Journal of the Korean Institute of Industrial Engineers
Vol. 27, No. 1, pp. 54-60, March 2001.

고정된 형태와 크기가 다른 설비의 배치를 위한 혼합 유전자 알고리듬

이문환[1†] · 이영해[1] · 정주기[2]

[1]한양대학교 산업공학과 / [2]근로복지공단

Hybrid Genetic Algorithm for Facility Layout Problems with Unequal Area and Fixed Shapes

Moon Hwan Lee[1] · Young Hae Lee[1] · Joo Gi Jeong[2]

[1]Dept. of Industrial Engineering, Hanyang University, Seoul
[2]Korea Labor Welfare Corporation, Seoul

In this paper, a shape-based block layout (SBL) approach is presented to solve the facility layout problem with unequal-area and fixed shapes. The SBL approach employs hybrid genetic algorithm (Hybrid-GA) to find a good solution and the concept of bay structure is used. In the typical facility layout problem with unequal area and fixed shapes, the given geometric constraints of unequal-area and fixed shapes are mostly approximated to original shape by aspect ratio. Thus, the layout results require extensive manual revision to create practical layouts and it produces irregular building shapes and too much unusable spaces. Experimental results show that a SBL model is able to produce better solution and to create more practical layouts than those of existing approaches.

Keywords : shape-based block layout (SBL), hybrid genetic algorithm, geometric constraints

1. Introduction

The problem of deciding proper positions of unequal-area and fixed shape facilities within a given building area has prompted a considerable amount of research in recent years. Most of the previous research focused on the relative position of unequal-area facilities within a given total area and rectangular shapes. In general, rigid geometric constraints such as fixed-size and shape are ignored because a huge amount of computational time is required to find an optimal solution even for a small problem. As a result, the previous approaches can not provide usable layouts without manual revision from a practical viewpoint (Kusiak and Heragu, 1987). Even for the considered geometric constraint problems, they produce irregular building shapes and too much unusable space within the building site (Tam, 1992a). In the slicing tree method by Tam (1992a, b) and the bay structure by Tate and Smith (1995), they also have some weaknesses of considering only the area and the shape factor by aspect ratio regardless of the fixed size of facilities.

Kusiak and Heragu (1987) present an excellent survey about the different formulations of problems and algorithms to solve layout problems. Since then, the applications of combinatorial optimization heuristics such as simulated annealing(SA), tabu searches (TS) and genetic algorithms (GA) to layout models have received increasing attention in recent years (Tam, 1992b ; Tate and Smith, 1995; Norman and Smith, 1997 ; Tam and Li, 1991). However, these algorithms

† Corresponding author : Moon Hwan Lee, Dept. of Industrial Engineering, Hanyang University, Ansan, Kyunggi-Do, 425-791 Korea,
Fax : 82-31-409-2423, e-mail : mhlee445@netsgo.com
Received June 1999, accepted January 2001 after 2 revisions.

have some limitations. That is, a GA requires a lot of computational effort. On the other hand, TS and SA can not guarantee the optimality of solutions, while they are excellent for rapidly finding local optimal solutions (Gen and Cheng, 1997, 2000). Therefore, in order to find good solutions, various methods of hybridization that incorporate the merits of local convergence methods or the strength of SA and TS algorithms into GA have started to appear in optimizing design problems recently (Davis, 1987; Fujita *et al*, 1993; Malek and Guruswamy *et al*, 1990). Consequently, efficient layout methods that can reduce unusable space and produce a more practical layout without a lot of manual revision is required, while it satisfies geometric constraints.

Due to these reasons, a SBL approach is proposed to cope with the weak point of manual revision and the rigid geometric constraints of individual facilities including building area. In order to find a good solution, also, Hybrid-GA, which adds the strength of SA and TS algorithms to GA, is presented in this paper. The objective of this paper is to demonstrate the effectiveness of proposed layout approach and Hybrid-GA in solving layout problems with the unequal-area and the fixed shapes of each facility. The experimental studies show that the proposed algorithm can be applied effectively in combinatorial optimization problems found in other areas.

