Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Industrial Engineering 47 (2004) 233–246

computers & industrial engineering

www.elsevier.com/locate/dsw

A genetic algorithm for facility layout problems of different manufacturing environments

M. Adel El-Baz

Department of Industrial and Systems Engineering, Faculty of Engineering, Zagazig University, Zagazig, Egypt

Received 6 May 2003; accepted 14 July 2004
Available online 11 September 2004

Abstract

This paper describes a genetic algorithm (GA) to solve the problem of optimal facilities layout in manufacturing systems design so that material-handling costs are minimized. The paper considers the various material flow patterns of manufacturing environments of flow shop layout, flow-line layout (single line) with multi-products, multi-line layout, semi-circular and loop layout. The effectiveness of the GA approach is evaluated with numerical examples. The cost performance is compared with other approaches. The results show the effectiveness of the GA approach as a tool to solve problems in facilities layout.

Keywords: Facility layout; Genetic algorithms; Combinatorial optimization; Material flow

1. Introduction

An effective facility layout design reduces manufacturing lead time, and increases the throughput, hence increases overall productivity and efficiency of the plant. The major types of arrangements in manufacturing systems are the process, the flow-line or single line, the multi-line, the semi-circular and the loop layout. The different layouts are illustrated in Fig. 1, where each box represents a location with the number in the top section indicates the location number and that in the lower section indicates the machine number. The single line facility layout problem is considered when multi-products with different production volume and different process routings need to be manufactured. The selection of a specific layout defines the way in which parts move from one machine to another machine. The selection

E-mail address: elbaza@mail2world.com.

0360-8352/$ - see front matter
doi:10.1016/j.cie.2004.07.001

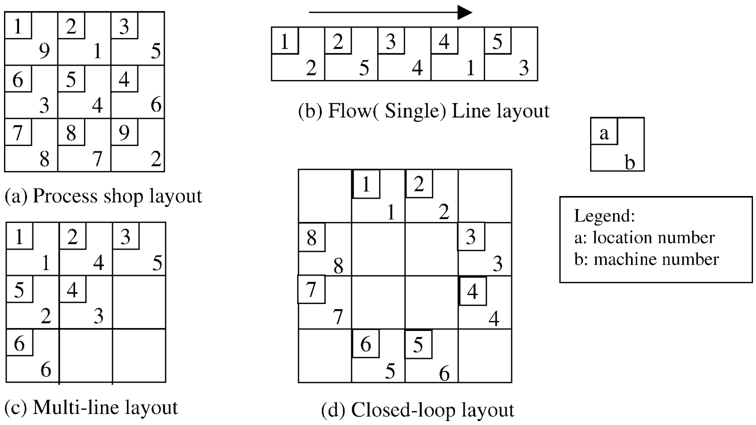

Fig. 1. Types of layout.

of a type of the machine layout is affected by a number of factors, namely the number of machines, available space, similarity of operation sequences and the material handling system used. There are many types of material handling equipment which include automated guided vehicles, conveyor systems, robots, and others. The selection of the material handling equipment is important in the design of a modern manufacturing facility.

The problem in machine layout design is to assign machines to locations within a given layout arrangement such that a given performance measure is optimized. The measure used here is the minimization of material handling cost. This problem belongs to the non-polynomial hard (NP-hard) class. The problem complexity increases exponentially with the number of possible machine locations.

This work presents a genetic algorithm (GA) approach to determine the optimal layout for the different material flow systems within a manufacturing facility. An optimization model is introduced to study the layout of machines for different patterns of material flow manufacturing environments. The effectiveness of the proposed approach is evaluated using numerical example problems benchmarked by previous researches for the flow shop type. Another numerical example applies the proposed approach to a flow-line with the multi-products is presented. The results show that the proposed approach provides an effective means to solve facility layout problems.

2. Mathematical model

The facility layout problem addressed here is the assignment of M machines to N locations in a manufacturing plant. During the manufacturing process, material flows from one machine to the next machine until all the processes are completed. The objective of solving the facility layout problem is therefore to minimize the total material handling cost of the system. To determine the material handling cost for one of the possible layout plans, the production volumes, production routings, and the cost table that qualifies the distance between a pair of machines/locations should be known. The following notations are used in the development of the objective function:

F_{ij} amount of material flow among machines i and j ($i,j=1,2,\ldots,M$).
C_{ij} unit material handling cost between locations of machines i and j ($i,j=1,2,\ldots,M$).

