Computer Integrated Manufacturing Systems Vol. 11, No. 1-2, p. 113-127, 1998

0951-5240/98 $ — see front matter
ELSEVIER
PII: S0951-5240(98)00018-4

# A genetic algorithm for facility layout problems

**K. L. Mak, Y. S. Wong and F. T. S. Chan***

*Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, Hong Kong*

**This paper is concerned with the application of the technique of genetic algorithms to solve the problem of optimal facilities' layout in manufacturing systems design. A mathematical model is developed to examine the machines' layout and the pattern of material flow for the typical job shop and flow shop manufacturing environments. The analysis also considers various practical aspects, such as the constraints of restricted areas and reserved machinery locations, and also the irregularity of the shapes of manufacturing plants, etc. A genetic approach is developed to provide the optimal solution to the facilities' layout problem. The effectiveness of the proposed approach is evaluated with numerical examples. Indeed, the results indicate that the proposed approach provides an effective means to solve problems in facilities' layout.** 

Keywords: genetic algorithm, optimal facility layout, job shop, flow shop

## Introduction

The facility layout design has been regarded as the key to improve plant productivity. Its objective is to obtain the most effective machine arrangement, so that the material handling costs are minimized. Indeed, an effective facility layout can reduce significantly the manufacturing lead time[1-3]. However, the facility layout problem belongs to the class of non-polynomial hard (NP-hard) problems which are unsolvable in polynomial time[4]. It suggests that the problem's complexity increases exponentially with the number of machinery locations. For instance, a system consisting of $M$ machines and $N$ machinery locations ($N \geqslant M$) will comprise a solution space with the size $N$. When $N$ is large, it is difficult, if not impossible, to produce the optimal solution within a reasonable time, even with the support of a powerful computer[5].

In the past, the facility layout problem was formulated as a quadratic assignment problem[6]. Since then, a number of attempts have been published to solve the problem by using various techniques[7], including the tree search algorithm[8], and the binary mixed integer programming technique[9]. However, such approaches still require a substantial amount of computational effort when the problem size is large. In recent years, the Group Technology (GT) that exploits the similarities in product processing has been proposed as an innovative approach to solve the problem. In order to simplify the material flow patterns, numerous grouping techniques have been developed to configure the layout of machines. In addition, various clustering heuristics are proposed[10], and the unit-valued entries of the part-machine incidence matrix are grouped into blocks along the matrix's diagonal. Another methodology involves the grouping of machines progressively, drawing reference from the similarity coefficients that was first proposed by McAuley[11], and later improved by a number of researchers, such as Seifoddini and Wolfe[12]. In addition, a machine chain similarity coefficient was recently proposed by Kazerooni et al.[13], so as to accommodate both the direct and indirect relationships among machines simultaneously. The network decomposition heuristic and mathematical programming are also frequently used to solve the problem[14,15]. However, these methods usually over-simplify the problem to such an extent that the resulting solution becomes unrealistic. Most grouping methods do not solve the facility layout

*Corresponding author.

problem directly. In fact, these methods generally assume that the problem can be solved easily once the machine groups have been established. In fact, the whole problem is decomposed into a number of sub-problems, which are easier to solve. However, a significant body of literature on facility layout[16,17] has pointed out the importance of machine arrangement to achieve line flows and to reduce materials handling costs. An effective facility layout has a definitely significant impact on traffic congestion and material flow patterns. Although the facility layout problem can be solved after the establishment of machine groups, the original scope of the problem is inevitably limited by this approach.

The present research aims at developing a general methodology to determine the optimal layout of machines by simplifying the material flow within a manufacturing plant. A mathematical model is introduced to study the layout of machines and the pattern of material flow for the typical job shop and flow shop manufacturing environments. In addition, the analysis also considers the capacities of the different manufacturing resources, the constraints of restricted areas and reserved machinery locations, the irregularity of the shape of manufacturing plants, and so on. An approach derived from genetic algorithms[18,19] is developed to provide the optimal solution to the facility layout problem. Unlike some of the existing genetic approaches[20,21], the proposed approach determines the optimal facility layout without initial involvement of the clustering of machines and parts. The effectiveness of the proposed approach is evaluated by using benchmark problems excerpted from Chan and Tansri[22], and Kazerooni et al.[20]. Indeed, the theoretical results so obtained show that the proposed approach provides an effective means to solve facility layout problems.

## Mathematical model

Facility layout design is the study of the assignment of $M$ machines to $N$ machinery locations ($N \geqslant M$) in a manufacturing plant. During the manufacturing process, materials flow from one machine to the next appropriate machines, 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. Indeed, the following aspects of the manufacturing system are considered:

(a) Plant configuration layout, including information of the restricted areas and the reserved machinery locations. In addition, the plant configuration layout can be irregular in shape.
(b) Cost table that qualifies the distance based material handling costs between a pair of machines.
(c) Part-wise production data, which provides information about production volumes, production routings, and the unit processing time for each manufacturing operation of the part type $i$ ($i = 1,2,\ldots,I$) during the planning period.

The following notations are used in the development of the mathematical model:

- $C_{m_1m_2}$ unit material handling cost between machines $m_1$ and $m_2$ ($m_1,m_2 = 1,2,\ldots,M$ and $m_1 \neq m_2$) per unit distance
- $D_{n_1n_2}$ rectangular distance between machinery locations $n_1$ and $n_2$ ($n_1,n_2 = 1,2,\ldots,N$ and $n_1 \neq n_2$)
- $E_m$ effective manufacturing time (capacity) of machine type $m$ ($m = 1,2,\ldots,S$)
- $F_{m_1m_2}$ amount of material flow among machines $m_1$ and $m_2$ ($m_1,m_2 = 1,2,\ldots,M$ and $m_1 \neq m_2$)
- $I$ number of part types manufactured by the manufacturing plant
- $L(m)$ machinery location, where machine $m$ ($m = 1,2,\ldots,M$) is assigned to
- $M$ total number of machines contained in the manufacturing system
- $N$ number of machinery locations contained in the plant configuration layout
- $Q_m$ number of type $m$ machines ($m = 1,2,\ldots,S$) required in the manufacturing system
- $S$ number of machine types
- $t_{im}$ unit processing time for part type $i$ ($i = 1,2,\ldots,I$) on machine type $m$ ($m = 1,2,\ldots,S$)
- $v_i$ production volume of part type $i$ ($i = 1,2,\ldots,I$) demanded during the planning period

The demand of the parts are assumed to be known and fixed during the planning period. In order to prevent any shortage of parts, the production capacity must be sufficiently large. Hence, the minimum number of machines $Q_m$ required in the manufacturing system is calculated as follows:

$$Q_m = \left\lceil \sum_{i=1}^{I} v_i t_{im} / E_m \right\rceil (m = 1,2,\ldots,S) \tag{1}$$

where $\lceil X \rceil$ is an integer which is just greater than the real number $X$.

In some cases, when rectangular distances are more appropriate than straight-line distances to be used in industrial settings[23], the cost function is defined as:

$$\text{Total cost} = \phi = \sum_{m_1=1}^{M} \sum_{m_2=1}^{M} F_{m_1m_2} C_{m_1m_2} D_{L(m_1)L(m_2)} \quad (m_1 \neq m_2) \tag{2}$$

The objective is to obtain an optimal facility layout plan for the machines by minimizing the total material handling cost incurred in the system.

## Genetic algorithms

When introduced initially by Holland[18], the genetic algorithm is a stochastic global search technique. It can explore the solution space intelligently by using the concept taken from the natural genetics and evolution theory[19]. Indeed, the genetic algorithm has been demonstrated to be robust and effective in various task domains, both theoretically and empirically, and even in the presence of non-linearity, multimodality, noise, etc., in the model describing the problem[24]. In the search process, candidate solutions in the solution space are encoded in the form of symbolic strings known as chromosomes. The simulation of genetic evolutionary processes is conducted in a pool of chromosomes. This pool is known as the population, and the number of chromosomes contained within the population is called the population size, $P$. The simulation of genetic evolutionary processes is performed on an iterative basis. The search algorithm extracts and analyses the topological information of the searched space, and can therefore guide the search to advance along a promising direction. Each iteration of the search process is called a generation. Indeed, the outline of the genetic search process used in this paper is summarized as follows:

*Step 1.* Generate an initial population of chromosomes randomly with a population size of $P$.
*Step 2.* Decode all chromosomes and evaluate the objective function values of their corresponding candidate solutions.
*Step 3.* Determine the fitness values of the chromosomes by using the objective function values so obtained.
*Step 4.* Remove the worst $\lfloor P \times R \rfloor$ chromosomes in accordance with their fitness values, and replace them by duplicating and inserting the best $\lfloor P \times R \rfloor$ chromosomes into the current population. $\lfloor X \rfloor$ is an integer which is just smaller than the real number $X$, and $R$ is the percentage of replication of the well-performed chromosomes in the current generation.
*Step 5.* Apply the selection operator to select $P$ chromosomes from the current population. The selected chromosomes are placed in a mating pool as parent chromosomes.
*Step 6.* Choose a pair of parent chromosomes from the mating pool without replacement. The crossover and mutation operators are then applied to produce a pair of new chromosomes.
*Step 7.* Insert the new chromosomes into a new population. If the population is not full, go to *Step 6.*
*Step 8.* Check the pre-specified automatic stopping criterion. If the stopping criterion is reached, the search process stops. The overall best chromosome will be selected and decoded. The corresponding candidate solution will be chosen as the final solution. Otherwise, proceed to the next generation with the current population replaced by the new one, and go to *Step 2.*