Section 2 defines a layout problem and representation scheme of proposed model is presented. In section 3, we describe the structure and algorithms of Hybrid-GA for the layout problem. The next section shows the experimental results for the proposed model in this paper followed by concluding remarks in section 5.

2. Definition and Representation

2.1 Definition of facility layout problem

The facility layout problem (FLP) considered in this paper can be defined as the problem of arranging n unequal-area facilities of sizes $w_i \times l_i$, where w_i and l_i are the given width and the length of each facility, respectively, $i = 1, \cdots, n$, within a given total space which can be bounded to the length or width of site area in a way to minimize the total material handling cost and slack area cost (refer below equation). The material handling cost between facility i and j is given by $c_{ij} f_{ij} d_{ij}$ where c_{ij} is the cost of one unit of flow between i and j, f_{ij} is the amount of flow between i and j, and d_{ij} is the rectilinear distance between i and j. The cost of slack area is given by $c_s S_A$ where c_s is the cost per unit square meter, S_A is defined as the difference between total site area and total area occupied by the facilities.

$$\text{Min } F = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} f_{ij} d_{ij} + c_s S_A, \quad i, j = 1, 2, \cdots, n$$

In the above equation, the first term represents material handling cost and the second term represents the cost of slack area as defined above. In the previous works, the second term is not normally considered or is represented in any other factors such as a penalty function(Tam, 1992b). However, in actual, the cost of slack area considerably affects the total cost where the land cost is very expensive.

2.2 Representation scheme of SBL

To arrange the unequal-area and fixed shapes of facilities, SBL approach based on bay structure is proposed. The representation scheme is as follow: the given total rectangular space (given building area) is divided into several bays of varying width according to the largest width among the facilities. Each facility with unequal-area and fixed-shape is then placed in the bay one by one, from left to right, according to the solution string of Hybrid- GA as shown in <Figure 1>. That is, the bay width is determined by the largest size among facilities within a bay. When the solution string is given by Hybrid-GA, the algorithm of detailed layout procedure is as follow:

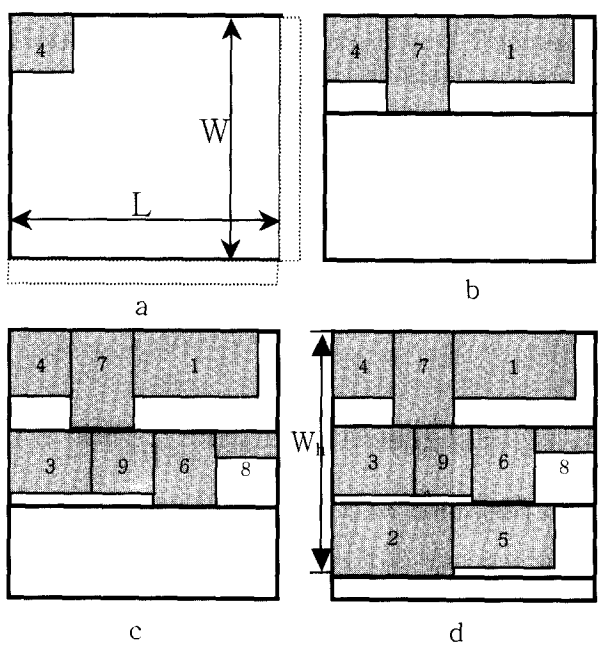


Figure 1. Representation scheme of SBL approach.

Algorithm 1: Layout procedure of SBL

Step 1. Put the facilities from the top of left corner to the direction of right according to the solution string until the sum of the length of each facility, $\sum l_i$, is less then or equal to the given building length, L ($\sum l_i \leq L$, $i = 1, \cdots, k$). If, $\sum l_i > L$, $i = 1, \cdots, k, k+1$, then create the first bay and Go to step 2 (refer <Figure 1(a) and (b)>).

