Int J Adv Manuf Technol (2006) 29: 766–771
DOI 10.1007/s00170-005-2576-2

ORIGINAL ARTICLE

R. Christu Paul · P. Asokan · V.I. Prabhakar

A solution to the facility layout problem having passages and inner structure walls using particle swarm optimization

Received: 28 July 2004 / Accepted: 19 January 2005 / Published online: 25 January 2006

Abstract This paper proposes a new approach called particle swarm optimization (PSO) to derive better solutions for unequal-area facility layouts that are to have inner walls and passages. PSO is a population based optimization tool, has fitness values to evaluate the population, update the population and search for the optimum with random techniques. A heuristic method is adopted for establishing the relationship between the facilities and passages. A comparative study is performed with the existing algorithm and it shows a better performance for the proposed algorithm. The objective of this study is to minimize material flow between facilities while at the same time satisfying the constraints of areas, aspect ratios of the facilities, and inner structure walls and passages. The proposed algorithm based on the PSO in this study was implemented with C++ language.

Keywords Facility layout problem · Heuristic method · Inner structure walls · Particle swarm optimization · Passage

1 Introduction

Layout design has a significant impact on the performance of a manufacturing or service industrial system and has been an active research area for many decades. Most literature for a layout design problem falls into two major categories, algorithmic and procedural approaches [1].

Facility layout problems (FLPs) concerning space layout optimization have been investigated in depth by researchers in many fields, such as industrial engineering, management science, and architecture, and various algorithms have been proposed to solve FLPs. However, these algorithms for the FLP cannot consider inner structure walls and passages within the block plan (or available area). They are also limited to a rectangular boundary shape of the block plan. In this study, an improved genetic algorithm (GA) is proposed for solving problems having the inner structure walls and passages within an available area of a curved boundary [2]. Islier [3] used a genetic algorithm to solve the facility layout problems.

Recently some attentions have been focused on a special class of search methods called extended neighborhood search which may be considered as generic heuristics methods [4]. The great advantage of these methods is to avoid being caught in local optima by sometimes accepting moves that worsen the objective function. It addresses some practical aspects which are facilities with different areas, shapes and orientations, any polygonal format for the border, fixed facilities, pick-up and drop off points, in case of machine layout problems. The three most known methods in literature are tabu search (TS), genetic algorithm (GA) and simulated annealing (SA). Heragu and Alfa compared the performance of SA and TS in FLP and they proposed a hybrid simulated annealing using a modified penalty algorithm [5].

A computerized plant layout analysis and evaluation technique (PLANET) also utilizes information about the material flow [6]. This algorithm establishes a layout by posing questions, such as which machine is placed next and the placement location. Three heuristic algorithms are applied to generate alternative layouts, and some manual evaluations/adjustments are also performed. A computer aided layout planning tool is adapted to facilitate the layout alternative generation process as well as to collect quantitative performance data such as flow distance, adjacent score and shape ratio [7].

Most of the published research work for facilities layout design deals with equal-area facilities [8]. By disregarding the actual shapes and sizes of the facilities, the problem is generally formulated as a quadratic assignment problem (QAP) of assigning equal area facilities to discrete locations on a grid with the objective of minimizing a given cost function. Heuristic techniques such as simulated annealing, simulated evolution, and various genetic algorithms developed for this purpose have also been applied for layout optimization of unequal area facilities by first subdividing the area of each facility in a number of “unit cells”.

The particle swarm optimization (PSO) technique has been developed by Eberhart and Kennedy in 1995 [9] and it is a sim-

R.C. Paul (✉) · P. Asokan · V.I. Prabhakar
Department of Production Engineering,
National Institute of Technology,
Tiruchirappalli, India
E-mail: r_christupaul@yahoo.co.in

ple evolutionary algorithm which differs from other evolutionary computation techniques in that it is motivated from the simulation of social behavior. PSO exhibits good performance in finding solutions to static optimization problems [10].

