数理解析研究所講究録
第 471 巻 1982 年 146-159

ALGORITHMS FOR CERTAIN PACKING PROBLEMS

Takao Ozawa and Atsuyuki Kakehi
小澤 孝夫　筧 敦行
Department of Electrical Engineering, Kyoto University

1. Introduction

Procedures using rectangular dissection have been proposed for layouting an electronic circuit on a planar semiconductor chip(1)(2). In the procedures circuit modules(components) are assigned to elementary rectangles obtained by the dissection. Unfortunately very often the sizes of elementary rectangles become far larger than those of modules assigned to them. Therefore a compaction procedure(3) is necessary to reduce the whole chip size. The layout compaction can be regarded as a special packing problem(4) with certain extra constraints. Especially the compaction should not harm the planarity of the circuit. Although the final object is to minimize the size of the chip where the circuit is laid out, it seems there is little hope to find efficient algorithms for achieving this goal. Here two packing problems approximating the layout compaction are formulated, and heuristic algorithms for solving the problems are presented.

2. Problem Formulation

Let R be the rectangle representing the chip. We quantize R and divide it into H horizontal bands. A band is a rectangle with unit height and the width equal to that of R. See Fig. 1.

The bands are numbered from 1 to H, the uppermost band being No. 1 and the lowermost one, No. H. Let $M_k(k=1,2,..,N)$ be a rectangular module with quantized height h_k and width w_k. We require M_k be placed within the region which consists of bands from b_{1k} to b_{2k}. The horizontal edges of M_k must be on the horizontal edges of some bands. If the height of the region(=$b_{2k}-b_{1k}+1$) is larger than that of M_k, we have freedom to choose the bands on which M_k is placed. Let y_k be the uppermost of the bands on which M_k is placed. We must have $b_{1k} \leq y_k \leq b_{2k}-h_k+1$.

Suppose a rectangular dissection is obtained for the given circuit. This means that b_{1k} and b_{2k} for $M_k(k=1,2,..,N)$ are fixed as well as the relative position of the modules. An example of such a dissection together with modules assigned to elementary rectangles is shown in Fig. 2. The blank spaces in rectangles are redundant(Wires connecting the modules are laid out on the spaces, and therefore they are not totally redundant.), and we want to reduce such spaces by comaction.

Our first formulation approximating the layout compaction is as follows.

Pl. Choose y_1, y_2, .., y_N to minimize

$$\max_{1\ j\ H} W_j, \text{ where } W_j = \sum_{k=1}^{N} w_k^* : w_k^* = w_k \text{ if } y_k \leq j \leq y_k+h_k-1$$
$$=0 \quad \text{otherwise}$$

subject to $b_{1k} \leq y_k \leq b_{2k}-h_k+1$ for k=1, 2, .., N.

The condition $y_k \leq j \leq y_k+h_k-1$ means that module M_k is placed on band j, and W_j is the total of the widths of the modules placed on band j. When we determine y_1, y_2, .., y_N by solving

P1, we move the modules to left(or right) as much as possible along the horizontal edges of bands while keeing their relative position as determined by the dissection. By doing this we remove redundant spaces between modules. In general the width of R thus achieved is more than the maximum of W_j's, since there can be waste spaces between modules, as shown in Fig. 3, but W_j includes only the spaces occupied by the modules on band j.

In our second formulation the waste spaces stated above are taken into consideration. This is not an easy task and the cost for achieving this is to prefix the order in which the positions of modules are determined.

P2. Choose y_k for k=1, 2 ,.., and N to minimize

$$\max_{1\leqq j\leqq H} V_j^{(N)}, \quad \text{where } V_j^{(0)}=0$$

$$V_j^{(k)} = \max_{1\leqq i\leqq h_k} V_{y_k+i-1}^{(k-1)}+w_k \quad \text{if } y_k\leqq j\leqq y_k+h_k-1$$

$$= V_j^{(k)} \qquad \text{otherwise,}$$

subject to $b_{1k}\leqq y_k\leqq b_{2k}-h_k+1$ for k=1, 2, .., and N.