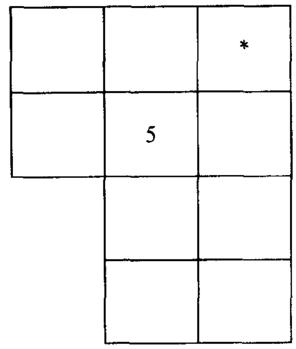


**Figure 1** An example of a plant configuration layout.

### *String representation*

In order to apply the technique of genetic algorithms to solve facility layout problems, a string representation scheme is required to encode the candidate solutions in the solution space in the form of symbolic strings, called chromosomes. In this paper, the entire manufacturing plant is divided into $N$ grids (say $n_x \times n_y$), and each grid represents a machinery location. The entire facility layout plan can thus be encoded as a chromosome by using a $n_x \times n_y$ matrix. This string representation scheme can be conveniently illustrated by using a simple example. *Figure 1* shows a plant containing 10 grids, with two of the grids labeled by a symbol '*' and a number '5'. A location assigned with a symbol '*' represents a restricted area where no machine is allowed to be located, whilst a location assigned with a number represents a reserved machinery location where only the specified machine is allowed to be placed. If there are seven machines in the manufacturing system, one of the possible facility layout plans is shown in *Figure 2*. This plan can be encoded as a chromosome, $S$, by using a $4 \times 3$ matrix as shown below:

$$S = \begin{bmatrix} 7 & 6 & -1 \\ 0 & 5 & 3 \\ -1 & 1 & 4 \\ -1 & 0 & 2 \end{bmatrix} \tag{2}$$

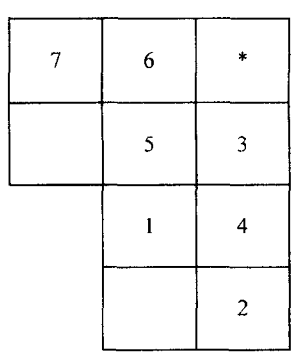


**Figure 2** A particular facility layout plan.

**Table 1 Part list and production data**

| | P1 | P2 | P3 |
|---|---|---|---|
| Production routing | 1-3-5 or 1-4-5 | 2-3-6 or 2-3-7 or 2-4-6 or 2-4-7 | 5-6 or 5-7 |
| Unit op. proc. Time (min.) | 6-2-5 | 5-8-5 | 2-4 |
| Production volume | 20 | 10 | 10 |

where $-1$ and 0 represent a restricted area and a dummy machine respectively.

*Fitness function*

As mentioned previously, the objective of solving a facility layout problem is to obtain an optimal facility layout plan by minimizing the total material handling cost (eqn (2)) incurred in the system. The material handling cost, however, does not depend solely on the facility layout plan. The cost is also determined by the type of machine loading policy to be used. Some machine loading policies commonly practised in industry include the First-Come-First-Serve (FCFS), the Shortest Processing Time First (SPTF), and the Longest Processing Time First (LPTF), etc. Nevertheless, the management usually establishes its own favourite machine loading policy in accordance with the manufacturing nature of the parts. In general, parts are manufactured by undergoing a series of machinery operations based on their production routings. A list of machinery operations is established for each type of machine to record its loading details. This includes the information of the part types that are going to be manufactured, such as their production volumes, and their unit processing times, etc. When there is more than one machine (e.g., $m_1, m_2, \ldots m_{Q_m}$) for a machine of type $m$ (where $m = 1, 2, \ldots, S$), the application of an FCFS policy means that the allocation of the machining operations of a part to machines begins from the first machining operation, i.e., from the top of the machining operation list, to the last machining operation at the bottom of the list. It should be noted, however, that whenever a machine is fully loaded during the allocation process, the remaining machining operations will be allocated to the next available machine, until all the listed machinery operations are successfully allocated. On the other hand, an SPTF policy means that the allocation of the machining operations of a part to machines depends on the processing times required. The machining operation with the shortest processing time is allocated first, and then followed by the machining operation with the next shortest processing time, with the rest following suit, until all machining operations in the list are allocated. The LPTF policy, however, works in the opposite way, where the machining operation with the longest processing time is given the highest priority to be allocated during the allocation process. The machining operation with the next longest processing time follows next, and so on, until all the machining operations are allocated.

A simple example is used to illustrate how different machine loading policies affect the material handling cost of a facility layout plan. In this example, there are seven machines in the manufacturing system, where the machines listed in each of the following brackets belong to the same machine type: (3, 4) and (6, 7). *Figure 2* presents the machine's location in the facility layout plan. *Table 1* shows a part list containing the information of the part types, their production routings, the operation processing time, and the production volumes. *Table 2* shows a cost table containing the information of unit material handling cost and the effective capacity of each machine type. The machine loading plans derived from the different machine loading policies are summarized in *Table 3*. *Table 3* also presents the slight variations among the different machine loading plans. As a result, the material flow patterns under the policies of FCFS, SPTF, and LPTF are also different, and their corresponding total material handling costs (eqn (2)) are equal to \$520, \$524, and \$500 respectively. The LPTF policy leads to the minimum material handling cost.

The total material handling cost of a candidate solution is then converted to the fitness value of its corresponding chromosome. In this paper, the fitness value of a chromosome is determined by using the following equation:

$$\text{Fitness} = \Phi = 1/\phi \tag{3}$$

where $\phi$ is the total material handling cost. A candidate solution with a small total material handling cost will lead to a chromosome with a large fitness value. As a result, the chromosome is given a greater chance to be selected as a parent chromosome.

*Selection, crossover and mutation operators*

*Selection operator*

The selection operator is used to guide the search direction of the genetic search process. It leads to an overall improvement of the chromosomes' cost

**Table 2 Material handling cost (\$ per trip) and effective capacities of machines**

| From/To | 2 | 3 or 4 | 5 | 6 or 7 | Effective capacity (min) |
|---|---|---|---|---|---|
| 1 | 0 | 20 | 0 | 0 | 150 |
| 2 | | 10 | 0 | 0 | 100 |
| 3 or 4 | | | 20 | 10 | 70 |
| 5 | | | | 10 | 120 |
| 6 or 7 | | | | – | 70 |

**Table 3 Machine loading plans**

| FCFS | | | | SPTF | | | | LPTF | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *m* | Part type | Unit proc. time (min) | Prod. vol. | *m* | Part type | Unit proc. time (min) | Prod. vol. | *m* | Part type | Unit proc. time (min) | Prod. vol. |
| 1 | Part 1 | 6 | 20 | 1 | Part 1 | 6 | 20 | 1 | Part 1 | 6 | 20 |
| 2 | Part 2 | 5 | 10 | 2 | Part 2 | 5 | 10 | 2 | Part 2 | 5 | 10 |
| 3 | Part 1 | 2 | 20 | 3 | Part 1 | 2 | 20 | 3 | Part 2 | 8 | 8 |
| | Part 2 | 8 | 3 | | Part 2 | 8 | 3 | 4 | Part 2 | 8 | 2 |
| 4 | Part 2 | 8 | 7 | 4 | Part 2 | 8 | 7 | | Part 1 | 2 | 20 |
| 5 | Part 1 | 5 | 20 | 5 | Part 3 | 2 | 10 | 5 | Part 1 | 5 | 20 |
| | Part 3 | 2 | 10 | | Part 1 | 5 | 20 | | Part 3 | 2 | 10 |
| 6 | Part 2 | 5 | 10 | 6 | Part 3 | 4 | 10 | 6 | Part 2 | 5 | 10 |
| | Part 3 | 4 | 5 | | Part 2 | 5 | 6 | | Part 3 | 4 | 5 |
| 7 | Part 3 | 4 | 5 | 7 | Part 2 | 5 | 4 | 7 | Part 3 | 4 | 5 |

performance as the search proceeds. The potential chromosomes with higher fitness values are given higher chances to be selected as parents to breed new chromosomes. The parent chromosomes are placed in a mating pool where crossover and mutation take place. In each generation, the worst $\lfloor P \times R \rfloor$ chromosomes are removed from the current population in accordance with their fitness values. The best $\lfloor P \times R \rfloor$ chromosomes are then duplicated and inserted into the same population to replace the rejected chromosomes. Afterwards, the selection operator is applied to select parent chromosomes from the population in accordance with the selection parameter, $\lambda_{\text{Select}}$, of the chromosome. The selection parameter, $\lambda_{\text{Select}}$, of a chromosome is defined as:

$$\lambda_{\text{Select}} = \Phi / \sum \Phi \tag{4}$$

where $\Phi$ and $\sum \Phi$ are the fitness values of the chromosome, and the sum of fitness values over the entire population respectively.