Particle swarm optimization is a swarm intelligence method that roughly models the social behavior of swarms [11]. PSO is characterized by its simplicity and straightforward applicability, and it has proved to be efficient on a plethora of problems in science and engineering. Several studies have been recently performed with PSO on multi objective optimization problems, and new variants of the method, which are more suitable for such problems, have been developed.

PSO has been recognized as an evolutionary computation technique and has features of both genetic algorithms (GA) and evolution strategies (ES) [12]. It is similar to a GA in that the system is initialized with a population of random solutions. However, unlike a GA each population individual is also assigned a randomized velocity, in effect, flying them through the solution hyperspace. As is obvious, it is possible to simultaneously search for an optimum solution in multiple dimensions. This paper makes an attempt to utilize the advantages of the PSO algorithm and the results are compared with the existing genetic algorithm and improved GA.

2 Problem formulation:

The mathematical model adopted by Lee et al. [2] has been used in this research work.

2.1 Input information:

1. Number of facilities to be allocated to the available area = 8.
2. Available area and its boundary shape = 240 20×12 rectangle.
3. Number of inner structure walls = 3.
4. Number and widths of each vertical and horizontal passage. Vertical passage = 2, 2×3 Horizontal passage = 2, 20×1.
5. Upper and lower bounds of the required area for each facility.
6. Upper and lower bounds of the required aspect ratio for each facility.
7. Material flows between facilities.

2.2 Objective function value [OBV]:

The objective is to minimize materials flow between facilities while at the same time satisfying the constraints of areas, aspect ratios of the facilities, and inner structure walls and passages. Finding the best facility layout means determining sequence and areas of the facilities to be allocated, and the location of passages [13].

$$\text{Minimize F} = \sum_{i=1}^{M}\sum_{j=1}^{M} f_{ij} d_{ij}$$

Table 1. Upper and lower bounds of the required area for each facility

Facility		1	2	3	4	5	6	7	8
Required area	Upper bound	18	21	17	23	22	18	16	20
	Lower bound	14	17	13	19	18	14	12	16

Table 2. Upper and lower bounds of the required aspect ratio for each facility

Facility		1	2	3	4	5	6	7	8
Required aspect ratio	Upper bound	1.2	1.5	1.6	1.3	1.8	1.4	1.3	1.8
	Lower bound	0.5	0.4	0.4	0.3	0.3	0.2	0.3	0.6

Subject to

$$g_1 = \alpha_i^{min} - \alpha_i \le 0,$$
$$g_2 = \alpha_i - \alpha_i^{max} \le 0,$$
$$g_3 = a_i^{min} - a_i \le 0,$$
$$g_4 = a_i - a_i^{max} \le 0,$$
$$g_5 = \sum_{i=1}^{M} a_i - A_{available} \le 0,$$
$$g_6 = (x_i^r - x_s^{i.s.w})(x_s^{i.s.w} - x_i^l) \le 0,$$

where, i, $j = 1, \ldots, M$ and $s = 1, \ldots, P$.

- f_{ij} : Material flow between the facility i and j,
- d_{ij} : Distance between centroids of the facility i and j,
- M : Number of the facilities,
- α_i : Aspect ratio of the facility i,
- α_i^{min} and α_i^{max} : Lower and upper bounds of the aspect ratio α_i,
- a_i : Assigned area of the facility i,
- a_i^{min} and a_i^{max}: Lower and upper bounds of the assigned area a_i,
- $A_{available}$: Available area,
- P : Number of the inner structure walls,

Table 3. Material flows between facilities

Facility	1	2	3	4	5	6	7	8
1	0	15	0	0	0	5	5	10
2	15	0	25	40	100	90	80	70
3	0	25	0	0	0	20	30	200
4	0	40	0	0	30	10	0	0
5	0	100	0	30	0	50	70	20
6	5	90	20	10	50	0	5	0
7	5	80	30	0	70	5	0	10
8	10	70	200	0	20	0	10	0

- $x_s^{i.s.w}$: Position (x-coordinate) of the inner structure wall s,
- x_i^l and x_i^r : x-coordinates of the left and right boundaries of the facility i.