Step 2. Determine the size of bay width based on the facilities positioned already. Bay width equals to that size of the largest width among k facilities. That is, the size of p^{th} bay B_p, $B_p = \max\{w_i\}$. Go to step 3 (refer <Figure 1(b)>).

Step 3. Continue the step 1 and 2 until n facilities are all positioned(refer <Figure 1(a)~(d)>).

In step 3, it is assumed that $\sum B_p \leq W$, $p=1, \cdots, q$. Because if the sum of B_p is greater than the building width, $\sum B_p > W$, then the solution string becomes infeasible.

<Figure 1> shows the layout procedure of SBL when the solution string for 9 facilities is ***471396825***. As shown in <Figure 1>, the bays are created after that the facilities of ***1***, ***8*** and ***5*** are placed, and the B_1, B_2 and B_3 is determined by the w_7, w_6 and w_2, respectively. The solution string, as given above, consists of a series of integers representing facilities' number. The order of integers represents the sequence that the facilities are placed within bays. As described in section 2.1, the slack area (S_A: unusable space) and the space utilization (S_U) is calculated by following equation (1) and (2), respectively. As shown in <Figure 1(d)> above example, the unusable space, S_A, is calculated by substituting W_h for W in S_A.

$$S_A = W \times L - \sum_{i=1}^{n} w_i \times l_i \quad (1)$$

$$S_U = \frac{|W \times L - S_A|}{W \times L} \quad (2)$$

3. Hybrid genetic algorithm (Hybrid-GA)

3.1 Structure of Hybrid-GA

The structure of Hybrid-GA can be classified into two types. First type: searches local solutions by employing a GA and find a final solution by TS and SA. Second type: reversed procedure of the first type, search local solutions by TS and SA and employ a genetic algorithm to find an optimal solution (Malek and Guruswamy *et al.*, 1990) (see <Figure 2>). The Hybrid-GA is to make up for a weakness in the genetic algorithm by employing the strong points of TS and SA that find local solutions rapidly. That is, the genetic algorithm is powerful to find global optimal solutions, however it takes a lot of computational efforts.

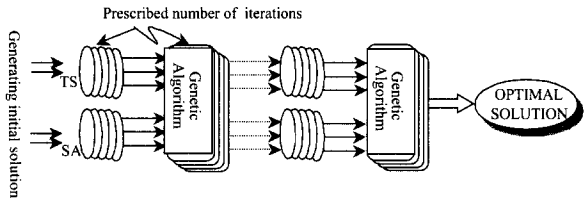


Figure 2. Structure of Hybrid-GA.

On the other hand, TS and SA are excellent for finding local optimal solution much more quickly than that of a GA. Consequently, the objective of Hybrid-GA is to find a better solution than that of when they are used alone with reasonable computational efforts. The problem of exchanging timing between local search and GA is difficult in Hybrid-GA. In this paper, however, not the method to add the genetic operation after waiting until the local search is converged completely, but the method to add the genetic operation in the middle of the process of local search is used. Since the FLP is NP-complete and some application areas of it require a good solution, a heuristic algorithm is developed (<Figure 3>).

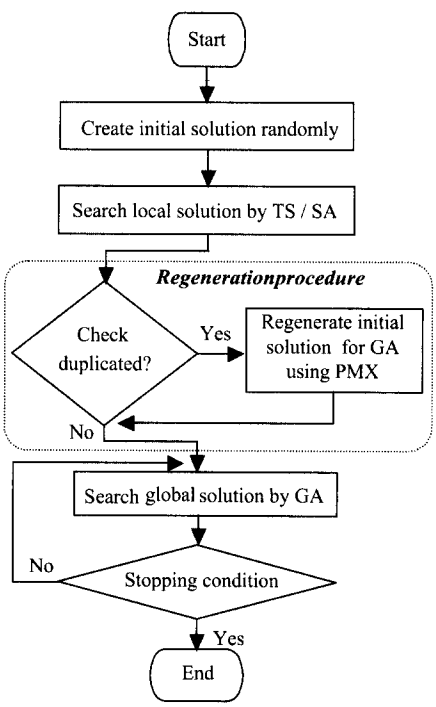


Figure 3. Solution procedure of Hybrid-GA.