In our second formulation modelu M_k is moved to right as much as possible along the horizontal edges of bands, as soon as its vertical position y_k is determined. $V_j^{(k)}$ is the position, measured from the left edge of R, of the right edge of the leftmost module on band j after M_k is moved to right(See Fig. 4), and

$$\max_{1\leqq i\leqq h_k} V_{y_k+i-1}^{(k-1)}$$

is the position of the right edge of the leftmost module on bands y_k, y_k+1, ..., and y_k+h_k-1 before M_k is placed on bands.

3. Algorithms

Since the number of modules in a circuit is very large, it is essential to use efficient algorithms for layouting them. Besides the formulations obtained in the previous section themselves are approximations of layout compaction. Therefore we do not stick to obtaining an optimal solution of P1 or P2, but rather aim to algorithms which can achieve good overall performance. Those presented in this section are so called heuristic algorithms.

We note that the vertical positions y_k's of modules must be fixed one by one in any event. In fixing the position of one module, the problem is how can the remaining modules be taken into consideration. Because of the reasons stated above, we avoid iterative procedure, that is, once the position of a module is fixed, it will not be changed again. Instead we set up two types of criterion in fixing the position. The first one is the possible maximum of the total width of modules laid out on a band. This value, denoted by U_j for band j, is first computed in step 1 (1) of Algorithms 1 and 2 below. When the position of a module is determined, this value is changed according to it.

The second criterion is a probabilistic one. Let $p_j(k)$ be the probability of module M_k being laid out on band j. We assume that the possibile positions of M_k one bands from b_{1k} to b_{2k} are equally probable, and we have:

$$p_j(k) = (j-b_{1k}+1)/B_k \quad \text{if } b_{1k} \leqq j \leqq b_{1k}+h_k-1$$
$$= h_k/B_k \quad \text{if } b_{1k}+h_k \leqq j \leqq b_{2k}-h_k$$
$$= (b_{2k}-j+1)/B_k \quad \text{if } b_{2k}-h_k+1 \leqq j \leqq b_{2k}$$
$$= 0 \quad \text{otherwise}$$

where $B_k=b_{2k}-b_{1k}-h_k+2$. See Fig.5. Algorithms 1E and 2E below use this criterion.

Algorithm 1 for P1.

step 1. (1) Set $K \leftarrow \{1,2,...,N\}$.

(2) For $j=1, 2,...,$ and N compute

$$U_j = \sum_{k=1}^{N} w_k^* \quad \text{where } w_k^* = w_k \quad \text{if } b_{1k} \leqq j \leqq b_{2k}$$
$$= 0 \quad \text{otherwise}$$

step 2. Choose $k \in K$ such that

$$\max_{y_k \leqq j \leqq y_k+h_k-1} U_j \quad \text{subject to } b_{1k} \leqq y_k \leqq b_{2k}-h_k+1$$

is minimum. Fix y_k.

step 3. For $j=b_{1k}, b_{1k}+1,...,$ and b_{2k}, set

$$U_j \leftarrow U_j - w_k \quad \text{if } b_{1k} \leqq j \leqq y_k-1 \text{ or } y_k+h_k \leqq j \leqq b_{2k}$$
$$\leftarrow U_j \quad \text{otherwise}$$

step 4. Set $K \leftarrow K-\{k\}$. If $K=\emptyset$ stop. Otherwise go to step 2.

If there are more than one module which give the minimum of max U_j at step 2, the module which gives the minimum of the next to max U_j is chosen. Further ties are broken in the same way.

Algorithm 1E for P1.

step 1. (1) Set $K \leftarrow \{1,2,\ldots,N\}$.

(2) For j=1, 2,...., and N compute

$$U_j = \sum_{k=1}^{N} w_k \, p_j(k)$$

step 2. Choose $k \in K$ such that

$$\max_{y_k \leqq j \leqq y_k+h_k-1} U_j \quad \text{subject to } b_{1k} \leqq y_k \leqq b_{2k}-h_k+1$$

is minimum.

step 3. For $j=b_{1k}, b_{1k}+1,\ldots,$ and b_{2k} set

$$U_j \leftarrow U_j - w_k \, p_j(k) \quad \text{if } b_{1k} \leqq j \leqq y_k-1 \text{ or } y_k+h_k \leqq j \leqq b_{2k}$$

$$\leftarrow U_j - w_k \, p_j(k) + w_k \quad \text{if } y_k \leqq j \leqq y_k+h_k-1$$

step 4. Set $K \leftarrow K-\{k\}$. If $K=\emptyset$, stop. Otherwise go to step 2.

