}}-y$, where y is the size of the tallest rectangle (now square) assigned above A in Theorem 2. Let p denote this square. As in Theorem 2, $A' \subset A$ is shown to be at least half occupied. But now, if w denotes the width of the bin, we observe that the cumulative occupied area of the upper and lower $w \times y$ slabs of the packing is at least wy and hence they are (when considered together) half occupied. This follows from the facts that p is at most as large as any square on the bottom of the bin, the bottom of the bin is full except possibly for a space at the right end, and the area of p must exceed y times the width of this space. Hence, the entire packing is at least half occupied from 0 to h_{BL}. It follows immediately that $h_{\mathrm{BL}} \leqq 2h_{\mathrm{OPT}}$. $\square$

We have seen that a BL algorithm can yield reasonably good packings when the list of pieces is sorted into decreasing order by width. That is, Theorem 3 shows that the bin height used is no more than three times the optimal bin height, and if the pieces are squares, then the packing can be no more than twice as high.

A natural question to ask next is, "For every set of pieces, is there *some* ordering of those pieces into a list such that the BL rule, when applied to that list, yields an optimal packing?" Our checkerboard example, which showed that a list of squares sorted into decreasing order by size can use up to twice as much space as an optimal packing, can be packed optimally by the BL algorithm if the list is sorted into *increasing* order by size. While it might be difficult in practice to actually determine an ordering for which the BL rule produces an optimum packing, it would be comforting to know that one was not excluding the possibility of finding an optimum packing by considering only bottom-up packings.

Unfortunately, there are sets of pieces for which no BL packing is optimal. That is, no matter what ordering is used, the BL algorithm will produce a suboptimal packing. In fact we shall present an example using squares only, which demonstrates that an optimal packing can be as little as $\frac{11}{12}$ the height of the best bottom-up packing.

THEOREM 4. *There exist sets of squares such that the ratio of the bin height used by the best bottom-up packing to that of an optimum packing is at least* $12/(11+\varepsilon)$ *for any sufficiently small* $\varepsilon > 0$.

Proof. Consider the set of squares of sizes (6,6,5,5,4,4,3,1,1) and a rectangle of width 15. An optimum packing, of height 11, is shown in Fig. 4. We first demonstrate that (up to obvious left-right symmetries) this is the only optimum packing, and then we modify the example slightly to obtain the theorem.

Since Fig. 4 is a tight packing, any optimum packing must have height 11. For an arbitrary optimum packing consider the 15×11 rectangle A that it packs to be divided into 15 disjoint 1×11 vertical slabs. Let the type of a slab be an ordered list of the sizes of the squares that the slab intersects. The only possible types are:

(a) 6-5
(b) 6-4-1
(c) 6-3-1-1
(d) 5-5-1
(e) 5-4-1-1
(f) 4-4-3

Let a denote the number of slabs of type (a), etc. The following equations must then hold, for a,b,c,d,e,f nonnegative integers:

$$\begin{aligned}
a+b+c+d+e+f &= 15\\
a+b+c &= 12\\
a+2d+e &= 10\\
b+e+2f &= 8\\
c+f &= 3\\
b+2c+d+2e &= 2
\end{aligned}$$

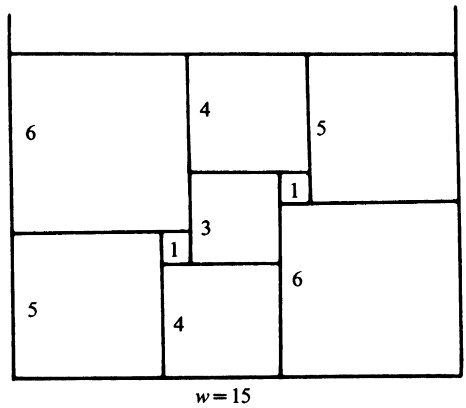


FIG 4. *Optimum vs. best* BL *packings.*

The first equation reflects the fact that there is a total of 15 slabs; the remaining five equations account for the presence of the squares of sizes 6,5,4,3,1 respectively. For example, the second equation states that the presence of exactly two 6×6 squares requires 12 slabs of types with 6's in them. By the last equation we have that either $c=0$ or $c=1$. Choosing $c=1$ yields a contradiction, and $c=0$ gives us the unique solution to the above equations:

$$\begin{aligned}&a=10\\&b=2\\&c=d=e=0\\&f=3.\end{aligned}$$

Any solution having the slab types in the above numbers must look like either Fig. 4 or its reflection. This can be proved by observing that (since $a=10$), each 5×5 is entirely above or below a 6×6. Take a particular 5×5 and consider the slab which intersects the adjacent 6×6 but does not contain the 5×5. This must be of type 6-4-1 or 6-5 (using the other five). The 6-5 possibility can't happen (since there would be a 6×3 unfilled space next to the other 6×6), and the pieces of the 6-4-1 must occur in the order 6,1,4 to prevent an unfillable gap between the 4×4 and the edge of A. The 4-4-3 slabs must come next: the pieces must be in the order 4,3,4 to leave room for the 6×6. Finally, the 6-4-1 and 6-5 slabs complete the picture. Thus, Fig. 4 represents the only way to pack the given squares into a 15×11 rectangle.

We now modify things so that (i) the 3×3 now has size $(3+\varepsilon)\times(3+\varepsilon)$ and (ii) the bin now has width $15+\varepsilon$. The preceding proof shows that (to within ε) the packing in Fig. 4 is still optimum. However, we note that in the modified packing there must be gaps on the bottom row between the 5×5 and 4×4 and between the 4×4 and 6×6 which add up to ε; otherwise the 1×1 and $(3+\varepsilon)\times(3+\varepsilon)$ will not be able to fit on top of the 4×4. Since no BL rule can produce those gaps, the optimum packing of Fig. 4 is unachievable. The best that such an algorithm can do is a packing of height 12. Thus we have

$$\frac{h_{\mathrm{BL}}}{h_{\mathrm{OPT}}}\geqq\frac{12}{11+\varepsilon}$$

for this example. □

As a final technical result we make an observation also made by R. E. Tarjan; viz., that the example of Theorem 1 can be modified to show that ordering the list by decreasing height can also lead to packings that are arbitrarily bad relative to the optimum.

THEOREM 1′. *For any $M > 0$, there exists a list of pieces ordered by decreasing height such that*

$$\frac{h_{\text{BL}}}{h_{\text{OPT}}} > M.$$

Proof. Let $k > 1$ be given and let the bin width be $w = k^k$. We shall define the BL packings proving the theorem by specifying the pieces row by row. The first two rows will each consist of w unit-width pieces. We shall define sequences $\{\delta_i\}$ and $\{\delta_i'\}$ such that the heights of pieces in the first row are given by

$$a_i = 1 + \delta_i, \qquad 1 \leqq i \leqq w,$$

and the heights of pieces in the second row are, indexed from left to right,

$$b_i = 1 + \delta_i', \qquad 1 \leqq i \leqq w.$$

The δ_i and δ_i' sequences are defined below so that $a_1 < a_t + b_t$ and

$$a_1 \geqq a_2 \geqq \cdots \geqq a_t \geqq b_t \geqq b_{t-1} \geqq \cdots \geqq b_1 \geqq 0. \tag{4}$$

Thus, the pieces will be packed in the order given by (4), the piece of height b_i will be on top of the piece of height a_i, $1 \leqq i \leqq w$, and the cumulative heights achieved in the second row will be $h_i = 2 + \delta_i + \delta_i'$, $1 \leqq i \leqq w$.

As in the proof of Theorem 1, define $r_i = \max\{m \mid k^m \text{ divides } i\}$. Thus, if i is a multiple of k but not k^2, then $r_i = 1$; if i is a multiple of k^2 but not k^3, then $r_i = 2$; etc. Clearly, $r_i = 0$ if i is not a multiple of k. Note that $\max_{1 \leqq i \leqq w}\{r_i\} = r_w = k$. Next, define

$$\delta_i = 1 - i/2w, \qquad i \leqq i \leqq w,$$
$$\delta_i' = i/2w - r_i\varepsilon, \qquad 1 \leqq i \leqq w,$$

where $0 < \varepsilon < 1/2wk$, and hence $\delta_i' \geqq 0$, $1 \leqq i \leqq w$. Note that $\delta_i > \delta_{i+1}$, $1 \leqq i < w$, and $\delta_w = \frac{1}{2} \geqq \delta_w' = \frac{1}{2} - r_w\varepsilon$. Since $\delta_{i+1}' - \delta_i' = 1/2w - (r_{i+1} - r_i)\varepsilon \geqq 1/2w - k\varepsilon > 0$, the ordering in (4) follows. Moreover, the cumulative heights in the second row are $h_i = 3 - r_i\varepsilon$. Note that these heights have the same notching effect as the heights of the first row of pieces in Theorem 1.