3 Proposed algorithm for the facility layout:

3.1 Overview of the proposed algorithm:

The algorithm proposed in this study is based on the particle swarm optimization (PSO). Particle swarm optimization (PSO) is an evolutionary computation technique. It is based on the social behavior of animals such as bird flocking, fish schooling, and swarm theory [14].

3.2 Representation of facility layout:

Figure 2 shows the arrangement of facilities in the available rectangular area with the inner structure walls and passages. The placement procedure of all the facilities follow an x-oscillatory pattern [3, 15]. Most of the earlier facility layout problems had not considered the passages and inner structure walls, but in a realistic condition there will be passages and inner structure walls within the facilities, wherein the material flow between the facilities takes place. The maximum allowable dimensions of each of the facilities within the given available rectangular area and its passages are given in Fig. 3.

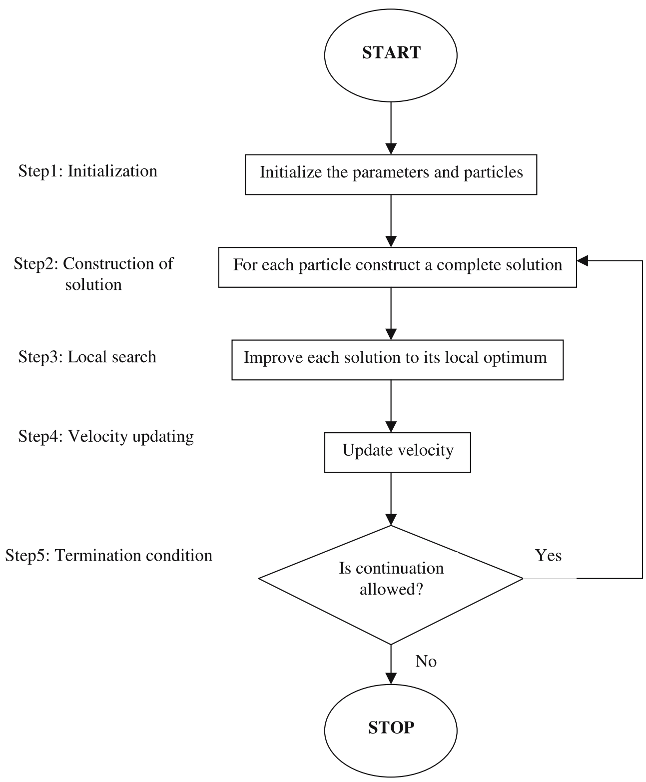


Fig. 1. PSO flow chart

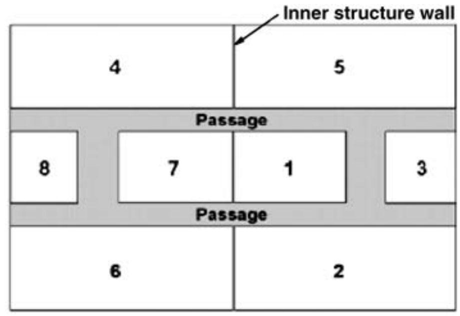


Fig. 2. Layout with inner structure walls and passages

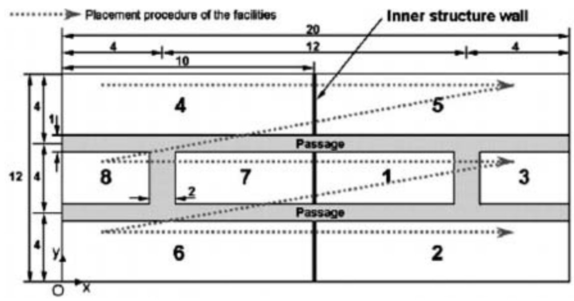


Fig. 3. Layout of facilities with dimensions

(Sequence of facilities)	(Areas of the facilities)	(Aspect ratios of the facilities)
4, 5, 8, 7, 1, 3, 6, 2	35, 35, 9, 15, 15, 9, 35, 35	0.8, 0.6, 1.1, 1.2, 1.4, 0.5, 0.4, 0.7