The algorithm is composed of two phases. In phase I of the algorithm, a kind of local solution sets for the initial solution of GA are searched by TS/SA and the regeneration procedure is processed in order to guarantee the randomness and control the search space of GA. In phase II, then, GA searches optimal solution. The procedures of Hybrid-GA proposed here are as follow.

Step 1. Randomly generate initial solutions according to the population size (N).

Step 2. Find local solutions over half-size of population by employing TS and SA algorithms, respectively.

Step 3. Check that "are there duplicated strings among local solutions?" in order to guarantee the randomness of initial solutions and control the search space of GA.

Step 4. Search global solution by GA until the stop condition is reached.

<Figure 3> shows the solution procedure of Hybrid-GA. In this research a second form of the structure of Hybrid-GA described earlier is used. That is, a local searches by TS and SA and a global search by GA finally. However, the search space is required to be controlled properly so that the optimal solution does not fall into a local optimum. In practice a number of solutions, which have same value, exist within population after a local search by TS and SA. Thus, the duplicated solution strings should be checked and regenerated before a global search is processed (<Figure 4>). The regeneration procedure and its algorithm are as follows.

Step 1. Check the number of duplicated solution strings, *Dn-t* and *Dn-s*, for each TS and SA, respectively. Go to step 2.

Step 2. Calculate adaptive crossover rate, A_c. The adaptive crossover rate, A_c, is calculated by the ratio of *Dn-t* (*Dn-s*) / *Sp-t*(*Sp-s*). *Sp-t* and *Sp-s* indicate the sub-population (sub-pop) size of TS and SA, respectively. Go to step 3.

Step 3. Generate the number of duplicated solution strings randomly until the total number of solution strings equal to N and A_c equal to zero. The new strings are generated by crossover operation between the best string and the worst value string within individual algorithm, TS and SA, respectively. Check the sub-pop size $n = N/2$ and go to step 4.

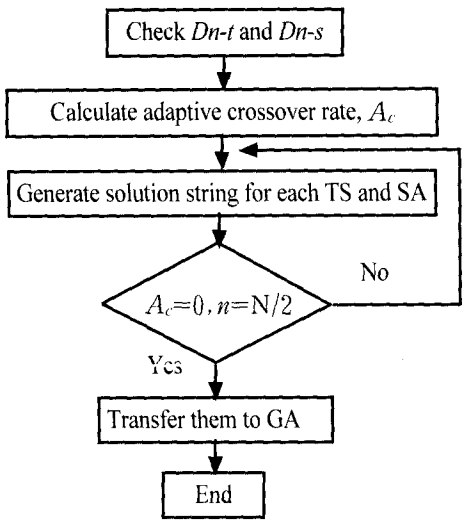


Figure 4. Regeneration procedure for duplicated sol.

Step 4. Transfer the new solution strings to the genetic algorithm.

The number of duplicated solution strings and adaptive crossover rate may be different in each time when the algorithm is repeated. Thus, the regeneration procedure is continued until stop conditions, A_c equal to zero and n equal to $N/2$, are satisfied as shown in <Figure 4>.

4. Numerical Example and Experimental Results

In this section, we represent numerical examples based on the proposed layout model in the previous section. We concentrate on the usefulness of shape-based block layout approach and Hybrid-GA from the practical viewpoint such as manual revision, solution quality and the checking of space utilization for design alternatives, etc. We consider two different layout approaches, Type1 and Type2. As described in the previous section, Type1 considers original shapes with unequal-areas, while Type2 considers unequal- areas with aspect ratio as a shape parameter.