Algorithm 2 for P2.

step 1. For j=1, 2, ..., and H set $Q_j \leftarrow 0$. Set $k \leftarrow 1$.

step 2. Choose y_k so that

$$\max_{y_k \leqq j \leqq y_k+h_k-1} Q_j \quad \text{subject to } b_{1k} \leqq y_k \leqq b_{2k}-h_k+1$$

is minimum. Fix y_k.

step 3. For $j=y_k, y_k+1,\ldots,$ and y_k+h_k-1 set

$$Q_j \leftarrow \max_{y_k \leqq i \leqq y_k+h_k-1} Q_i + w_k \, .$$

step 4. If k=N, stop. Otherwise set $k \leftarrow k+1$ and go to step 2.

Algorithm 2E for P2.

step 1. For j=1, 2, ..., and H set

(1) $Q_j \leftarrow 0$.

(2) $$P_j \leftarrow \sum_{k=1}^{N} w_k\, p_j(k)$$

(3) $U_j \leftarrow P_j + Q_j$.

Set $k \leftarrow 1$

step 2. Choose y_k so that

$$\max_{y_k \leq j \leq y_k+h_k-1} U_j \quad \text{subject to } b_{1k} \leq y_k \leq b_{1k}-h_k+1$$

is minimum. Fix y_k.

step 3. For $j=b_{1k}$, $b_{1k}+1$, ..., and b_{2k} set

(1) $$Q_j \leftarrow \begin{cases} \max_{y_k \leq i \leq y_k+h_k-1} Q_i + w_k & \text{if } y_k \leq j \leq y_k+h_k-1 \\ \leftarrow Q_j & \text{otherwise} \end{cases}$$

(2) $P_j \leftarrow P_j - w_k p_j(k)$,

(3) $U_j \leftarrow P_j + Q_j$.

step 4. If k=N, stop. Otherwise set $k \leftarrow k+1$, and go to step 2.

If we apply Algorithm 1 to the example shown in Fig. 2, we obtain U_j as shown in Table 1 below, where I is the number of iterations of step 3. The modules chosen at step 2's and their vertical positions are given in the last two rows of Table 1. We get $w_k p_j(k)$ as given in Table 2 for the example. Applying Algorithm 2E to the example, we get U_j as shown in Table 3. The layout of modules after compaction is shown in Fig. 6.

Table 1

K={1, 2, 3, 4, 5, 11, 12}

	I=	0	1	2	3	4	5	6	7
	j=1	5	5	5	5	5	5	5	5
	2	6	4	4	4	4	4	4	4
	3	9	9	8	8	8	5	5	5
	4	12	12	11	9	9	7	4	4
U_j	5	12	12	11	9	9	7	7	7
	6	10	10	10	8	8	6	6	4
	7	10	10	10	8	8	8	5	3
	8	8	8	8	6	6	6	6	6
	9	9	9	9	7	4	4	4	4
	k=		1	5	12	3	11	4	2
	y_k=		1	1	2	8	7	5	4

Table 2

	k=	1	2	3	4	5	11	12
	j=1	0.50				0.25		
	2	0.50				0.50		0.29
	3		0.67		0.75	0.75		0.57
	4		1.33		1.50	0.50	0.50	0.57
$w_k\,p_j(k)$	5		2.00		2.25	0.25	1.00	0.57
	6		1.33		1.50		1.50	0.57
	7		0.67		0.75		1.50	0.57
	8			1.50			1.00	0.57
	9			1.50			0.50	0.29

Table 3

	k=	1	2	3	4	5	6-10	11	12	
	j=1	3.25	2.25	2.25	2.25	2.25	5.00	5.00	5.00	5.00
	2	2.79	3.79	3.79	3.79	3.79	4.29	4.29	4.29	6.00
	3	3.74	3.74	3.07	3.07	5.32	4.57	4.57	4.57	6.00
	4	6.40	6.40	5.07	5.07	6.57	6.07	6.07	5.57	5.00
U_j	5	8.07	8.07	8.07	8.07	5.82	5.57	6.57	5.57	5.00
	6	5.90	5.90	6.57	6.57	5.07	5.07	6.07	4.57	4.00
	7	4.49	4.49	5.82	5.82	5.07	5.07	6.07	6.57	6.00
	8	4.07	4.07	4.07	5.57	5.57	5.57	5.57	6.57	6.00
	9	4.29	4.29	4.29	2.79	2.79	2.79	2.79	6.29	6.00
	y_k=	2	5	8	3	1		7	2	

Algorithm 2E was computer programmed. An example of computer outputs is shown in Fig. 7.