Fig. 4. Representation of parameters in PSO

3.3 Distance calculation method between the facilities:

Another importance issue is how to obtain the correct travel distance. Two popular calculation methods are the Euclidean distance method and the rectilinear distance method [16]. For both, the distance from a point 'i' to another point 'j' is equal to the distance from 'j' to 'i'. However in reality, the travel distance from 'i' to 'j' is not necessarily equal to the travel distance from 'j' to 'i'. In the existing facility layout algorithms, a rectilinear distance method has been used to calculate the distance between the facilities d_{ij}. In this method, the distance d_{ij} is calculated by summing the vertical and horizontal distances between the facilities, as is shown below.

$$d_{ij} = |x_i - x_j| + |y_i - y_j|$$

However, this rectilinear distance method cannot be employed for our problem having passages and inner structure walls, hence a new method is employed for the distance calculation between the facilities via the passages. In this method, some nodes are established for every facilities as shown in Fig. 5, i.e., $N_1, N_2, N_3 \ldots N_{12}$.

For every particle in the PSO algorithm we change the area and aspect ratio of each facility within their upper and lower bound. When the width of the facilities is held constant, then it is the length of the facilities which vary when the area varies.

The nodes $N_1, N_2, N_3 \ldots N_{12}$ are established with respect to the length of each facilities, i.e., taking the left side as the origin. For example from Fig. 5.

The area of facility 4 which is allocated in the 'first' Position, i.e., L1 = 35

We know, the width of this facility = 3.5 (from Fig. 3).

Therefore, the length of this facility $l_1 = 35/3.5 = 10$.

Now, $N_1 = (l_1/2) = 5$.

Similarly, $N_4 = l_3 + 2 + (l_4/2)$ and $N_6 = l_3 + 2 + l_4 + l_5 + 2 + (l_6/2)$.

This is how the nodes are established with respect to the length of the facilities. Now, the distance between two facilities will be the distance between its corresponding nodes along with their centroid to its node distance.

Hence, distance between facilities 1 and 8 will be $(N_{10} - N_1) + 4 + (N_{11} - N_8) + c_1 + c_8$. Where c_1 and c_8 are the distance between the centroid of facility 1 and 8 respectively to its nodes N_1 and N_8.

3.4 Particle swarm optimization

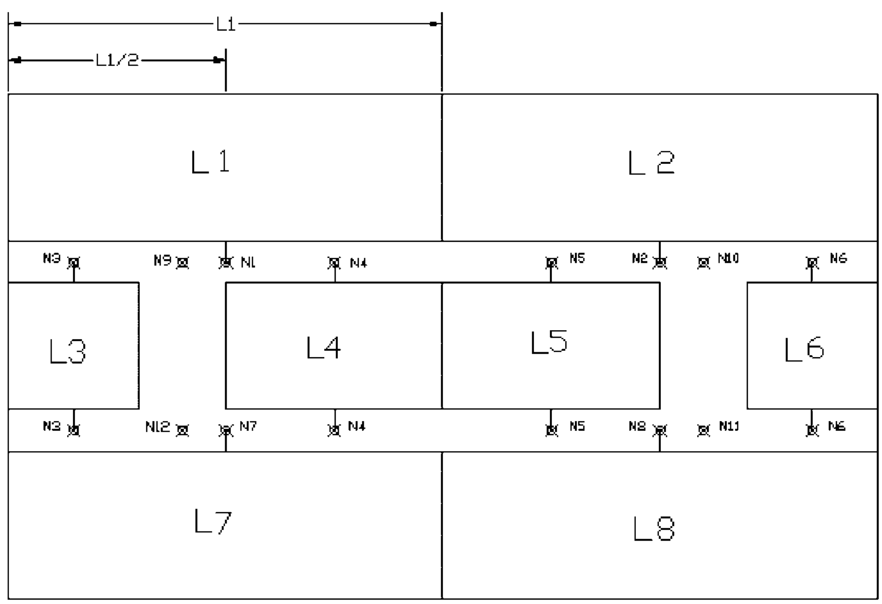
Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [9]. The underlying motivation for the development of PSO algorithm was social behavior of animals such as bird flocking, fish schooling, and swarm theory. Similar to genetic algorithms (GA), PSO is a population based optimization tool, both have fitness values to evaluate the population, both update the population and search for the optimum with random techniques, both systems do not guarantee success. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, particles update themselves with internal velocity. They also have memory, which is important to the algorithm. Also, the potential solutions, called particles, are "flown" through the problem space by following the current optimum particles. Compared to GA, the information sharing mechanism in PSO is significantly different. In GAs chromosomes share information with each other. So the whole population moves like a group toward an optimal area. In PSO, only Gbest gives out the information to others. It is a one-way information sharing mechanism. The evolution only looks for the best solution. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases. The advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas, such as function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied [14].