D_{ij} rectilinear distance between locations of machines i, and j
C total cost of material handling system.

The total cost function is defined as:

$$C = \sum_{i=1}^{M} \sum_{j=1}^{M} F_{ij} C_{ij} D_{ij} \tag{1}$$

The evaluation function considered in this paper is the minimization of material handling cost, which is the criterion most researchers prefer to apply in solving layout problems. However, the proposed approach applies to other objective functions as well.

3. Genetic algorithms

3.1. Approach

Early researchers of the facilities layout problem believed that the best approach to solutions was through the development of the general quadratic assignment problem (QAP). By using the QAP, the facilities layout problem can be optimally solved by applying implicit enumeration approaches such as cutting plane, branch and bound approaches, or other operations research techniques. The exact solution is obtained from optimal methods in a reasonable time only when the problem size is small. It has been shown that the solution times for the QAP are likely to increase exponentially as a function of the number facilities to be located. GAs have received a great deal of attention in the recent literature due to the fact that they do not rely on analytical properties of the function to be optimized which make them well suited to a wide class of optimization problems.

The GA is a stochastic search technique (Goldberg, 1989; Michalewicz, 1992). It can explore the solution space by using the concept taken from natural genetics and evolution theory (Kazerooni, Luonge, & Abhary, 1995; Tavakkoli & Shayan, 1997; Venugopal & Narendran, 1992; Zhang, Zhu, & Luo, 1997). In recent years, GA has been proposed as an innovative approach to solve the facility layout problem (Al-Hakim, 2000; Gau & Meller, 1999; Hamamoto, 1999; Islier, 1998; Rajasekharan, Peters, & Yang, 1998). GA starts with an initial set of random solutions for the problem under consideration. This set of solutions is known as the population. The individuals of the population are called ‘chromosomes’. The chromosomes of the population are evaluated according to a predefined fitness function, which in this case is the material handling cost. The chromosomes evolve through successive iterations called ‘generations’. During each generation, through merging and modifying chromosomes of a given population, creates a new population. Merging chromosomes is known as ‘crossover’ while modifying an existing one is known as ‘mutation’. Crossover is the process in which the chromosomes are mixed and matched in a random fashion to produce a pair of new chromosomes (offspring). Mutation operator is the process used to rearrange the structure of the chromosome to produce a new one. The selection of chromosomes to crossover and mutate is based on their fitness function. Once a new generation is created, deleting members of the present population to make room for the new generation forms a new population. The process is iterative until a specific stopping criterian is reached. The outline of the genetic search process used in this paper is summarized as follows:

1. Randomly generate an initial population of chromosomes with a population size P.
2. Evaluate each chromosome in the population according to the material handling cost equation.
3. Determine the average fitness for the whole population.
4. Use elitist strategy to fix the potential best number of chromosomes by deleting the worst number of each generation, and copying the best numbers into the succeeding generation. The total number of chromosomes is kept constant for computational economy and efficiency. The average of whole chromosomes acts as a guide to which chromosomes are eliminated and which of them 'gets reproduced' in the next generation. This process is applied to eliminate members with a fitness value $P(k)$ greater than 1.5 times the average of the chromosomes and copying the best number of chromosomes instead.
5. Apply the Monte Carlo selection technique to select parent chromosomes from the current population. This is used for choosing randomly the parents for the crossover and mutation.
6. Apply the crossover and mutation operators to generate a new population based on the values of crossover and mutation probabilities (p_c and p_m, respectively). The rest of the population is brought from the previous population, which has the best fitness value.
7. Check the pre-specified automatic stopping criterion. If the stopping criterion is reached, the search process stops. Otherwise, proceed to the next generation, and go to step 2. The flow chart of the GA optimization procedure is shown in Fig. 2.