In this paper, the selection scheme of the remainder stochastic sampling with replacement proposed by Brindle[25] is adopted. In this case, the expected number of chromosomes, $e_p$, for each chromosome in the mating pool is calculated by:

$$e_p = \lambda_{\text{Select}} \times P \tag{5}$$

Chromosomes are then reproduced and placed into the mating pool with the number of samples equal to the integer part of their $e_p$ values. The fractional parts of the $e_p$ values are used to calculate weights in the traditional roulette wheel selection procedure with the probability of selection, $P_{\text{Select}}$, for each chromosome defined as:

$$P_{\text{Select}} = \text{frac}(e_p) / \sum \text{frac}(e_p) \tag{6}$$

where $\text{frac}(e_p)$ is the fraction part of the $e_p$ value, and $\sum \text{frac}(e_p)$ is the sum of the fraction parts of the $e_p$ values over the entire population. The chromosomes are then randomly selected in accordance with their own probabilities of selection, $P_{\text{Select}}$, to fill the remaining slots in the mating pool.

*Crossover operator*

In general, the crossover operator transfers a portion of genetic codes between two parent chromosomes selected from the mating pool. It leads to an exploitation of the solution space by introducing variations to the parent chromosomes. Firstly, a pair of parent chromosomes is chosen from the mating pool without replacement. The probability of applying the crossover operator to these two chromosomes is called the probability of crossover, $P_{\text{Cross}}$. If the decision is not to cross the chromosomes, they will be cloned to produce a pair of offspring chromosomes, where the offspring chromosomes are identical to their parents. Otherwise, the parent chromosomes will be crossed to produce two offspring chromosomes by using the crossover operator. In this paper, a new crossover operator is proposed, and a simple example is used to illustrate its operation. In order to facilitate the presentation, the pairs of parent and offspring chromosomes are identified as (***S1***, ***S2***) and (***C1***, ***C2***) respectively. Consider a pair of parent chromosomes (***S1***, ***S2***) shown below:

$$\boldsymbol{S1} = \begin{array}{c:cc:} 1 & 4 & 7 \\ \hdashline 8 & 2 & 5 \\ 3 & 6 & 9 \\ \hdashline \end{array} \qquad \boldsymbol{S2} = \begin{array}{c:cc:} 4 & 1 & 6 \\ \hdashline 2 & 5 & 7 \\ 9 & 3 & 8 \\ \hdashline \end{array}$$

Firstly, a cutting section is chosen at random. The genes bounded within the cutting section, i.e., (2, 5, 6, 9) in ***S1***, and (5, 7, 3, 8) in ***S2***, are exchanged, so that a portion of genetic codes from ***S1*** is transferred to ***S2***, and vice versa. The structures of the resultant chromosomes then become:

$$\boldsymbol{S1'} = \begin{array}{c:cc:} 1 & 4 & 7 \\ \hdashline 8 & 5 & 7 \\ 3 & 3 & 8 \\ \hdashline \end{array} \qquad \boldsymbol{S2'} = \begin{array}{c:cc:} 4 & 1 & 6 \\ \hdashline 2 & 2 & 5 \\ 9 & 6 & 9 \\ \hdashline \end{array}$$

At this stage, several genes are found to exist in more than one position in the resultant chromosomes (e.g., 3, 7, and 8 in ***S1'*** and 2, 6 and 9 in ***S2'***). These genes are termed as repeated genes in the following discussion. Indeed, each pair of repeated genes indicates that the machine represented by the value of that pair of genes has been allocated in two different machinery locations in the layout plan. In this connection, modification of the layout plan is necessary before it can be accepted. Since ***S1'*** is produced by changing the genes in ***S1*** from 2 to 5, 5 to 7, 6 to 3, and 9 to 8, a backward replacement procedure can be implemented to change the values of those repeated genes outside the cutting section from 3 to 6, 7 to 2 (the combined result of changing from 7 to 5 and then 5 to 2), and 8 to 9. Similarly, the repeated genes outside the cutting section in ***S2'*** can also be replaced by changing 2 to 7, 6 to 3, and 9 to 8. Thus the offspring chromosomes become:

$$C1 = \begin{array}{c|cc|} 1 & 4 & 2 \\ \hline 9 & 5 & 7 \\ 6 & 3 & 8 \\ \hline \end{array} \qquad C2 = \begin{array}{c|cc|} 4 & 1 & 3 \\ \hline 7 & 2 & 5 \\ 8 & 6 & 9 \\ \hline \end{array}$$

*Mutation operator*

The mutation operator is used to safeguard the search process from premature convergence to a local optima. It is an immediate operation that follows the crossover operation. It attempts to rearrange the structure of a chromosome at random. The probability of mutating a single gene is called the probability of mutation, $P_{\text{Mutate}}$, which is usually a small number. For each gene in a chromosome, an arbitrary choice is made to decide whether the mutation operation is performed or not. If the decision is not to perform the mutation operation, the gene will be kept unchanged. Otherwise, the gene is mutated by swapping its contents randomly to the other gene, on the condition that neither of the genes' contents is equal to $-1$. If the content of a gene is equal to $-1$, it means that the gene is representing a restricted area, and no mutation should be allowed. The mutation operation is then applied to the next gene, and the entire process is repeated, until all genes in the chromosome are tried.

## Numerical examples

*Example from Chan and Tansri (1994)*

The effectiveness of the proposed approach can be conveniently illustrated by using numerical examples. The first example is taken from Chan and Tansri[22], and the system specifications are presented in *Tables 4* and *5*. The plant configuration layout is a 3 by 3 grid. In order to obtain a robust design, a certain percentage of chromosomes with high fitness values will be retained as the chromosomes for the next

**Table 4 Flow of materials between machines (number of trips per period)**

| 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 |

**Table 5 Material handling cost between machines ($ per trip)**

| 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 |

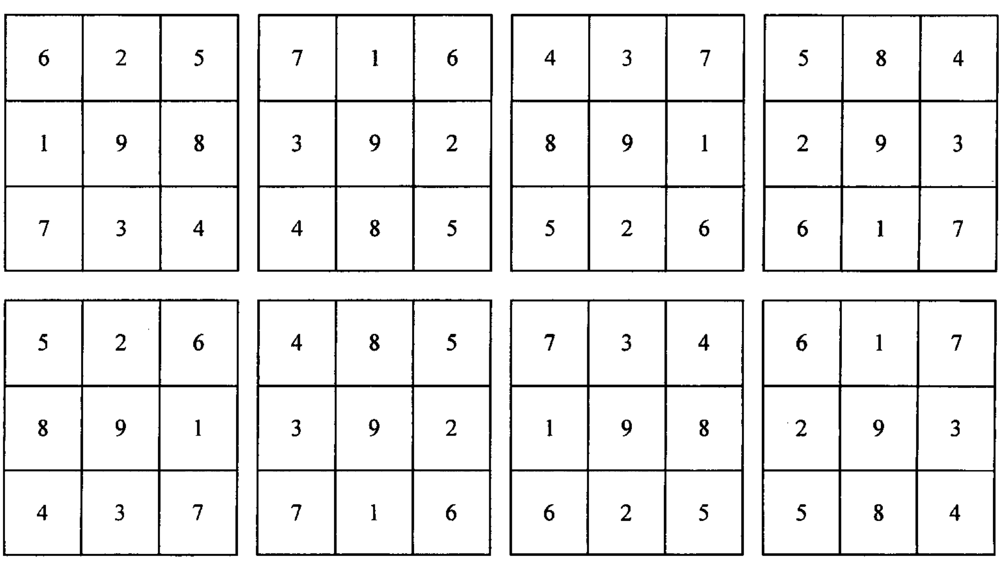


**Figure 3** Optimal facility layouts for the Chan and Tansri [22] example.

generation. The general guideline proposed by Chan and Tansri[22] is adopted to determine the values of the five genetic parameters. 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, $P_{\text{Cross}}$, and the probability of mutation, $P_{\text{Mutate}}$.

Nineteen sets of experiments are conducted to determine an appropriate combination of the population size, $P$, and the generation size, $G$. In order to evaluate the performance of the search processes in connection with the different combinations of the population and generation sizes, the experimental results are compared with the global optimal solutions. The exhaustive search method is applied to determine the global optimal solutions, and eight optimal machine layouts are obtained. *Figure 3* shows the optimal machine layouts, and the corresponding material handling cost is $4818 (this result is different from that of Chan and Tansri[22], which is probably due to some printing errors in their paper). In addition, three widely used crossover operators are also included in these experiments. They are the partially mapped crossover (PMX)[26], the order crossover (OX)[27], and the cycle crossover (CX)[28]. These three crossover operators are originally proposed to solve travelling salesman problems. Recently, Chan and Tansri[22] have attempted to use these operators to solve facility layout problems, and have reported that the PMX operator did provide excellent results. It is therefore appropriate to include these three operators in the experiments, in order to verify the effectiveness of the proposed crossover operator. To ensure fairness, the same set of selection and mutation operators are used to evaluate the performance of the crossover operators.