4. Concluding Remarks

Mathematical modeling or problem formulation is the first step to be taken to accomplish automated circuit layout. This is an important but very difficult step, since there are many engineering restrictions to be observed in layouting a circuit. Especially, both the sizes of modules and the connection among them need be considered, that is, graph theoretical or combinatorial handling only is not enough. Here layout compaction is approximated by packing problems with regions for placing modules being specified. Only compaction in horizontal direction is described. Compaction in vertical direction which is to follow the compaction in horizontal direction needs another formulation.

Computer outputs of Algorithm 2E indicate usefulness of the algorithm, The crucial point with this algorithm is the order in which the positions of modules are determined. In our case this order can be obtained in a natural way from rectangle dissection. More waste spaces tend to appear as more positions of modules are fixed. Modification of the algorithm such as dividing the original rectangle into two or four subrectangles and then applying the algorithm to the subrectangles would result in better layouts.

References

(1) T. Ohtsuki, N. Sugiyama and H. Kawakita, "An optimization technique for integrated circuit design," Proc. Kyoto International Conference on Circuit and System Theory, pp. 67-68, 1970.

(2) S. Nishikawa, K. Okada, K. Noshita, K Naemura and K. Hashimoto, "Algorithms of Musashino experimental design automation system for ic mask patterns MEDIC," Trans. Inst. Elect. and Commun. Eng. of Japan, vol. 55-A, No. 6, pp. 289-296, 1972.

(3) M. K.Hsueh, "Symbolic layout compaction," in *Computer Design Aids for VLSI Circuits*, P. Antognetti, D. O. Pederson and H. Deman Edit. pp. 499-541, Sijthoff and Noordhoff, Alphen aan den Rijn, 1981.

(4) B. S. Baker, E. G. Coffman and R. L. Rivest, "Orthognal packings in two dimensions," SIAM J. Computing, vol. 9, pp.846-855, 1980.

(5) P. J. Federico, "Squaring Rectangles and Squares," in *Graph Theory and Related Topics*, J. A. Bondy and U.S. R. Murty Edit. pp. 173-196, Academic Press, N. Y., 1979.