3.4.1 Algorithm

PSO algorithm:

Step1: Generate initial solution randomly for all particles.
Step2: Assign Pbest[i] = initial solution where i = 1,2, ... N (N: no of particles).
Step3: Find best among all particles and assign this to Gbest.
Step4: Generate initial velocities randomly for all particles.
Step5: Add velocities to the corresponding particles, i.e., Present[i] (new) = Present[i] (old) + V[i].
Step6: Update velocity according to V[i] = V[i] (present) + C1 * (Pbest[i] – present[i]) + c2 * (Gbest[i] – present [i]).
Step7: if number of iterations < cyc Goto step5.
Step8: Ubest is the best among all Gbest If number of iterations < cond Goto step1.
Step9: Write Ubest. Stop.

Fig. 5. Layout with nodes for distance calculation



3.4.2 Numerical illustration

The algorithm is demonstrated using the eight facilities problem using the input data given in Sect. 2.1. The constraints followed in this problem are:

1. Area of facilities chosen should be within its upper and lower bound.
2. Aspect ratio of facilities chosen should be within its upper and lower bound.
3. Sum of all areas of the facilities should be within the available rectangular area.
4. Facilities should not intersect with each other.

Numerical illustration of the eight facilities problem is explained below. The notations followed are:

OBV: Objective value,
Pbest: Particle best,
Gbest: Global best,
Ubest: Universal best,
Gvel: Global velocity.

No of particles: In this study eight particles are considered.

Dimension of particles: It is equal to number of facilities (i.e., 8).

Representation of a particle: For a problem having eight facilities and are supposed to be arranged in the sequence represented in the following way with their corresponding area and aspect ratio.

Table 4. Initial solution of particles

Particle no	Facility configuration S1	OBV	
1	1 2 3 4 5 8 7 6 16 19 13 19 21 20 12 17 0.6 1.2 1.4 0.4 0.4 1.3 0.3 0.8	1300.00	Pbest [1]
2	2 3 4 7 5 6 1 8 19 15 19 12 21 18 14 19 0.6 1.3 1.0 0.4 0.4 1.1 0.5 1.0	1987.50	Pbest [2]
3	3 4 5 6 7 1 8 2 21 18 14 15 18 16 20 0.6 0.9 1.4 0.4 0.3 0.9 0.6 0.9	1791.25	Pbest [3]
4	4 5 6 8 3 1 2 7 21 20 14 16 16 18 17 15 0.4 1.3 1.3 0.7 0.6 0.9 0.5 0.7	6152.50	Pbest [4]
5	5 6 7 8 1 2 4 3 20 16 12 16 17 21 19 16 0.5 1.2 1.0 0.7 0.6 1.1 0.3 0.9	1601.25	Pbest [5]
6	6 7 8 1 4 5 2 3 16 14 16 14 22 22 17 16 0.4 0.9 1.5 0.6 0.3 1.2 0.5 0.9	1827.50	Pbest [6]
7	7 8 1 4 2 6 3 5 18 14 19 20 18 13 21 0.4 1.4 1.0 0.4 0.6 1.1 0.5 0.9	1937.50	Pbest [7]
8	2 7 8 5 3 6 1 4 14 16 18 16 18 14 22 0.6 0.9 1.5 0.5 0.6 1.1 0.5 0.7	2416.25	Pbest [8]