**Table 6 The experimental settings for different combinations of the population and generation sizes**

| Expt. | *P* | *G* | No. of trials | % of exploration |
|---|---|---|---|---|
| 1 | 20 | 10 | 200 | 0.0551 |
| 2 | 40 | 10 | 400 | 0.1102 |
| 3 | 100 | 10 | 1000 | 0.2756 |
| 4 | 200 | 10 | 2000 | 0.5511 |
| 5 | 500 | 10 | 5000 | 1.3779 |
| 6 | 20 | 20 | 400 | 0.1102 |
| 7 | 40 | 20 | 800 | 0.2205 |
| 8 | 100 | 20 | 2000 | 0.5511 |
| 9 | 200 | 20 | 4000 | 1.1023 |
| 10 | 20 | 40 | 800 | 0.2205 |
| 11 | 40 | 40 | 1600 | 0.4409 |
| 12 | 100 | 40 | 4000 | 1.1023 |
| 13 | 200 | 40 | 8000 | 2.2046 |
| 14 | 20 | 100 | 2000 | 0.5511 |
| 15 | 40 | 100 | 4000 | 1.1023 |
| 16 | 100 | 100 | 10000 | 2.7557 |
| 17 | 20 | 200 | 4000 | 1.1023 |
| 18 | 40 | 200 | 8000 | 2.2046 |
| 19 | 10 | 500 | 5000 | 1.3779 |

$R = 0.5\%$, $P_{\text{Cross}} = 0.6$ and $P_{\text{Mutate}} = 0.001$.

In general, an increase in the sizes of population and generation can produce better solutions, since the number of sampling solutions from the solution space is enlarged. However, the computational effort in searching the space will also increase, which is contradictory to the original objective of using genetic algorithms 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. In this example, there are 362 880 (9!) possible solutions in the solution space, and thus the maximum number of evaluations should be less than 10 886. *Table 6* lists the suggested combinations of the population and generation sizes. Each experiment is run 10 times with the genetic parameters of $R = 5\%$,

**Table 7 The experimental results for different combinations of the population and generation sizes**

| | Proposed crossover | | | PMX | | | OX | | | CX | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Expt. | Best | Avg. | # | Best | Avg. | # | Best | Avg. | # | Best | Avg. | # |
| 1 | 5233 | 5504.4 | 0 | 4938 | 5434.8 | 0 | 5307 | 5480.9 | 0 | 5026 | 5622.2 | 0 |
| 2 | 5040 | 5286.7 | 0 | 5039 | 5263.8 | 0 | 5032 | 5266.5 | 0 | 5133 | 5409.8 | 0 |
| 3 | 4818 | 5024.8 | 1 | 4938 | 5164.9 | 0 | 4992 | 5216.1 | 0 | 5020 | 5201.7 | 0 |
| 4 | 4818 | 4891.4 | 2 | 4818 | 4966.8 | 2 | 4818 | 5046.5 | 1 | 4818 | 5086.9 | 1 |
| 5 | 4818 | 4833.2 | 7 | 4818 | 4892.3 | 5 | 4818 | 4947.7 | 1 | 4818 | 4911.4 | 1 |
| 6 | 5225 | 5481.2 | 0 | 4938 | 5402.1 | 0 | 5307 | 5480.9 | 0 | 5026 | 5589.6 | 0 |
| 7 | 4927 | 5174.6 | 0 | 4992 | 5184.6 | 0 | 5032 | 5198.2 | 0 | 5133 | 5402.1 | 0 |
| 8 | 4818 | 4889.1 | 4 | 4818 | 4991.7 | 2 | 4927 | 5125.8 | 0 | 4818 | 5145.2 | 1 |
| 9 | 4818 | 4846.5 | 5 | 4818 | 4919.8 | 2 | 4818 | 4960.7 | 1 | 4818 | 4973.8 | 3 |
| 10 | 5225 | 5462.2 | 0 | 4938 | 5402.1 | 0 | 5307 | 5471.8 | 0 | 5026 | 5566.5 | 0 |
| 11 | 4927 | 5163.8 | 0 | 4992 | 5180.7 | 0 | 5014 | 5182.0 | 0 | 5133 | 5378.7 | 0 |
| 12 | 4818 | 4871.4 | 4 | 4818 | 4919.5 | 3 | 4862 | 5006.7 | 0 | 4818 | 5090.5 | 1 |
| 13 | 4818 | 4840.0 | 5 | 4818 | 4887.9 | 4 | 4818 | 4894.7 | 4 | 4818 | 4902.9 | 4 |
| 14 | 5225 | 5453.0 | 0 | 4938 | 5337.0 | 0 | 5020 | 5326.9 | 0 | 5026 | 5471.0 | 0 |
| 15 | 4818 | 5141.6 | 1 | 4927 | 5122.4 | 0 | 4872 | 5156.2 | 0 | 5039 | 5304.5 | 0 |
| 16 | 4818 | 4866.0 | 5 | 4818 | 4863.9 | 4 | 4818 | 4912.5 | 2 | 4818 | 4988.9 | 2 |
| 17 | 4818 | 5303.9 | 1 | 4938 | 5224.6 | 0 | 4862 | 5229.4 | 0 | 4862 | 5266.3 | 0 |
| 18 | 4818 | 5141.4 | 1 | 4862 | 5088.4 | 0 | 4818 | 5061.3 | 1 | 4862 | 5129.9 | 0 |
| 19 | 4818 | 5184.3 | 1 | 4818 | 5166.1 | 1 | 4818 | 5115.5 | 1 | 4862 | 5272.0 | 0 |
| Total | | | 37 | | | 23 | | | 11 | | | 13 |

Best = The material handling cost of the best solution among the 10 runs. Avg. = The average of the best material handling costs among the 10 runs. # = No. of runs which yielded one of the eight optimal solutions. The optimal material handling cost is $4818.

**Table 8** The experimental settings for studying the effects of changing $R$, $P_{Cross}$, and $P_{Mutate}$

| Expt. | $R$ (%) | Expt. | $P_{Cross}$ | Expt. | $P_{Mutate}$ |
|---|---|---|---|---|---|
| 20 | 0 | 27 | 0.5 | 33 | 0.000 |
| 21 | 2 | 28 | 0.6 | 34 | 0.001 |
| 22 | 4 | 29 | 0.7 | 35 | 0.003 |
| 23 | 5 | 30 | 0.8 | 36 | 0.005 |
| 24 | 6 | 31 | 0.9 | 37 | 0.010 |
| 25 | 8 | 32 | 1.0 | 38 | 0.030 |
| 26 | 10 | | | 39 | 0.050 |

$P_{Cross} = 0.6$, and $P_{Mutate} = 0.001$. The experimental results (*Table 7*) are expressed in terms of:

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

In *Table 7*, the number of successful runs required to obtain one of the eight optimal solutions among

**Table 9** The experimental results for studying the effect of changing $R$

| | Proposed crossover | | | PMX | | | OX | | | CX | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Expt. | Best | Avg. | # | Best | Avg. | # | Best | Avg. | # | Best | Avg. | # |
| 20 | 4818 | 5043.0 | 2 | 5065 | 5218.7 | 0 | 5065 | 5218.7 | 0 | 4978 | 5122.9 | 0 |
| 21 | 4818 | 4946.1 | 4 | 4872 | 5083.1 | 0 | 4872 | 5083.1 | 0 | 4818 | 5033.2 | 1 |
| 22 | 4818 | 4856.0 | 4 | 4862 | 5059.5 | 0 | 4862 | 5059.5 | 0 | 4818 | 5092.5 | 1 |
| 23 | 4818 | 4889.1 | 4 | 4818 | 4991.7 | 2 | 4927 | 5125.8 | 0 | 4818 | 5145.2 | 1 |
| 24 | 4818 | 4936.9 | 4 | 4818 | 5048.6 | 1 | 4818 | 5048.6 | 1 | 4872 | 5165.4 | 0 |
| 25 | 4818 | 4962.7 | 1 | 4938 | 5080.4 | 0 | 4938 | 5080.4 | 0 | 4938 | 5177.8 | 0 |
| 26 | 4818 | 4943.6 | 3 | 4818 | 5052.6 | 1 | 4818 | 5052.6 | 1 | 4938 | 5133.8 | 0 |
| Total | | | 22 | | | 4 | | | 2 | | | 3 |

# = No. of runs which yielded one of the eight optimal solutions. The optimal material handling cost is $4818. The genetic parameters are $P = 100$, $G = 20$, $P_{Cross} = 0.6$ and $P_{Mutate} = 0.001$.