3.2. String representation

The technique of GAs requires a string representation scheme (chromosomes). In this paper, the entire manufacturing plant/department is divided into N grids and each grid represents a machine location. In this study, a form of direct representation for strings is used. Fig. 3 shows different examples of different types of production plant layout with their encoded chromosomes representation. This chromosome string representation indicates one of the possible machine layout plans of each production type. Examples of flow shop layout containing 9 machines/departments, production flow line contains 5 workstations, multi-line production system contains 6 machine locations, and a closed-loop layout type of 8 machines are presented in the figure. A location assigned with the letter 'e' represented an empty area where no machine is allowed to be located.

3.3. Selection operator

The selection operator is applied to select parent chromosomes from the population. A Monte Carlo selection technique is applied. A parent selection procedure operates as follows:

1. Caculate the fitness F_{sum} (Eq. (1)) of all population members.
2. Generate a random number (n) between 0 and F_{sum}.
3. Return the first population member whose fitness, when added to the fitness of the preceding population members, is greater than or equal to n.

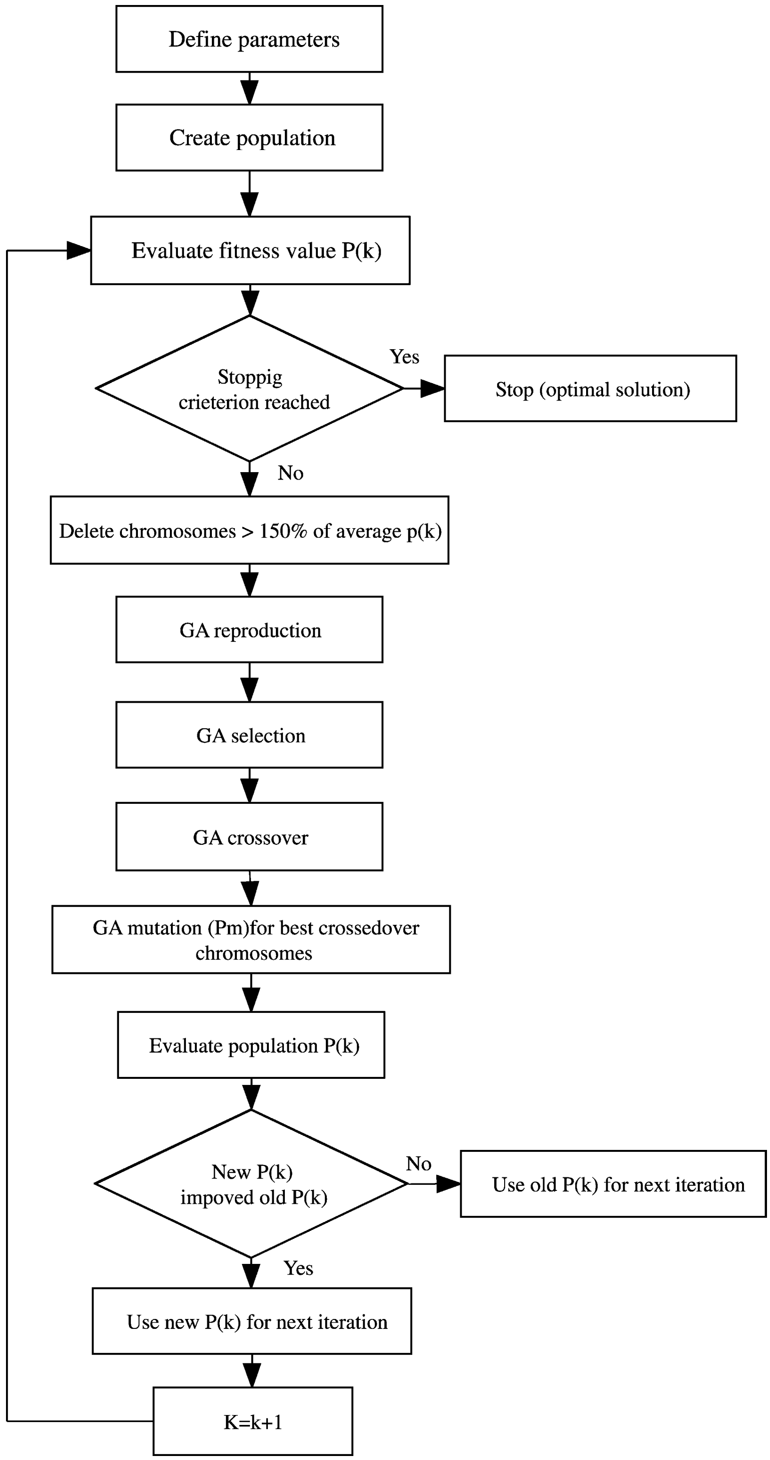

Fig. 2. Flow chart of GA optimization procedure.