The setting for numerical examples can be summarized as follows. The population size (N) is considered from 100 to 400 in order to investigate the effect of solution quality and converging speed with respect to given pop-size. Crossover rate, $p_c = 0.25$, mutation rate, $p_m = 0.1$, and max generation number for stop condition, $n = 5000$ in the GA. Genetic operators are PMX, CX and OX for crossover operation and inversion, insertion and swap for

Table 1. Results for enumeration and Hybrid-GA

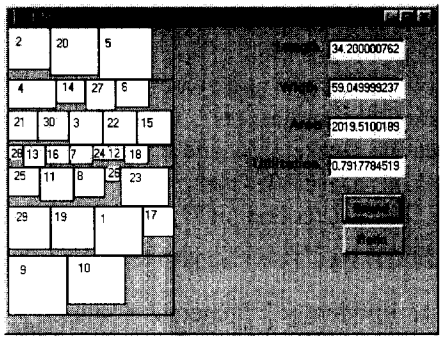
Facility No.	Optimal Sol.	Hybrid-GA
5	512	512
6	896	896
7	1319	1319
8	1940	1940
9	2673	2673
10	3779	3779

mutation operation. Selection, ranking and tournament are used for reproduction rules. In local search the iteration number of TS and SA is set 100, respectively. The test data sets are based on Tam (1992b), Nugent *et al.* (1968) and Woo and Park (1997)(refer to Appendix). The proposed algorithm was implemented in C++ on a PENTIUM II 400 processor with 256-MB of main memory.

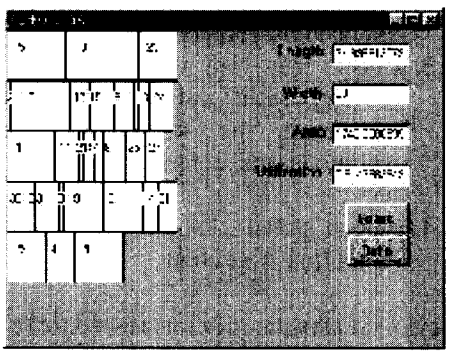
As for the verification of the Hybrid-GA model for finding optimal solutions, the results were compared with those obtained by enumeration method over 5~10 facilities. As shown in <Table 1>, the Hybrid-GA model exactly found the optimal solutions obtained by the enumeration method.

As for the test problem, the optimal solutions obtained by proposed layout model are far better than those of previous works as shown in <Table 2>. For the performance of SBL method, percentage reduction in best value(costs) of 7.1~39.2% and 7.9~40.4% was achieved in comparison with existing algorithm using GA alone and aspect ratio approach, respectively.

Please note, however, that since the solutions obtained by Tam (1992b) are based on an objective function that includes the facility shape- based penalties, the best solution may not be the optimal layout when the penalties are not considered. Thus, the comparison with Tam seems to be somewhat meaningless. The average and best solutions are based on 5000 cycles of Hybrid-GA for 9 cases (3×3; combination of operators), while previous works are based on 5000 generations for GA alone. The slack area costs in the objective function are not included for the purpose of comparison.



(a) Type 1(Best solution: 46275)



(b) Type 2 (Best solution: 51373)

Figure 5. Final layout for the two type's model.

The final layout of the best solutions in cases of Type1 (fixed size) and Type 2 (aspect ratio) when n=30 are shown in <Figure 5>. As founded in earlier works, we can see that the Type2 layout method using aspect ratio requires extensive manual revision to create practical layouts. Moreover, the solution quality is worse than that of Type1 with unequal-area and fixed shapes. The Type1, however, has a disadvantage

Table 2. The performance of proposed model for each 15~30 facilities layout. (Best / Average)

n	Layout type	Previous works		Lee *et al.*	Performance of proposed model(Type 1)	
		Woo & Park	Tam		Cost reduction	Type 1 vs. Type 2
15	Type 1 Type 2	9120/ 9855/	13762/ 12240/	5537/5694 9287/9524	39.2% ↓	40.4% ↑
20	Type 1 Type 2	21885/ 22656/	26921/ 28646/	20329/20769 22067/22391	7.1% ↓	7.9% ↑
30	Type 1 Type 2	50492/ 52869/	55668/ 58824/	46275/48424 51373/53435	8.4% ↓	10.0% ↑

† Type 1: SBL; Type 2: Aspect ratio. Previous works → GA alone, Lee *et al.*→ Hybrid-GA.
† Example of performance calculation: 0.392 = | 9120-5537 | / 9120, 0.404 = | 9287-5537 | / 9287.