GBest Configuration is 1 2 3 4 5 8 7 6 :: 16 19 13 19 21 20 12 17 :: 0.6 1.2 1.4 0.4 0.4 1.3 0.3 0.8 (OBV: 1300.000000)
Corresponding particle no(s): 1

Table 5. Initial velocity: position of particles after first iteration

Particle no	Velocity V1	Cell configuration S2	OBV	
1	(5 4)	1 2 3 5 4 8 7 6	1165.00	PBest [1]
2	(2 5)	2 5 4 7 3 6 1 8	1878.75	PBest [2]
3	(7 2)	3 8 5 6 7 1 4 2	4771.25	
4	(4 2)	4 8 6 5 3 1 2 7	2332.50	PBest [4]
5	(4 3)	5 6 8 7 1 2 4 3	1360.00	PBest [5]
6	(6 2)	6 5 8 1 4 7 2 3	2898.75	PBest [6]
7	(4 3)	7 8 4 1 2 6 3 5	1480.00	PBest [7]
8	(7 8)	2 7 8 5 3 6 4 1	2485.00	PBest [8]

GBest Configuration is 1 2 3 5 4 8 7 6 (OBV: 1165.00)
Corresponding particle no(s): 1

Table 6. Velocity updating: position of particles after second iteration

Particle No	Velocity V2	Cell configuration S3	OBV
1	(5,4)	1 2 3 4 5 8 7 6	1300.00
2	(2,5) (1,7) (2,7)	7 1 3 5 2 8 4 6	1572.50
3	(7,2) (1,6) (2,8)	8 6 3 5 4 1 2 7	2547.50
4	(4,2) (1,6) (2,7)	8 7 3 2 4 1 5 6	1766.25
5	(4,3) (1,5) (2,6)	4 8 5 3 1 2 7 6	2081.25
6	(6,2) (1,4) (2,7)	5 7 3 1 4 2 8 6	2190.00
7	(4,3) (1,4) (2,5) (3,7)	3 4 7 1 2 8 5 6	1580.00
8	(7,8) (1,8) (2,8)	7 1 3 5 4 8 6 2	1020.00

GBest Configuration is 7 1 3 5 4 8 6 2 (OBV: 1020.00)
Corresponding particle no(s): 8

Example: 1 2 3 4 5 8 7 6 :: 16 19 13 19 21 20 12 17 :: 0.6 1.2 1.4 0.4 0.4 1.3 0.3 0.8.

Which means that the facilities are arranged in the sequence 1 4 2 3 6 7 8 5 with its corresponding areas 16 19 13 19 21 20 12 17 and aspect ratios 0.6 1.2 1.4 0.4 0.4 1.3 0.3 0.8.

Initial solution. Initial solution is generated randomly for all eight particles and one such set of random solution generated is shown below. Particle 1: Sequence of facilities, areas of facilities, aspect ratios of facilities.

In this order all particles are generated randomly.

Ubest. In order to extend the search to newer regions, the system is restarted from the generation of initial solution after certain number of iterations and Ubest is the best among all Gbest solutions. This process will be repeated till maximum iterations, i.e., ten iterations, are reached.

4 Results and discussion

To evaluate the efficiency of the proposed algorithm, a comparative test of the proposed algorithm was performed with genetic algorithm and Islier's algorithm.