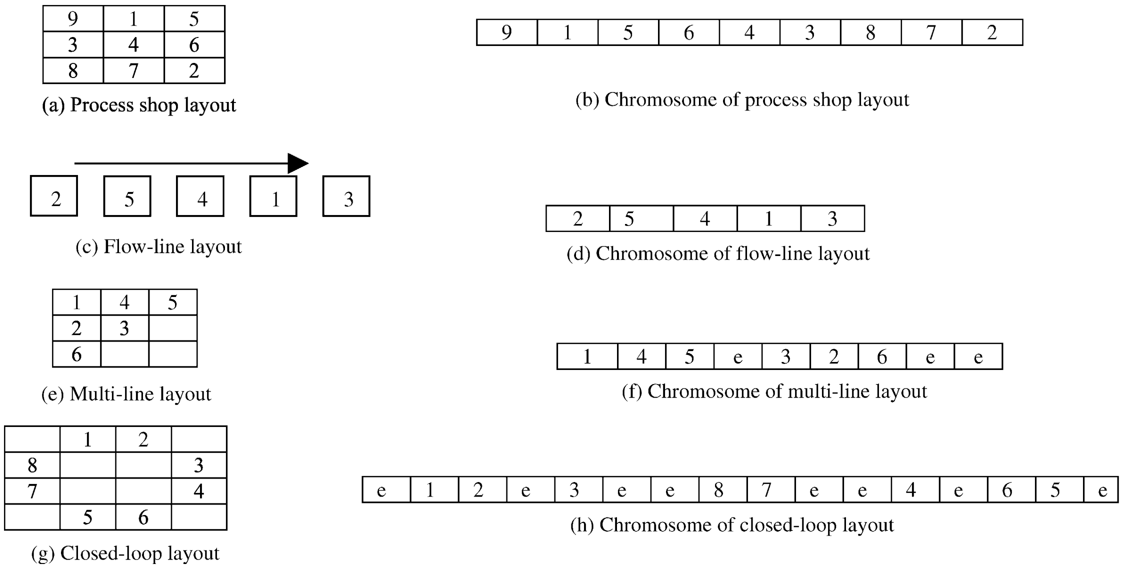

Fig. 3. Types of layout and their chromosomes representation.

4. Repeat Step 3 for the second population member and check that the new selected member is not the same as the first member.

3.4. Crossover operator

The probability of crossover p_c is the probability of applying the crossover operator to these chromosomes. The remainder of chromosomes will produce offspring chromosomes, identical to their parents. Otherwise, the selected chromosomes to crossover will be crossed to produce two offspring chromosomes by using crossover operator. In this paper a new crossover operator is proposed as follows. Consider a pair of parent chromosomes (P_1, P_2) shown below:

P_1	1	4	7	5	2	8	3	6	9
P_2	4	1	6	7	5	2	9	3	8

First, select two random numbers to be aligned to the parents string. Suppose the two random numbers in this example are 4 and 7. The genes with double-lined borders within the cutting section, i.e. (5,2,8,3) in P_1 and (7,5,2,9) in P_2, are exchanged so that a portion of genetic codes from P_1 is transferred to P_2, and vice versa. The structures of the resultant chromosomes then become:

P_1^-	1	4	7	7	5	2	9	6	9
P_2^-	4	1	6	5	2	8	3	3	8

At this stage, several genes are found to exist in more than one position in the resultant chromosomes (e.g. 7, 9 in P_1^-, and 3, 8 in P_2^-). This would mean that a machine represented more than one position has more than one machine location in the layout plan. A backward replacement procedure can then be implemented to change the values of those repeated genes outside the cutting section. The repeated genes outside the cutting section of P_1^- can be replaced by changing 7 in the third gene by 5, since the gene 5 in P_1 is produced by exchanging with the gene 7 in P_2. Gene 5 is however also repeated in P_1^- in the cutting section. So, this gene is changed again to 2. The gene 9 in P_1^- can also be changed by the gene 3 to obtain the offspring c_1. The repeated genes outside the cutting section in P_2^- can also be replaced by changing gene 8 to 7 (the combined result of changing from 8 to 2, 2 to 5 and 5 to 7) and the gene 3 to 9. Thus the offspring chromosome c_1, and c_2 will be:

c_1	1	4	8	7	5	2	9	6	3
c_2	4	1	6	5	2	8	3	9	7

The cutting section in the proposed crossover operator is selected randomly by two genes in the string. If the chromosome has large size genes, the cutting section is differing from small to large, which reflects the flexibility of the proposed approach. Also, for the flexibility of the proposed crossover, the empty location (letter 'e' in the chromosome) in the proposed approach is treated as any location number without any change in the procedure. The algorithm performs checking after the first step of exchanging cutting section in the parents to replace the repeated location outside the cutting section in the chromosome as explained earlier and the number of empty locations in the offspring chromosomes before crossover will be the same after crossover.

3.5. Mutation operator

The mutation operator is used to rearrange the structure of a chromosome. In this study, the swap mutation is used, which is simply selecting two genes at random and swapping their contents. The probability of mutating a single gene is called the probability of mutation p_m, which is usually a small number. Mutation helps to increase the searching power. In order to explain the need of mutation, consider the case where reproduction or crossover may not produce a good solution to a problem. During the creation of a generation it is possible that the entire population of strings is missing a vital gene of information that is important for determining the correct or the most nearly optimum solution (e.g. need only to swap one gene with another gene). Future generations that would be created using reproduction and crossover would not be able to alleviate this problem. Here, mutation becomes important.

3.6. Stopping criterion

The program is terminated when either the maximum number of generations is reached, or until the population converges.

4. Numerical examples

4.1. Sample example I

A comparative evaluation of the proposed approach is made using bench-mark numerical examples. The first example is taken from Chan and Tansri (1994) and compared with the work of Mak, Wong,

Table 1
Flow of materials between machines

From/to	2	3	4	5	6	7	8	9
1	100	3	0	6	35	190	14	12
2		6	8	109	78	1	1	104
3			0	0	17	100	1	31
4				100	1	247	178	1
5					1	10	1	79
6						0	1	0
7							0	0
8								12

and Chan (1998) that used the same example to evaluate their work. The plant flow of materials between machines and material handling cost between machines are presented in Tables 1 and 2, respectively. The plant configuration layout is a 3×3 grid. The general guidelines proposed by both the works of Chan and Tansri and Mak et al. is considered to evaluate the performance of the proposed approach.

Their work conducted 19 sets of experiments to determine an appropriate combination of the population size P and the generation size G. The experimental results are compared with the global optimal solutions in order to evaluate the performance of the search process for the different combinations of the population and generation sizes. The exhaustive search method is applied to determine the global optimal solution and eight optimal machine layouts are obtained (Mak et al., 1998).

Fig. 4 shows the resulting optimal machine layouts giving a material handling cost of $4818. In addition, the work done by Chan and Tansri (1994) is included three widely used crossover operators. They are the partially mapped crossover (PMX), the order crossover (OX), and the cycle crossover (CX). These authors reported that the PMX operator provided excellent results. The proposed approach used in this paper is therefore compared with the PMX crossover approach, in addition to the approach proposed by Mak et al. (1998).

The experiments designed in both works used a set of genetic parameters values. The genetic parameters are the population size, P, the generation size, G, the percentage of replication of well-performed chromosomes in each generation, R, the probability of crossover p_c, and the probability of mutation p_m. In this example, using 9 machines, there are 362 880 possible solutions in the solution

Table 2
Material handling cost between machines

From/to	2	3	4	5	6	7	8	9
1	1	2	3	3	4	2	6	7
2		12	4	7	5	8	6	5
3			5	9	1	1	1	1
4				1	1	1	4	6
5					1	1	1	1
6						1	4	6
7							7	1
8								1