Table 3. Average and standard deviation (SD) of CPU Time for each algorithm

Algorithm	$n=15$	$n=20$	$n=30$
	Average / SD	Average / SD	Average / SD
TS	24.8 / 7.6	42.4 / 15.0	47.5 / 13.8
SA(α=0.9)	86.0 / 37.2	156.0 / 42.8	47.7 / 15.1
GA	28.2 / 9.4	40.6 / 16.2	67.7 / 27.4
Hybrid-GA	34.1 / 11.9	50.7 / 20.1	83.9 / 35.3

† Note: The results based on Pop-size N= 100~400 and 10 iterations for each pop-size

that it has lower space utilization than that of Type 2 as shown in <Figure 5>, 77.7% and 91.4%, respectively. In <Figure 5>, length, width and area indicate building area conditions, which can be changed by layout designer according to real- world situations.

For the computational efforts, the Hybrid-GA was not better than those of each algorithm alone, GA, TS and SA, respectively. As shown in <Table 3>, it takes a little more seconds in average than the other algorithms. We also found that the optimal solution could be improved according to the pop-size is increase, while the number of converging generation is decreased (see <Table 4>). In the experiments of proposed model, actually, the most of optimal solutions were found within 650 generations in average. Thus, if we consider 1000 generations at least as for the stopping conditions in the model, the computational efforts will be less took than the above results.

5. Concluding Remarks

We have presented a shape-based block layout approach to facility layout problems considering the unequal-area and the fixed shape of individual facilities. We found that the approach considered in this research could produce more practical layout alternative and provide more flexible designs as well as providing better optimal solution than those of existing layout approach. As alluded before, from a designer's viewpoint, the other design factors which are not included in numerical example can be practically considered at the same time such as aisle width, the direction of facilities, land cost and building area condition, etc. However, our research has some limitations. First of all, the proposed model in this paper does not deal with irregular building shape because we tried to avoid too much specific cases. We also assumed that the irregular shape of each facility could be fixed so as to be equivalent to the rectangular shape. But, sometimes it might be very restrictive to impose this assumption. For the structure of Hybrid-GA, a more efficient and general algorithm should be developed henceforth.

These limitations and some weakness could hamper applying the research results to a real- world situation.

Table 4. The trend of best/average solution according to the change of pop-size for $n=15$, 20 and 30

(a) $n=15$ — Best / Average

Pop_size	100	200	300	400	Total
Type 1 layout	5616/5711	5616/5714	5537/5683	5616/5669	5537/5694
Type 2 layout	9416/9582	9325/9432	9414/9566	9287/9519	9287/9524
Best layout	9 11 8 7 12 5 6 10 1 2 13 4 3 14 15				

(b) $n=20$ — Best / Average

Pop_size	100	200	300	400	Total
Type 1 layout	20374/20830	20442/21034	20329/20355	20357/20858	20329/20769
Type 2 layout	22067/22505	22014/22431	22180/22360	22077/22269	22067/22391
Best layout	10 5 1 6 3 8 11 17 13 14 18 12 7 16 19 2 15 4 9 20				

(c) $n=30$ — Best / Average

Pop_size	100	200	300	400	Total
Type 1 layout	47862/49523	47451/48789	47449/48531	46275/46853	46275/48424
Type 2 layout	51373/53584	52293/53280	52166/53889	51502/52990	51373/53435
Best layout	2 20 5 4 14 27 6 21 30 3 22 15 28 13 16 7 24 12 18 25 11 8 26 23 29 19 1 17 9 10				

However, this type of layout method proposed here, which can consider the other design factors as stated above, could be applied usefully for the optimal design of facility layout problems and the evaluation of relevant systems. Finally, notice that the performance of heuristic algorithms is problem dependents.