Let $s = w/k = k^{k-1}$. As in Theorem 1 the remaining rectangles will be of height 1, with widths in the order given by:

$$\begin{array}{ll} s & \text{of width } 1, \\ \dfrac{s}{k} & \text{of width } k, \\ \dfrac{s}{k^2} & \text{of width } k^2, \\ & \vdots \\ \dfrac{s}{k^{k-1}} = 1 & \text{of width } k^{k-1}. \end{array}$$

The pieces pack in a pattern similar to that of Theorem 1, using a total height of $k + 3 - O(k\varepsilon)$.

Note that a different packing could pack all the pieces in rows 3 and above into one row of height 1, and the remaining pieces into two rows of height 3 as in the above packing. Thus an optimum packing has a height no greater than 4. Therefore, the bottom-up packing is at least $k/4$ times higher than an optimum packing. Since k is arbitrary, the result follows. $\square$

3. Conclusions. This paper is but a beginning in the study of fast, effective approximation algorithms for packing pieces in two dimensions. We have seen that, although performance of these algorithms can be very poor, simple measures such as ordering on piece size can produce algorithms with much more reasonable performance. Indeed, with such algorithms it appears that worst-case performance can only be approached by essentially pathological cases.

Subsequent to the work [8] on which this paper is based, considerable activity has arisen in two-dimensional bin-packing. Performance bounds have been found for so-called level-oriented algorithms [9], [10], which in terms of worst-case performance are superior to the bottom-up algorithms. Also, examples have been found [11] which show that a best BL packing can be as much as $5/4$ worse than the optimum packing.

Many open problems related to our models remain for future research. Questions that should be resolved are those connected with the specialization to squares and those arising from list orderings we have not considered. For example, what is a tight bound for the special case of increasing squares? Another question concerns the implementation of the BL algorithms. The complexity of such algorithms for decreasing widths is open. How does one efficiently maintain the structure of available space as the packing sequence progresses?

REFERENCES

[1] E. G. CODD, *Multiprogram scheduling*, Comm. of the ACM, Parts 1 and 2, 3 (1960), pp. 347–350, Parts 3 and 4, 3 (1960), pp. 413–418.

[2] M. R. GAREY AND R. L GRAHAM, *Bounds on multiprocessing scheduling with resource constraints*, this Journal, 4 (1975), pp. 187–200. (See also Chap. 4 of [7]).

[3] P. ERDŐS AND R. L. GRAHAM, *On packing squares with equal squares*, Tech. Rep. STAN-CS-75-483, Computer Science Dept., Stanford University, March, 1975.

[4] D. J. KLEITMAN AND M. M. KRIEGER, *An optimal bound for two-dimensional bin-packing*, Proc. 16th Annual Symposium on Foundations of Computer Science, Los Angeles, Oct. 1975.

[5] D. KLEITMAN AND M. KRIEGER, *Packing squares in rectangles; I*, Annals of the New York Academy of Sciences, Vol. 175, Article 1, pp. 253–262, July, 1970.

[6] A. MEIR AND L. MOSER, *On the packing of squares and cubes*, Journal of Combinatorial Theory, 5 (1968), pp. 126–134.

[7] J. D., ULLMAN, *Complexity of sequencing problems*, Computer and Job-shop Scheduling Theory, E. G. Coffman, Jr. (ed), John Wiley and Sons, New York, 1975.

[8] B. S. BAKER, E. G. COFFMAN, JR., AND R. L. RIVEST, *Orthogonal packings in two dimensions*, Proc. Allerton Conf., 1978 University of Illinois, Champaign-Urbana.

[9] E. G. COFFMAN, JR., M. R. GAREY, D. S. JOHNSON AND R. E. TARJAN, *Performance bounds for level-oriented two-dimensional packing algorithms*, this Journal, this issue, pp. 808–826.

[10] DANIEL D. K. D. B. SLEATOR, *A 2.5 times optimal algorithm for packing in two dimensions*, Inform. Process. Lett. 10 (1980), pp. 37–40.

[11] D. BROWN, private communication.