**Table 10** The experimental results for studying the effect of changing $P_{Cross}$

| | Proposed crossover | | | PMX | | | OX | | | CX | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Expt. | Best | Avg. | # | Best | Avg. | # | Best | Avg. | # | Best | Avg. | # |
| 27 | 4818 | 4970.2 | 2 | 4818 | 5094.8 | 1 | 4927 | 5126.9 | 0 | 5039 | 5189.5 | 0 |
| 28 | 4818 | 4856.0 | 4 | 4818 | 4991.7 | 2 | 4818 | 5048.6 | 1 | 4818 | 5033.2 | 1 |
| 29 | 4818 | 4898.7 | 1 | 4818 | 5011.3 | 2 | 4818 | 5061.9 | 1 | 4818 | 5026.3 | 3 |
| 30 | 4818 | 4905.6 | 2 | 4818 | 4979.3 | 2 | 4872 | 5070.3 | 0 | 4862 | 5061.2 | 0 |
| 31 | 4818 | 4942.7 | 5 | 4818 | 4992.4 | 2 | 4862 | 5030.9 | 0 | 4818 | 4993.5 | 3 |
| 32 | 4818 | 4929.0 | 1 | 4818 | 4983.5 | 1 | 4818 | 5014.8 | 1 | 4862 | 5003.5 | 0 |
| Total | | | 15 | | | 10 | | | 3 | | | 7 |

# = No. of runs which yielded one of the eight optimal solutions. The optimal material handling cost is $4818. The values of the genetic parameters for the different crossover operators are: the proposed crossover operator: $P = 100$, $G = 20$, $R = 4\%$ and $P_{Mutate} = 0.001$; the PMX operator: $P = 100$, $G = 20$, $R = 5\%$ and $P_{Mutate} = 0.001$; the OX operator: $P = 100$, $G = 20$, $R = 6\%$ and $P_{Mutate} = 0.001$; the CX operator: $P = 100$, $G = 20$, $R = 2\%$ and $P_{Mutate} = 0.001$.

**Table 11** The experimental results for studying the effect of changing $P_{Mutate}$

| | Proposed crossover | | | PMX | | | OX | | | CX | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Expt. | Best | Avg. | # | Best | Avg. | # | Best | Avg. | # | Best | Avg. | # |
| 33 | 4818 | 4887.8 | 3 | 4818 | 4983.9 | 3 | 4862 | 5086.7 | 0 | 4818 | 5087.1 | 1 |
| 34 | 4818 | 4856.0 | 4 | 4818 | 4979.3 | 2 | 4818 | 5014.8 | 1 | 4818 | 4993.5 | 3 |
| 35 | 4818 | 4961.6 | 2 | 4818 | 4951.6 | 1 | 4818 | 5059.0 | 1 | 4818 | 5012.2 | 1 |
| 36 | 4818 | 4887.8 | 2 | 4862 | 5013.7 | 0 | 4872 | 5117.4 | 0 | 4818 | 4994.1 | 2 |
| 37 | 4818 | 4881.2 | 2 | 4818 | 4987.5 | 1 | 4862 | 5073.8 | 0 | 4818 | 4993.3 | 1 |
| 38 | 4818 | 4895.8 | 4 | 4818 | 4985.7 | 1 | 4818 | 5014.9 | 1 | 4818 | 4986.9 | 3 |
| 39 | 4818 | 4944.6 | 2 | 4862 | 5029.3 | 0 | 4872 | 5010.6 | 0 | 4818 | 4955.7 | 1 |
| Total | | | 19 | | | 8 | | | 3 | | | 12 |

# = No. of runs which yielded one of the eight optimal solutions. The optimal material handling cost is $4818. The values of the genetic parameters for the different crossover operators are: the proposed crossover operator: $P = 100$, $G = 20$, $R = 4\%$ and $P_{Cross} = 0.6$; the PMX operator: $P = 100$, $G = 20$, $R = 5\%$ and $P_{Cross} = 0.8$; the OX operator: $P = 100$, $G = 20$, $R = 6\%$ and $P_{Cross} = 1.0$; the CX operator: $P = 100$, $G = 20$, $R = 2\%$ and $P_{Cross} = 0.9$.

the operators PMX, OX, and CX are 23, 11, and 13, respectively. The results agree with those provided by Chan and Tansri[22], except for the OX operator. Indeed, the results correlate with the original expectations, since the PMX operator performs better than the OX and CX operators in solving facility layout problems. However, the proposed crossover with 37 successful runs performs even better than the PMX operator. In addition, the proposed crossover operator also works well in a number of combinations of the population and generation sizes, including ($P = 500, G = 10$), ($P = 100, G = 20$), and ($P = 200$, $G = 20$), etc. The PMX, OX, and CX operators provide good results only in a few cases. This preliminary result shows that the proposed crossover operator is more robust than the PMX, OX, and CX operators in solving facility layout problems.

**Table 12** The optimal values of the genetic parameters

| | Proposed crossover | PMX | OX | CX |
|---|---|---|---|---|
| $P$ | 200 | 200 | 200 | 200 |
| $G$ | 40 | 40 | 40 | 40 |
| $R$ | 4 | 5 | 6 | 2 |
| $P_{\mathrm{Cross}}$ | 0.6 | 0.8 | 1.0 | 0.9 |
| $P_{\mathrm{Mutate}}$ | 0.001 | 0.001 | 0.001 | 0.030 |

*Sensitivity analysis of the R values*

Besides the population and generation sizes, the solution quality can also be affected by the percentage of replication of well-performed chromosomes in each generation, $R$, the probability of crossover, $P_{\mathrm{Cross}}$, and the probability of mutation, $P_{\mathrm{Mutate}}$. The effects of these genetic parameters on the solution quality are studied in accordance with the experimental settings depicted in *Table 8*. In conducting experiments (20–26), each experiment is run 10 times with the genetic parameters $P = 100$, $G = 20$, $P_{\mathrm{Cross}} = 0.6$, and $P_{\mathrm{Mutate}} = 0.001$. *Table 9* presents the experimental results. The results indicate that the proposed crossover operator outperforms the PMX, OX, and CX operators in terms of the number of successful runs. In addition, the proposed crossover operator works well in a wide range of $R$, which implies that it is insensitive to the change of $R$. Once

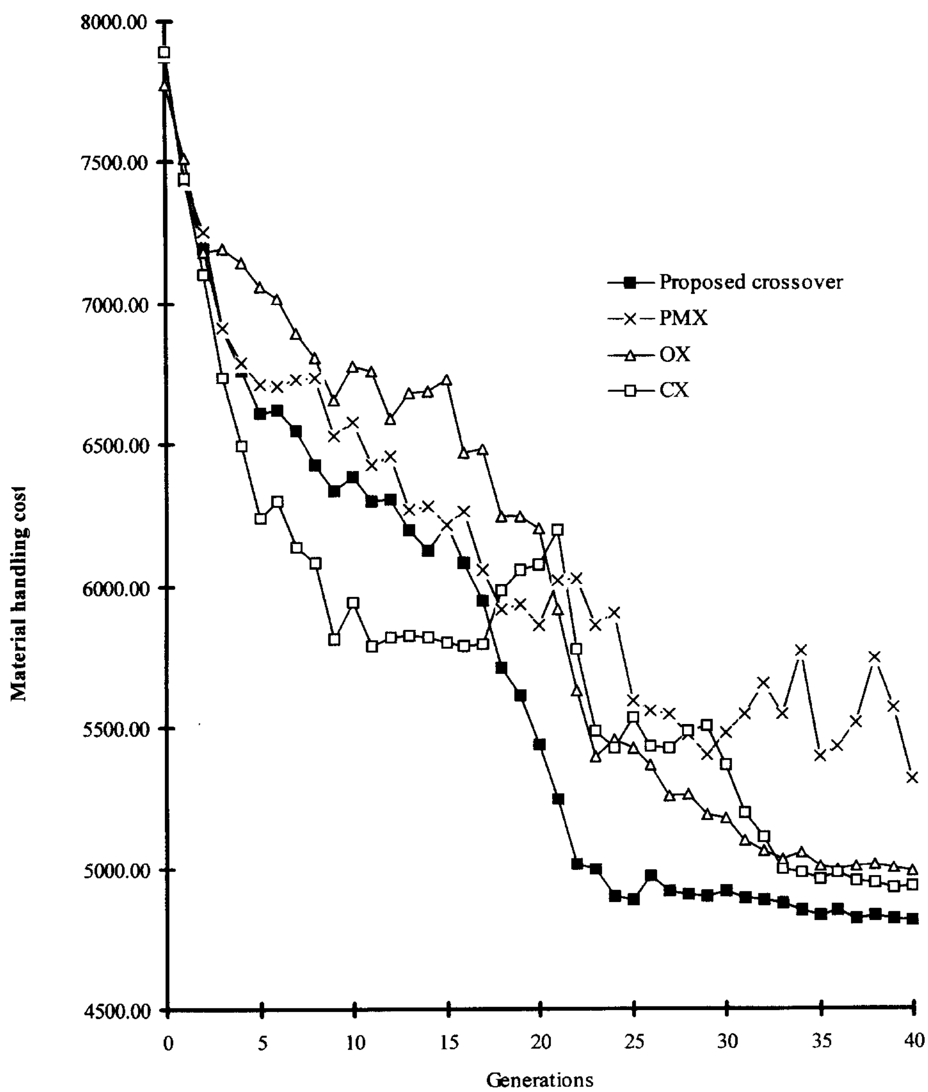


**Figure 4** Results of the generation-average total cost.

again, it shows that the proposed crossover operator is quite vigorous.