Appendix : Test data for geometric characteristics of each facility

Block no.	Area	Aspect ratio		Fixed size	
		Upper bound	Lower bound	Width	Length
1	24.0	1.00	0.80	5.6	4.3
2	16.0	1.15	0.75	3.7	4.3
3	36.0	1.85	0.60	7.75	4.65
4	8.0	1.10	0.30	4.0	2.0
5	21.0	1.18	0.90	4.2	5.0
6	17.5	1.00	0.50	5.0	3.5
7	3.6	1.40	0.30	2.0	1.8
8	15.4	1.25	0.60	4.0	3.85
9	20.0	1.0	0.90	4.0	5.0
10	19.5	1.80	1.20	4.0	4.875
11	16.0	1.10	0.85	4.0	4.0
12	9.0	1.0	0.90	3.2	2.8
13	9.0	1.10	0.80	3.0	3.0
14	25.0	1.05	0.92	5.0	5.0
15	4.0	1.15	0.85	2.0	2.0
16	3.0	1.0	0.90	1.7	1.76
17	4.0	1.0	1.0	2.0	2.0
18	9.0	1.0	1.0	3.0	3.0
19	4.5	1.10	0.70	2.5	1.8
20	5.0	1.50	0.50	3.1	1.6
21	16.0	1.10	0.85	4.3	3.72
22	9.0	1.0	0.90	3.2	2.8
23	9.0	1.10	0.80	3.0	3.0
24	25.0	1.05	0.92	5.2	4.8
25	4.0	1.15	0.85	2.0	2.0
26	3.0	1.0	0.90	1.7	1.76
27	4.0	1.0	1.0	2.0	2.0
28	9.0	1.0	1.0	3.0	3.0
29	4.5	1.10	0.70	2.2	2.05
30	5.0	1.50	0.50	1.8	2.78

References

Davis, L. (1987), *Genetic algorithms and simulated annealing*, Morgan Kaufman Publishers, Los Altos.

Fujita, K., Akagi, S. and Hirokawa, K. (1993), Hybrid approach for optimal nesting using a genetic algorithm and a local minimization algorithm, *Advances in Design Automation*, ASME, 477-484.

Gen, M. and Chen, R. (1997), *Genetic algorithms & engineering design*, John Wiley & Sons, Inc., New York.

Gen, M. and Chen, R. (2000), *Genetic algorithms & engineering optimization*, John Wiley & Sons, Inc., New York.

Kusiak, A. and Heragu, S. S. (1987), The facility layout problem, *European Journal of Operational Research*, **29**, 229-251.

Malek, M., *et al.* (1990), A hybrid algorithm technique, Technical Report, Texas Univ. at Austin, TR-89-06.

Nugent, C. E., Vollman, T. E. and Ruml, J. (1968), An experimental comparison of techniques for the assignment of facilities to locations, *Operations Research*, **16**, 150-173.

Tam, K. A. (1992), A simulated annealing algorithm for allocating space to manufacturing cells, *Int. J. Prod. Res.*, **30**(1), 63-87.

Tam, K. A. (1992), Genetic algorithms, function optimization, and facility layout design, *European Journal of Operational Research*, **63**, 322-346.

Tam, K. A. and Li, S. G. (1991), A hierarchical approach to the facility layout problem, *Int. J. Prod. Res.*, **29**(1), 165-184.

Tate, D. M. and Smith, A. E. (1995), Unequal-area facility layout by genetic search, *IIE Transactions*, **27**, 465-472.

Woo, S. S. and Park, Y. B. (1997), Applying a genetic algorithm to a block facility layout, *Korean Management Science Review*, **14**(1), 67-76.