6	2	5
1	9	8
7	3	4

7	1	6
3	9	2
4	8	5

4	3	7
8	9	1
5	2	6

5	8	4
2	9	3
6	1	7

5	2	6
8	9	1
4	3	7

4	8	5
3	9	2
7	1	6

7	3	4
1	9	8
6	2	5

6	1	7
2	9	3
5	8	4

Fig. 4. Optimal facility layouts for sample example I.

space (9!). However, the reason for using GAs is to obtain reasonable solutions by minimal evaluations. Hence, it is appropriate to limit the total number of evaluations in each experiment to less than 3% of the total number of solutions in the solution space. Table 3 lists the different suggested combinations of the population and generation sizes. Each experiment is run 10 times. Both of the works of Chan and Tansri (1994) and Mak et al. (1998) used the genetic parameters of $R=5\%$, $p_c=0.6$, and $p_m=0.001$ and have reported that these work well with these owen parameters. The approach proposed here uses a cut-off of 1.5 times the average fitness value of the whole population of each generation as a replication ratio, $p_c=0.9$, and $p_m=0.1$ per populations. The experimental results shown in Table 3 are expressed in terms of:

1. The material handling cost of the best solution among the 10 runs (Best).
2. The average of the best material handling costs among the 10 runs (Avg).
3. The number of runs needed to obtain one of the eight optimal solutions (#).

In general, an increase in the population and generation sizes can provides better solutions since the number of sampling solutions from the solution space is enlarged. The general cost performance for the three different approaches is studied with the used sampling solution space. The different combinations of population P and generations G of the 19 experiments represent a different solution space.

The results of the study show that the proposed approach is more efficient than the two other approaches when solving facility layout problem. The results listed in Table 3 show that the proposed approach produces 81 successful runs to obtain one of the eight optimal solutions among 37 of the work of Mak et al. (1998) and 23 among the work of Chan and Tansri (1994). The proposed approach success to obtain the optimal solution in 17 experiments from the 19 among 12 of the work of Mak et al. and 8 of the work of Chan and Tansri. The approach proposed here provides good results for the different combinations of the population and generation sizes.

4.2. Sample example II

This example is taken from Kazerooni, Luonge, Abhary, Chan, and Pun (1996) and the results are compared with the works of Chan and Tansri (1994) and Mak et al. (1998). Table 4 presents the part list and the corresponding production data of the parts. The material handling costs are assumed to be the same among machines. The problem seeks to locate 24 types of machines in a 5×6 machine location grid, giving 2.65×10^{32} (30!) possible solution in the solution space making the determination of the optimal solution by using the exhaustive search method is impossible in this case. The proposed approach is applied to solve the problem with the genetic parameters $P=200$, $G=40$ $p_c=0.9$, and $p_m=$ 0.1 per population. For comparison, the work proposed by Mak et al. and the three widely used crossover

Table 3
The experimental results for sample problem I

Exp.	*P*	*G*	No. of trials	Proposed approach			Mak et al.			PMX (Chan and Tansri)		
				Best	Avg.	#	Best	Avg.	#	Best	Avg.	#
1	20	10	200	5039	5310.1	0	5233	5504.4	0	4938	5434.8	0
2	40	10	400	4818	5231.9	1	5040	5286.7	0	5039	5263.8	0
3	100	10	1000	4818	4961	2	4818	5024.8	1	4938	5164.9	0
4	200	10	2000	4818	4895.9	5	4818	4891.4	2	4818	4966.8	2
5	500	10	5000	4818	4822	9	4818	4833.2	7	4818	4892.3	5
6	20	20	400	4872	5172.9	0	5225	5481.2	0	4938	5402.1	0
7	40	20	800	4818	5052	1	4927	5174.6	0	4992	5184.6	0
8	100	20	2000	4818	4855.2	4	4818	4889.1	4	4818	4991.7	2
9	200	20	4000	4818	4842.1	6	4818	4846.5	5	4818	4919.8	2
10	20	40	800	4818	5074.1	2	5225	5462.2	0	4938	5402.1	0
11	40	40	1600	4818	4979.5	2	4927	5163.8	0	4992	5180.7	0
12	100	40	4000	4818	4842.8	7	4818	4871.4	4	4818	4919.5	3
13	200	40	8000	4818	4842.1	6	4818	8440	5	4818	4887.9	4
14	20	100	2000	4818	4940.9	5	5225	5453	0	4938	5337	0
15	40	100	4000	4818	4862.7	6	4818	5141.6	1	4927	5122.4	0
16	100	100	10 000	4818	4826.8	8	4818	4866	5	4818	4863.9	4
17	20	200	4000	4818	4893.6	6	4818	5303.9	1	4938	5224.6	0
18	40	200	8000	4818	4858.3	7	4818	5141.4	1	4862	5088.4	0
19	10	500	5000	4818	4983.7	4	4818	5184.3	1	4818	5166.1	1
						81			37			23