*Sensitivity analysis of the $P_{cross}$ values*

The best values of $R$ for each crossover operator are then used to study the effect of changing $P_{\mathrm{Cross}}$ on the solution quality. The selected values of $R$ for the proposed crossover operator, and for the PMX, OX and CX operators are 4%, 5%, 6% and 2% respectively. Each of the experiments (27–32) is run 10 times, with the genetic parameters $P = 100$, $G = 20$ and $P_{\mathrm{Mutate}} = 0.001$. The experimental results in *Table 10* show that the proposed crossover operator works well at probability 0.6 and 0.9. However, it seems that the proposed crossover operator works better at probability 0.6, since its corresponding average value ($4856.0) is less than that obtained at probability 0.9 ($4942.7). The best values of the probability of crossover for the PMX, OX, and CX operators are 0.8, 1.0, and 0.9, respectively. These results are slightly different, when compared with Chan and Tansri[22], which is probably due to the difference in the selection operation. Both the proposed crossover operator and the PMX operator have successfully located the global optimal solution in all experiments. However, the success rate of the proposed crossover operator is higher than that of the PMX operator, which reveals that the proposed crossover operator is more reliable.

*Sensitivity analysis of the $P_{mutate}$ values*

The probabilities of crossover, $P_{\mathrm{Cross}}$, for the proposed crossover operator, and for the PMX, OX, and CX operators are then fixed at 0.6, 0.8, 1.0, and 0.9, respectively. When joined with their respective values of $R$ fixed at 4%, 5%, 6%, and 2%, the effect of changing $P_{\mathrm{Mutate}}$ on the solution quality is studied. Experiments (33–39) are conducted in such a way that each experiment is run 10 times with the genetic parameters $P = 100$ and $G = 20$. *Table 11* lists the results of the experiment, and again the proposed crossover operator outperforms the PMX, OX, and CX operators. Although it seems that the PMX operator has a greater number of successful runs at probability 0.000, the best value of the probability of

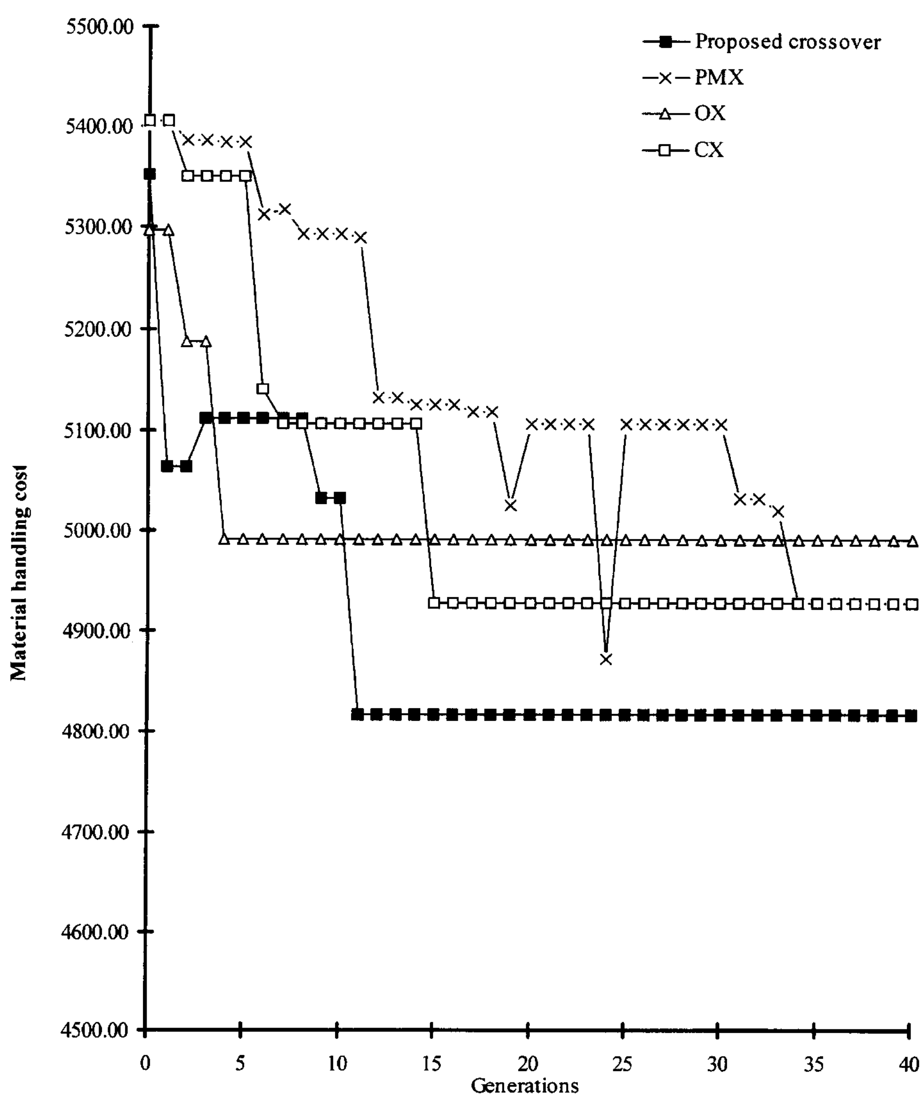


**Figure 5** Results of the best-of-generation total cost.

mutation should be 0.001, since its corresponding average value ($4979.3) is less than that obtained at probability 0.000 ($4983.9). The best values of the probability of mutation for the proposed crossover operator, PMX, OX, and CX are 0.001, 0.001, 0.001, and 0.030, respectively. The proposed crossover and the CX operators can also locate the global optimal solution in all the experiments. Indeed, the CX operator works surprisingly well with the different values of $P_{\mathrm{Mutate}}$. The CX operator is shown to outperform even the PMX operator, if its parameter settings are tuned carefully. However, this also becomes the CX operator's drawback, because this implies that its performance is sensitive to the parameter settings.

*Cost performance of the various crossover operators in a search process*

**Table 13 Part list and production data**

| | P1 | P2 | P3 | P4 | P5 |
|---|---|---|---|---|---|
| Production routing | 22-1-13-21 | 3-20-24 | 14-7-23-24 | 15-6-18-8-12 | 15-6-18-8-12-5 |
| Unit op. proc. time* | 2-3-4-1 | 1-1-2 | 4-4-3-4 | 1-3-3-2-4 | 1-1-3-3-2-3 |
| Production volume | 130 | 150 | 125 | 145 | 65 |
| | **P6** | **P7** | **P8** | **P9** | **P10** |
| Production routing | 9-17-10 | 9-17-10 | 4-16 | 22-1-13-21 | 2-11-19-5-21 |
| Unit op. proc. time* | 2-2-1 | 2-1-1 | 1-1 | 4-2-4-4 | 2-3-3-3-1 |
| Production volume | 78 | 95 | 160 | 85 | 105 |
| | **P11** | **P12** | **P13** | **P14** | **P15** |
| Production routing | 3-20 | 3-20 | 2-11-19 | 2-11-19-5 | 3-20 |
| Unit op. proc. time* | 1-3 | 1-1 | 1-1-2 | 3-4-3-4 | 3-3 |
| Production volume | 130 | 140 | 150 | 185 | 78 |
| | **P16** | **P17** | **P18** | **P19** | **P20** |
| Production routing | 22-1-13-21 | 1-13-22 | 15-6-18-8-12 | 4-16 | 10-17-12 |
| Unit op. proc. time* | 1-3-2-2 | 2-2-3 | 2-1-4-3-2 | 1-4 | 2-3-4 |
| Production volume | 95 | 160 | 85 | 105 | 130 |
| | **P21** | **P22** | **P23** | **P24** | **P25** |
| Production routing | 4-16 | 2-5-11-19 | 3 | 20-12 | 7-14-23 |
| Unit op. proc. time* | 3-4 | 3-3-2-3 | 2 | 2-4 | 4-1-2 |
| Production volume | 105 | 130 | 140 | 150 | 185 |
| | **P26** | **P27** | **P28** | **P29** | **P30** |
| Production routing | 15-6-18-8-10 | 15-6-18-8-12 | 4 | 9-17 | 6-18-8-12 |
| Unit op. proc. time* | 4-2-2-2-1 | 3-3-3-2-3 | 2 | 3-1 | 2-4-2-1 |
| Production volume | 145 | 65 | 78 | 95 | 160 |
| | **P31** | **P32** | **P33** | **P34** | **P35** |
| Production routing | 3-20-17 | 14-7-23-24-16 | 22-1-13-21-2 | 3-20 | 11-19-5 |
| Unit op. proc. time* | 2-4-2 | 4-4-4-2-2 | 2-3-4-2-4 | 1-4 | 1-3-2 |
| Production volume | 85 | 105 | 130 | 150 | 125 |
| | **P36** | **P37** | **P38** | **P39** | **P40** |
| Production routing | 20-12-21 | 16-11-14 | 4-16 | 4-16 | 1-13-19 |
| Unit op. proc. time* | 4-2-2 | 3-1-4 | 1-1 | 1-2 | 1-2-4 |
| Production volume | 145 | 65 | 78 | 95 | 160 |

*It is assumed that the unit operation processing time for each operation is the same in the first study. The figures listed here are used in the second study only.