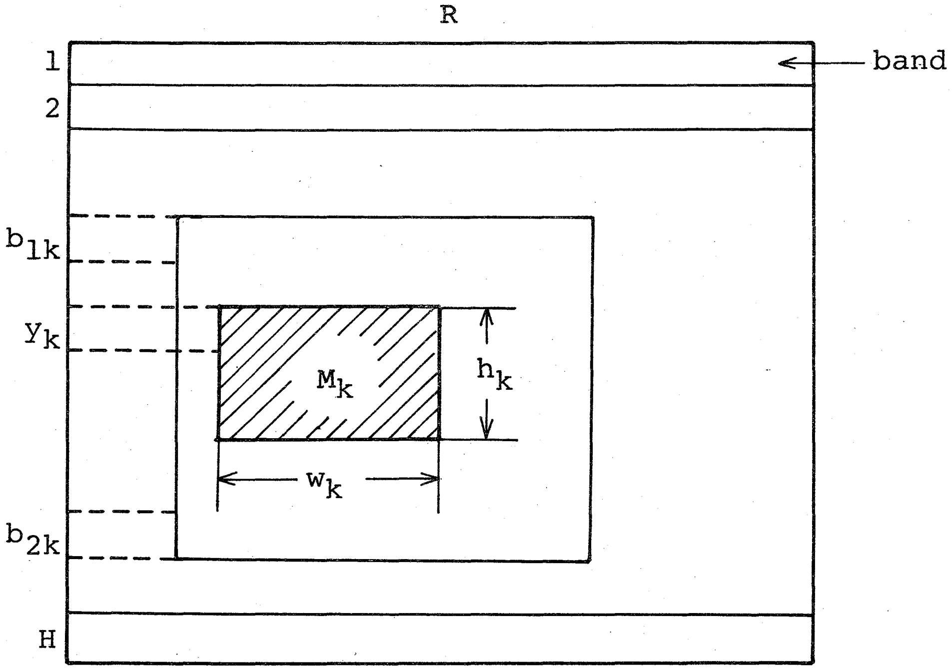


Fig. 1 Quantization of R and M_k

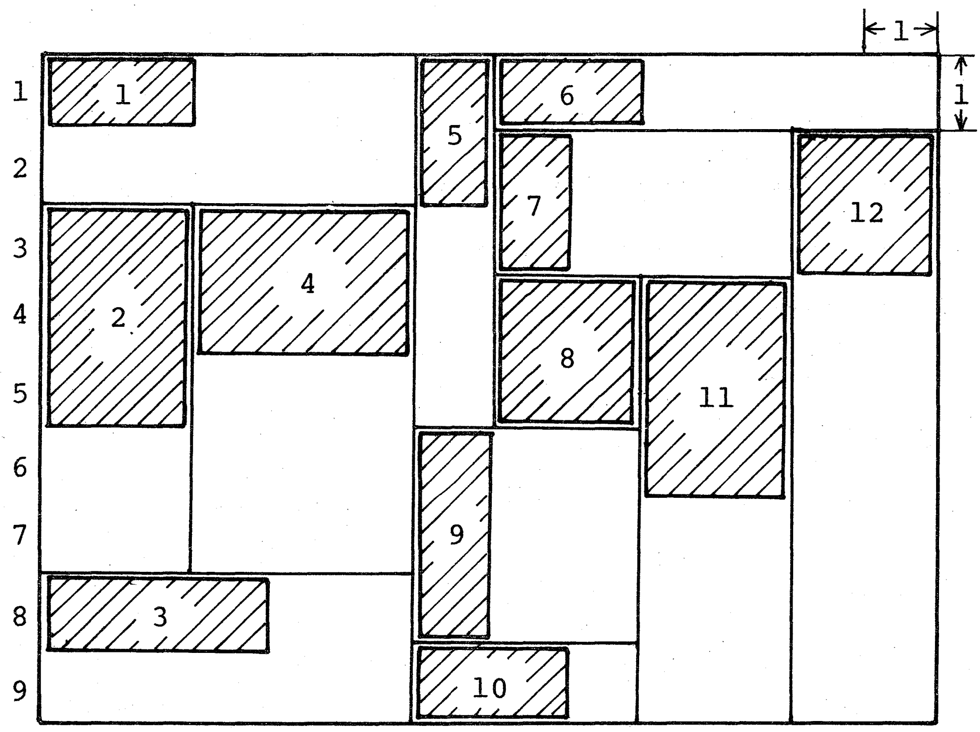


Fig. 2 Example of dissection

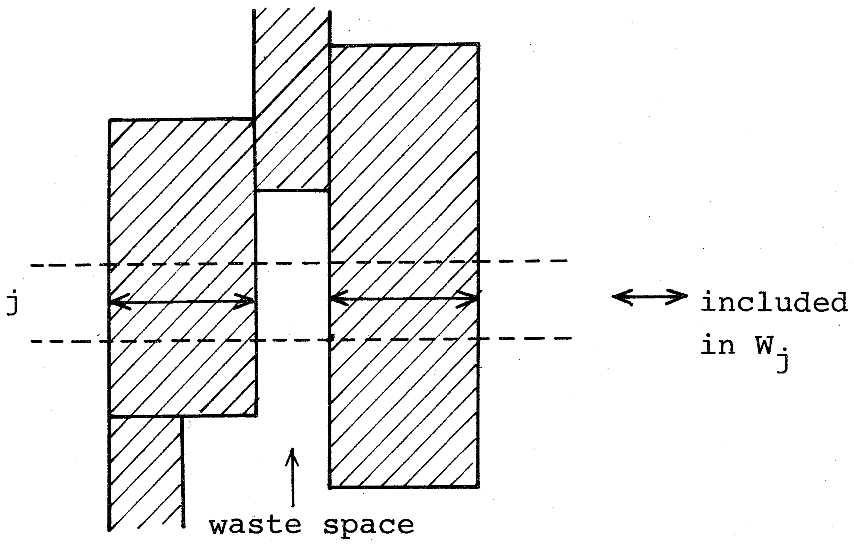