Table 4
Part list and production data for sample example II

Product	Volume	Production routing					
		mch_1	mch_2	mch_3	mch_4	mch_5	mch_6
P01	130	22	1	13	21		
P02	150	3	20	24	0		
P03	125	14	7	23	24		
P04	145	15	6	18	8	12	
P05	65	15	6	18	8	12	5
P06	78	9	17	10			
P07	95	9	17	10			
P08	160	4	16	0			
P09	85	22	1	13			
P10	105	2	11	19	5	21	
P11	130	3	20				
P12	140	3	20				
P13	150	2	11	19			
P14	185	2	11	19	5		
P15	78	3	20	0	0		
P16	95	22	1	13	21		
P17	160	1	13	22			
P18	85	15	6	18	8	12	
P19	105	4	16				
P20	130	10	17	12			
P21	105	4	16				
P22	130	2	5	11	19		
P23	140	3					
P24	150	20	12				
P25	185	7	14	23			
P26	145	15	6	18	8	10	
P27	65	15	6	18	8	12	
P28	78	4					
P29	95	9	17				
P30	160	6	18	8	12		
P31	85	3	20	17	0		
P32	105	14	7	23	24	16	
P33	130	22	1	13	21	2	
P34	150	3	20				
P35	125	11	19	5			
P36	145	20	12	21			
P37	65	16	11	14			
P38	78	4	16	0			
P39	95	4	16	0			
P40	160	1	13	19			

operation namely the PMX, OX, and CX operators listed in Mak et al. are considered to evaluate the proposed approach. They used a genetic parameters of $P=200$, $G=40$, $R=4\%$, $p_c=0.6$, and $p_m=0.001$. Thirty runs of genetic searches are conducted for each proposed solution and the results are shown in Table 5.

Table 5
Results of solving sample problem II

Method	Best (30 runs)	Avg. (30 runs)	Worst (30 runs)	Successful hits
Proposed approach	11 862	11 871.8	13 373	23
Mak et al.	12 892	15 087.7	18 657	11
PMX	14 947	18 355.9	20 654	0
OX	22 406	24 301.7	26 926	0
CX	14 717	17 216.5	20 654	0

	3	24	16	4	22
9	20	23	5	13	1
17	12	7	19	21	26
10	8	14	11	2	25
	18	6	15		

Fig. 5. The best facility layout of sample example II.

Table 5 gives a comparison between the different methods indicating that the approach proposed here obtains a more efficient solution as compared to the other methods. The results show that during the 30 runs, the PMX, OX, and CX operators do not produce the best solution obtained by the work of Mak et al. (1998) and they obtain their best solution 11 times from the 30 runs. The proposed approach also obtains a better solution than the work of Mak et al. and obtains this result 23 out of the 30 runs. Again, the proposed approach still obtains good results while the values of genetic parameter are not

Table 6
Input information for 18 parts and 12 machines

Part	Production volume	Machine routs
P01	100	M1–M4–M2–M6
P02	120	M3–M5–M12–M10
P03	50	M2–M4–M12–M6
P04	45	M5–M8–M10
P05	60	M3–M5–M12–M6
P06	80	M4–M2–M4–M6
P07	90	M1–M5–M9
P08	120	M3–M7–M10–M4–M8
P09	140	M1–M4–M6
P10	180	M3–M12–M8–M10
P11	80	M2–M6–M2–M4–M6
P12	60	M11–M9–M10–M8
P13	70	M1–M4–M5–M7
P14	150	M2–M4–M6–M2–M6
P15	120	M3–M7–M9–M10
P16	120	M3–M10–M12–M9–M12
P17	100	M5–M10–M8–M9–M12
P18	90	M2–M8–M9–M10

6	2	4	1	8	10	12	5	9	3	7	11

Fig. 6. Layout of sample example III.

affecting the quality of the solutions. The best facility layout obtained by the proposed approach is presented in Fig. 5.