**Table 14 Results of solving the facility layout problem**

| Crossover operator | Best (30 runs) | Avg. (30 runs) | Worst (30 runs) | Successful hits |
|---|---|---|---|---|
| The proposed crossover | 12982 | 15087.7 | 18657 | 11 |
| PMX | 14947 | 18355.9 | 20654 | 0 |
| OX | 22406 | 24301.7 | 26926 | 0 |
| CX | 14717 | 17216.5 | 20654 | 0 |

The total material handling cost of the best solution obtained among all the runs is $12982. Successful hits: the number of runs that have hit the best solution obtained.

The general cost performance of the various crossover operators in a search process is studied with the respective optimal values of the genetic parameters shown in *Table 12*, and the results are presented in *Figures 4* and *5*. In *Figure 4*, the generation-average total cost is plotted against the generation number, whereas *Figure 5* shows the behaviour of the corresponding best-of-generation total cost. Both the total costs converge very rapidly in just a few generations for the OX and CX operators. Indeed, these two operators have resulted in a fairly smooth cost performance. In particular, there are no sudden changes in the best-of-generation total cost when the search processes start to converge. However, it is important to note that such an early premature convergence may lead to a local optimal solution. The cost performance of the PMX operator, on the other hand, is less smooth. Some sudden changes in the best-of-generation total cost are detected as the generation advances. These changes, however, provide a means to enable the total cost to escape from being trapped in a local optimum, and to progress to become the global optimal solution eventually. However, the search process is time-consuming, in this case, because its convergence is the slowest. *Figures 4* and *5* show that the total costs still have not reached their corresponding steady state after 40 generations. In fact, this operator is expected to perform better if the search process is allowed to continue with more generations. The proposed crossover operator exhibits similar behaviour in the cost performance as the PMX operator. *Figures 4* and *5* reveal that the changes in the total costs begin to diminish after 30 generations. According to *Figure 5*, the best-of-generation total cost converges rapidly in the beginning of the search, and the global optimal solution has already been located at as early as the 11th generation. Among all the crossover operators, the proposed crossover operator is the only one that can successfully locate the global optimal solution. The proposed operator is therefore proved to be effective and efficient in solving facility layout problems.

The results of the preliminary study show that the proposed crossover operator outperforms the PMX, OX, and CX operators in solving facility layout problems. In addition, the results of the study listed in *Table 12* also suggest the appropriate values of the genetic parameters for the proposed crossover, and also the PMX, OX, and CX operators.

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

(a) No constraint imposed (total material handling cost = $12982).

| | | | | | |
|---|---|---|---|---|---|
| 4 | 22 | 1 | 13 | 26 | 28 |
| 16 | 9 | 3 | 21 | 5 | 2 |
| 15 | 24 | 20 | 17 | 19 | 11 |
| 6 | 18 | 8 | 25 | 12 | 14 |
| * | 27 | 10 | 29 | 23 | 7 |

(b) Under the FCFS policy (total material handling cost = $15545).

| | | | | | |
|---|---|---|---|---|---|
| 4 | 16 | 15 | 26 | 1 | 22 |
| 27 | 18 | 6 | 21 | 13 | 2 |
| 25 | 8 | 12 | 23 | 19 | 11 |
| 9 | 10 | 17 | 14 | 5 | 28 |
| * | 24 | 20 | 7 | 3 | 29 |

(c) Under the SPTF policy (total material handling cost = $15709)

| | | | | | |
|---|---|---|---|---|---|
| 4 | 16 | 14 | 11 | 2 | 28 |
| 24 | 23 | 7 | 19 | 5 | 25 |
| 3 | 20 | 12 | 21 | 8 | 27 |
| 29 | 17 | 22 | 10 | 18 | 6 |
| * | 9 | 1 | 13 | 26 | 15 |

(d) Under the LPTF policy (total material handling cost = $14946)

**Figure 6** The best facility layouts obtained under different machine loading policies for the Kazerooni et al. [20] example.

*Example from Kazerooni et al.* (1996)

The proposed approach is again applied to solve another facility layout problem excerpted from Kazerooni *et al.*[20]. If the proposed approach is testified to be sturdy, the values of the genetic parameters obtained in the previous study can still provide good results in solving this facility layout problem. In this paper, the best routing configuration determined by Kazerooni *et al.*[20] is used as an input for the proposed approach. Since the unit operation processing time and the effective capacities of machines are not provided in Kazerooni *et al.*[20], it is assumed that the unit operation processing time for each operation is the same, and there is no capacity constraint for the machines. In addition, the material handling costs are also assumed to be the same among machines. *Table 13* presents the part list and the corresponding production data of the parts. Since the problem needs to locate 24 types of machines ($m = 1,2,\ldots,24$) in a 5 by 6 machinery location grid, there are $2.65 \times 10^{32}$ (30!) possible solutions in the solution space. The determination of the global optimal solution by using the exhaustive search method is impossible in this case. Hence, the proposed approach is applied to solve the problem with the genetic parameters $P = 200$, $G = 40$, $R = 4\%$, $P_{\text{Cross}} = 0.6$, and $P_{\text{Mutate}} = 0.001$. Since there is no capacity constraint for the machines, the total material handling cost is, therefore, not affected by the machine loading policy. For the sake of comparison, the three widely used crossover operators, namely, the PMX, OX, and CX operators, are also applied to solve the problem with the respective optimal values of the genetic parameters shown in *Table 12*. Thirty runs of genetic search are conducted for each crossover operator, and the results are shown in *Table 14*.

*Table 14* shows that the proposed crossover operator again outperforms all the three crossover operators. As a matter of fact, the results show that during the 30 runs, the PMX, OX and CX operators cannot even hit the best solution obtained from the proposed approach. By using the optimal values of the genetic parameters, the CX operator once again obtains a better solution than those obtained by the PMX and OX operators. Among the 30 runs, the proposed crossover operator is observed to be able to hit the best solution obtained 11 times successfully. This result is reasonable, because the proposed crossover operator is believed to be less sensitive to the values of genetic parameters when compared to the other three crossover operators. In this connection, the proposed crossover operator can still have a good chance to locate a good solution, even though the values of the genetic parameters are not specifically tuned for this problem. The best facility layout obtained by using the proposed crossover operator is presented in *Figure 6*(a).

The problem is now re-visited when there are capacity constraints for the machines. In this case, the

**Table 15 Effective capacities of the different types of machines**

| Machine types | Effective capacities |
|---|---|
| 1 | 1800 |
| 2 | 2000 |
| 3 | 1800 |
| 4 | 1400 |
| 5 | 2000 |
| 6 | 1400 |
| 7 | 1400 |
| 8 | 1600 |
| 9 | 1000 |
| 10 | 1000 |
| 11 | 1800 |
| 12 | 2000 |
| 13 | 1800 |
| 14 | 1800 |
| 15 | 1400 |
| 16 | 1800 |
| 17 | 1200 |
| 18 | 1400 |
| 19 | 1800 |
| 20 | 2000 |
| 21 | 1800 |
| 22 | 1800 |
| 23 | 1400 |
| 24 | 1200 |

**Table 16 Results of solving the facility layout problem with imposed constraints under various machine loading policies**

| Crossover operator | Best (30 runs) | Avg. (30 runs) | Worst (30 runs) | Successful hits |
|---|---|---|---|---|
| FCFS | | | | |
| The proposed crossover | 15545 | 16350.3 | 18432 | 13 |
| PMX | 16607 | 19763.4 | 24037 | 0 |
| OX | 23807 | 25165.1 | 26337 | 0 |
| CX | 15638 | 18136.9 | 19642 | 0 |
| SPTF | | | | |
| The proposed crossover | 15709 | 16757.9 | 18762 | 12 |
| PMX | 17092 | 20621.4 | 26337 | 0 |
| OX | 23285 | 24807.4 | 26711 | 0 |
| CX | 16661 | 18789.3 | 21624 | 0 |
| LPTF | | | | |
| The proposed crossover | 14946 | 15799.3 | 17603 | 16 |
| PMX | 15283 | 17269.9 | 19622 | 0 |
| OX | 24758 | 26130.5 | 27249 | 0 |
| CX | 15286 | 17839.4 | 19762 | 0 |

The total material handling costs of the best solutions obtained under the machine loading policies of FCFS, SPTF and LPTF are \$15545, \$15709 and \$14946, respectively. Successful hits: the number of runs that have hit the best solution obtained.