Fig. 3 Waste space

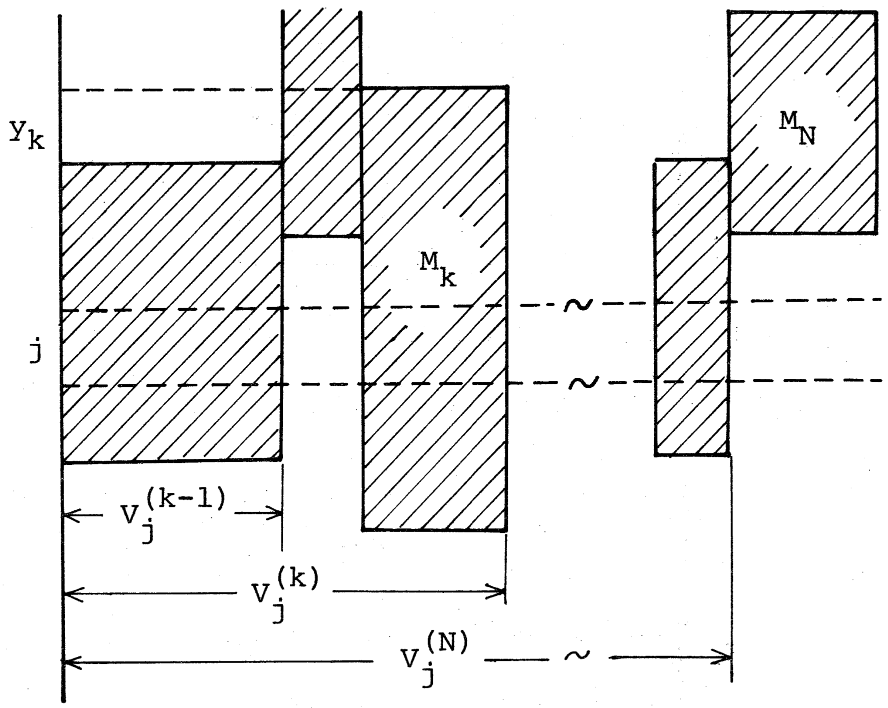


Fig. 4 $V_j^{(k)}$

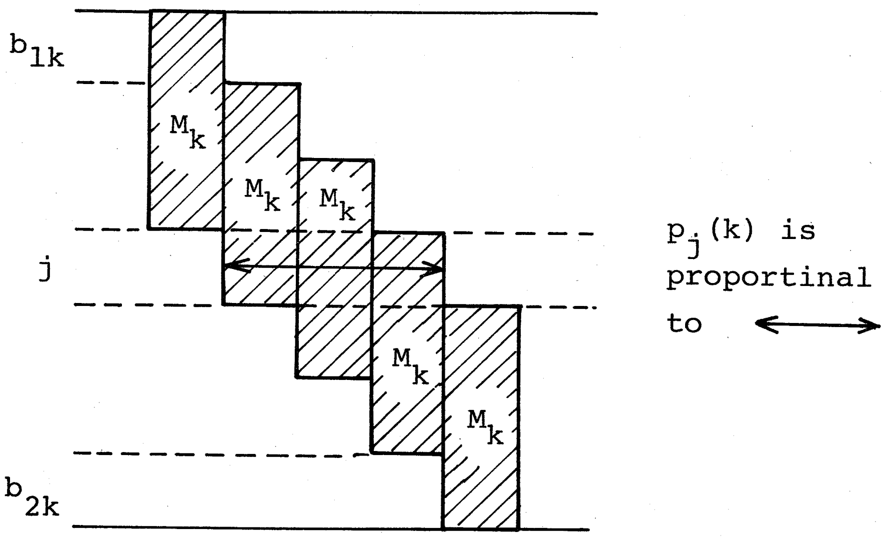


Fig. 5 $p_j(k)$

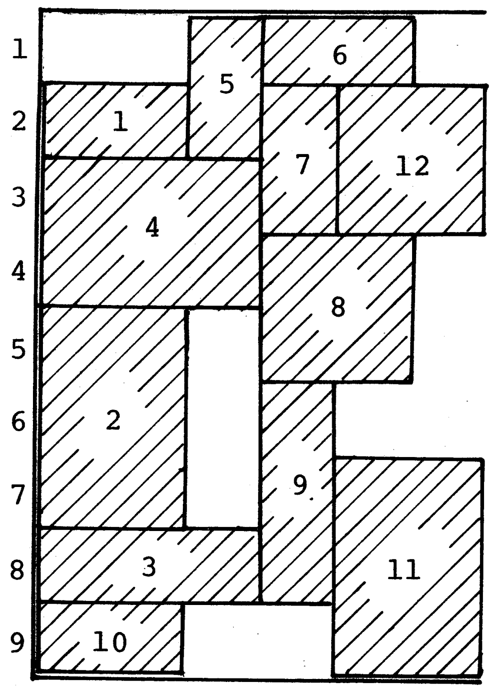
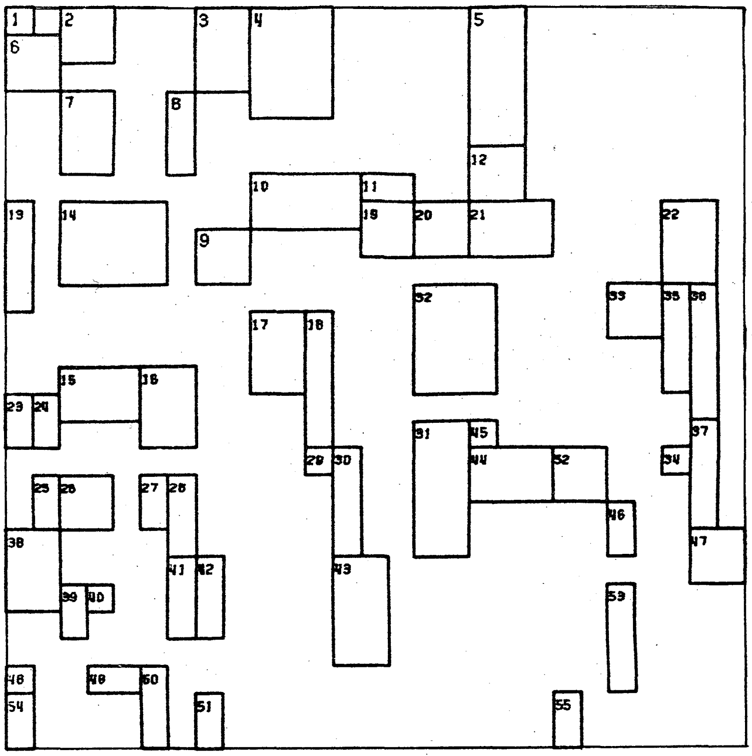
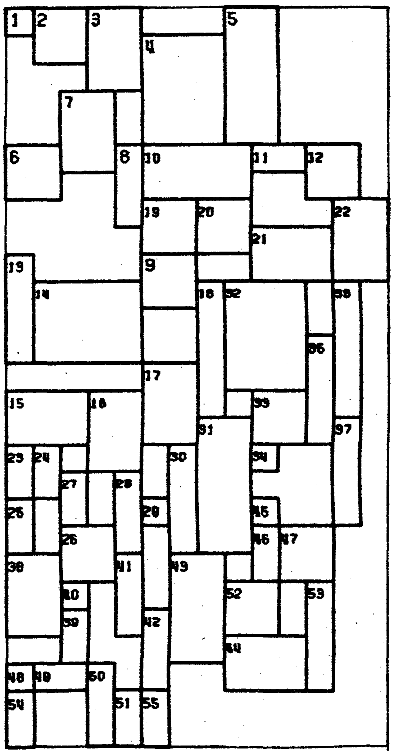


Fig. 6 Layout after compaction



BEFORE COMPACTION



AFTER COMPACTION
LENGTH**1 AND WIDTH**2

Fig. 7 Layout before and after compaction.