The testing was performed for eight facilities and the results obtained using the particle swarm optimization algorithm

Table 7. Optimum results

Optimum result	
The optimum sequence of the facilities	2 3 5 8 1 7 4 6
The area of each facilities	17 13 22 17 16 16 22 17
The aspect ratio of each facilities	1.4 1.2 1.0 0.5 0.8 0.4 0.2 0.2
The objective value (OBV)	981.25

Table 8. Comparison of results of GA, improved GA and PSO

GA		Improved GA		PSO	
OBV	Mean OBV	OBV	Mean OBV	OBV	Mean OBV
1173.75	1486.00	1155.00	1313.125	981.25	1056.12

are presented below. Also, the results obtained in this work are compared with that of the results obtained using the genetic algorithm and improved GA. PSO has been recognized as an evolutionary computation technique and evolution strategy. This algorithm was run ten times in a Pentium IV system (1.1 GHz, 128 MB RAM) and it took 1.56 min CPU time. Also PSO is more simple and robust and it has taken few lines of code and requires only specification of the problem and a few parameters in order to solve it.

5 Conclusion

In this paper, the efficient facility layout algorithm was proposed for solving the facility layout problem having inner structure walls and passages. The facility layout problem with inner structure walls and passages was mathematically formulated. The layout of facilities was modeled using the particle swarm optimization algorithm. A new method was proposed for calculating distances between the facilities. A comparison with existing algorithms (GA and improved GA) was performed to evaluate the proposed algorithm's efficiency. The comparison results show that the proposed algorithm is superior to the existing one.

References

1. Tam K-Y, Li S-G (1991) A hierarchical approach to the facility layout problem. Int J Prod Res 29(1):165–184
2. Lee K-Y, Han S-N, Roh M-I (2003) An improved genetic algorithm for facility layout problems having inner structure walls and passages. Comput Oper Res 30:117–138
3. Islier AA (1998) A genetic algorithm approach for multiple criteria facility design. Int J Prod Res 36(6):1549–1569
4. Chwif L, Barretto MRP, Moscato LA (1998) A solution to the facility layout problem using simulated annealing. Comput Ind 36:125–132
5. Heragu SS, Alfa AS (1992) Experimental analysis of simulated annealing based algorithms for the layout problem. Eur J Oper Res 57(2):190–202
6. Chan WM, Chan CY, Ip WH (2002) A heuristic algorithm for machine assignment in cellular layout. Comput Ind Eng 44:49–73
7. Yang T, Kuo C (2003) Decision aiding a hierarchical AHP/DEA methodology for the facilities layout design problem. Eur J Oper Res 147:128–136
8. Mir M, Imam MH (2001) A hybrid optimization approach for layout design of un-equal area facilities. Comput Ind Eng 39:49–63
9. Kennedy J, Eberhart R (1995) Particle swarm optimization. Proc IEEE International Conference on Neural Networks, IV:1942-1948
10. Parsopoulos KE, Vrahatis MN (2001) Particle swarm optimization for imprecise problems. www.swarmintelligence.org
11. Parsopoulos E, Tasoulis DK, Vrahatis N (2003) Multiobjective optimization using parallel vector evaluated Particle swarm optimization. www.swarmintelligence.org
12. Tandon V, El-Mounayri H, Kishawy H (2002) NC end milling optimization using evolutionary computation. Int J Mach Tools Manuf 42:595–605
13. Drezner Z (1987) A heuristic procedure for the layout of a large number of facilities. Int J Manage Sci 33(7):907–915
14. Eberhart RC, Shi Y (1998) Comparison between genetic algorithms and particle swarm optimization. Proc 7th ICEC, pp 611–616
15. Hassan M-M-D, Hogg G-L, Smith D-R (1986) SHAPE:A construction algorithm for area placement evaluation. Int J Prod Res 24(5): 1283–1295
16. Ho YC, Moodie CL (1998) Machine layout with a linear single-row flowpath in an automated manufacturing system. Int J Manuf Syst 17(1):1–22