4.3. Sample example III

This example applies the proposed approach on a flow line with multi products. The flow line layout configuration arranged the machines along a straight track with a fixed path material handling equipment. The ordering of machines in the layout is made to be as close as possible to the sequence in which parts to be processed in the layout visit the machines. Table 6 presents the input data for this example. The material handling costs are assumed to be the same among machines. The genetic parameters used in this example are $P=100$, $G=40$, $p_c=0.9$, $p_m=0.1$ per population, and a number of runs $=10$. The material handling cost obtained from this example is 11 440. The solution layout is presented in Fig. 6.

5. Conclusion

This paper proposes an approach using GAs to solve facility layout problems. The proposed approach considers different types of manufacturing layout environments. They include flow shop layout, flow-line layout with multi-products, multi-line layout, semi-circular and loop layout. The proposed GA approach produces the optimal machine layout which minimizes the total material handling cost. The effectiveness of the proposed approach has been examined by using two benchmark problems, the first used by Chan and Tansri (1994), and the second used by Kazerooni et al. (1996). The results are also compared with the work of Mak et al. (1998). The comparison indicates that the proposed approach is more efficient and has a higher chance of obtaining the best solution for the facility layout problem. A third example to solve the facility layout problem of flow line with multi products shows that the proposed approach provides good results in all sample problems. In addition, it has an advantage for tackling different manufacturing types of layout environments.

References

Al-Hakim, L. (2000). On solving facility layout problems using genetic algorithms. *International Journal of Production Research*, *38*(11), 2573–2582.

Chan, K.c., & Tansri, H. (1994). A study of genetic crossover operations on the facility layout problem. *Computers and Industrial Engineering*, *26*(3), 537–550.

Gau, K. Y., & Meller, R. D. (1999). An iterative facility layout algorithm. *International Journal of Production Research*, *37*(16), 3739–3758.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. New York: Addison-Wesley.

Hamamoto, S. (1999). Development and validation of genetic algorithm-based facility layout a case study in the pharmaceutical industry. *International Journal of Production Research*, *37*(4), 749–768.

Islier, A. A. (1998). A genetic algorithm approach for multiple criteria facility layout design. *International Journal of Production Research*, *36*(6), 1549–1569.

Kazerooni, M., Luonge, L., & Abhary, K. (1995). Cell formation using genetic algorithms. *International Journal of Flexible Automation and Integrated Manufacturing*, *3*(3–4), 283–299.

Kazerooni, M., Luonge, L., Abhary, K., Chan, F., & Pun, F. (1996). An integrated method for cell layout problem using genetic algorithms. *Proceedings of the 12th International Conference on CAD/CAM Robotics and Factories of the Future,* pp. 752–762.

Mak, K. L., Wong, Y. S., & Chan, T. S. (1998). A genetic algorithm for facility layout problems. *Journal of Computer Integrated Manufacturing Systems*, *1*(1–2), 113–123.

Michalewicz, Z. (1992). *Genetic algorithms + data structures = evolution programs*. Berlin: Springer.

Rajasekharan, M., Peters, B. A., & Yang, T. (1998). A genetic algorithm for facility layout design in flexible manufacturing systems. *International Journal of Production Research*, *36*(1), 95–110.

Tavakkoli, R., & Shayan, E. (1997). Facilities layout design by genetic algorithms. *Proceeding of 22nd International Conference on Computer and Industrial Engineering,* pp. 415–418.

Venugopal, V., & Narendran, T. A. (1992). Genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Journal of Computers and Industrial Engineering*, *22*, 469–480.

Zhang, Y., Zhu, X., & Luo, Y. (1997). Applying genetic algorithms to task planning of multi-agent systems. *Proceeding of 22nd International Conference on Computer and Industrial Engineering,* pp. 411–414.