**Table 17 The production plan of the best solution obtained under the machine loading policy of LPTF**

| | Production routing | Production volume | | Production routing | Production volume |
|---|---|---|---|---|---|
| P1 | 22-1-13-21 | 130 | P21 | 4-16 | 105 |
| P2 | 3-20-24 | 150 | P22 | 2-5-11-19 | 101 |
| P3 | 14-7-23-24 | 125 | | 2-5-11-28 | 29 |
| P4 | 15-6-18-8-12 | 145 | P23 | 3 | 140 |
| P5 | 15-6-27-8-25-5 | 65 | P24 | 20-12 | 150 |
| P6 | 9-17-10 | 78 | P25 | 7-14-23 | 185 |
| P7 | 9-17-10 | 95 | P26 | 15-6-18-8-10 | 145 |
| P8 | 4-16 | 160 | P27 | 15-6-27-8-25 | 65 |
| P9 | 22-1-26-21 | 85 | P28 | 4 | 78 |
| P10 | 2-11-28-5-21 | 105 | P29 | 9-17 | 95 |
| P11 | 3-20 | 123 | P30 | 6-18-8-12 | 160 |
| | 3-29 | 7 | P31 | 3-29-17 | 85 |
| P12 | 3-29 | 140 | P32 | 14-7-23-24-16 | 105 |
| P13 | 2-11-19 | 150 | P33 | 22-1-13-21-2 | 130 |
| P14 | 2-11-19-5 | 185 | P34 | 3-20 | 150 |
| P15 | 3-29 | 78 | P35 | 11-28-5 | 125 |
| P16 | 22-1-13-21 | 60 | P36 | 20-12-21 | 145 |
| | 22-1-26-21 | 35 | P37 | 16-11-14 | 65 |
| P17 | 1-13-22 | 160 | P38 | 4-16 | 78 |
| P18 | 15-6-18-8-25 | 8 | P39 | 4-16 | 95 |
| | 15-6-27-8-25 | 77 | P40 | 1-131-19 | 160 |
| P19 | 4-16 | 105 | | | |
| P20 | 2-5-11-19 | 92 | | | |
| | 2-5-11-28 | 38 | | | |

Machines $m$ = 12,25 belong to the machine of type 12, machines $m$ = 13,26 belong to the machine of type 13, machines $m$ = 18,27 belong to the machine of type 18, machines $m$ = 19,28 belong to the machine of type 19, and machines $m$ = 20,29 belong to the machine of type 20.

unit operation processing time for each operation is generated by using random numbers as shown in *Table 13*. In addition, the effective capacities of the different types of machines are shown in *Table 15*. *Figure 6*(b–d) illustrate the plant configuration layout, where the symbol '*' represents a restricted area. Moreover, the location on the upper left corner is a reserved machinery location, where only type 4 machine is allowed to be located. By using eqn (1), the number of each type of machines is calculated as $Q_m = 1$ $(m = 1, 2, \ldots, 24)$ except $Q_{12} = Q_{13} = Q_{18} = Q_{19} = Q_{20} = 2$. Hence, a total number of 29 ($M = \sum_{m=1}^{24} Q_m$) machines are required in the system. It is also assumed that the machines listed in each of the following brackets belong to the same machine type: (12, 25), (13, 26), (18, 27), (19, 28), and (20, 29). Three machine loading policies, namely, the First-Come-First-Serve (FCFS), the Shortest Processing Time First (SPTF), and the Longest Processing Time First (LPTF), are used to derive the optimal facility layout. The effects of these machine loading polices on the total material cost are also studied. Similarly, 30 runs of genetic search are conducted for each of the four crossover operators (i.e. the proposed crossover, PMX, OX and CX operators) for each machine loading policy, and the results are shown in *Table 16*. It can be noted that the proposed crossover operator outperforms the other crossover operators in all cases. The best facility layouts obtained under different machine loading policies are illustrated in *Figure 6*(b–d). In addition, *Table 16* also reveals that the best facility layout obtained under the LPTF policy incurs the lowest total material handling cost. *Table 17* presents the corresponding production plan.

*Table 16* also shows that the proposed crossover operator is very promising in solving the facility layout problem. The number of successful hits on the best solutions obtained under the FCFS, SPTF and LPTF policies are 13, 12, and 16, respectively. Furthermore, both the average and the worst solutions are very close to the best solution obtained under each machine loading policy. The computation time required to solve the problem genetically over 40 generations is about 3.5 min on a Pentium-200 based PC. The effectiveness and efficiency of the proposed crossover operator is again reinforced.

## Conclusion

This paper has presented the use of genetic algorithms as a general methodology to solve facility layout problems. A mathematical model has been developed to examine the machines' layout and the pattern of material flow for the typical job shop and flow shop manufacturing environments, and analysis has been presented to consider various practical aspects, such as the constraints of restricted areas and reserved machinery locations, and also the irregularity of the shapes of manufacturing plants, etc. A genetic approach has been proposed to derive the optimal machine layout which minimizes the total material handling cost. A new crossover operator has been introduced, and its effectiveness has been studied by using the benchmark problem excerpted from Chan and Tansri[22]. The cost performance of the proposed crossover operator has been compared with those of the partially mapped crossover (PMX), the order crossover (OX), and the cycle crossover (CX) operators. The results have shown that the proposed crossover operator is the best operator, scoring superior results to those obtained by the PMX, CX and OX operators. The same genetic approach has also been used to solve another facility layout problem excerpted from Kazerooni *et al.*[20] with a much larger problem size. The results obtained by using the proposed crossover, PMX, OX and CX operators under three different machine loading policies have been compared. The results of the comparison have shown that the proposed approach is robust, and its success rate of hitting the best solution obtained is high. Indeed, the proposed approach provides a very effective means to solve facility layout problems.

## References


1 **Reynolds WA and Cheung MK** 'The use of industrial engineering in Hong Kong manufacturing industry', *Int. J. Prod. Res.* Vol 22 No 6 (1984) 983–999

2 **Hyer NL and Wemmerlöv U** 'Group technology in the US manufacturing industry: a survey of current practice', *Int. J. Prod. Res.* Vol 27 No 8 (1989) 1287–1304

3 **Wemmerlöv U and Hyer NL** 'Cellular manufacturing in the US industry: a survey of users', *Int. J. Prod. Res.* Vol 27 No 9 (1989) 1511–1530

4 **Garey MR and Johnson DS** *Computers and intractability: a guide to the theory of NP-completeness.* W.H. Freeman, New York, (1979).

5 **Kaku BK, Thompson GL and Morton TE** 'A hybrid heuristic for the facilities layout problem', *Computers and Operations Research* Vol 18 (1991) 241–253

6 **Koopmans TC and Beckmann M** 'Assignment problems and the location of economic activities', *Econometrica* Vol 25 (1957) 53–76

7 **Hanan M and Kurtzberg JM** 'A review of the placement and quadratic assignment problems', *SIAM Review* Vol 14 (1972) 324–342

8 **Pierce JF and Crowston WB** 'Tree search algorithms in quadratic assignment problems', *Nav. Res. Log. Quart.* Vol 18 (1971) 1–36

9 **Love RF and Wong JY** 'Solving quadratic assignment problems with rectangular distances and integer programming', *Nav. Res. Log. Quart.* Vol 23 No 4 (1976) 623–627

10 **King JR and Nakornchai V** 'Machine-component group formation in group technology: review and extension', *International Journal of Production Research* Vol 20 No 2 (1982) 117–133

11 **McAuley J** 'Machine grouping for efficient production', *The Production Engineer* Vol 51 No 2 (1972) 53–57

12 **Seifoddini H and Wolfe PM** 'Application of the similarity coefficient method in group technology', *IIE Transaction* Vol 18 No 3 (1987) 271–277

13 **Kazerooni M, Luong LHS and Abhary K** 'A genetic algorithm based cell design considering alternative routing', *International Journal of Computer Integrated Manufacturing Systems* Vol 10 .No 2 (1997) 93–107

14 **Vannelli A and Kumar KR** 'A method for finding minimal bottleneck cells for grouping part-machine families', *International Journal of Production Research* Vol 24 No 2 (1986) 387–400

15 **Boctor FF** 'A linear formation of the machine-part cell formation problem', *International Journal of Production Research* Vol 29 No 2 (1991) 343–356

16 **Hollier RH** 'The layout of multi-product lines', *International Journal of Production Research* Vol 2 No 1 (1963) 27–57

17 **Carrie A** 'The layout of multi-product lines', *International Journal of Production Research* Vol 13 No 6 (1975) 541–557

18 **Holland JH** *Adaptation in natural and artificial systems.* The University of Michigan Press, Ann Arbor, (1975).

19 **Goldberg DE** *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, Reading, (1989).

20 **Kazerooni M, Luong LHS, Abhary K, Chan FTS and Pun FKF** 'An integrated method for cell layout problem using genetic algorithms', *In: Proceedings of 12th International Conference on CAD/CAM Robotics and Factories of the Future*, August (1996), 752–762.

21 **Venugopal V and Narendran TT** 'A genetic algorithm approach to the machine-component grouping problem with multiple objectives', *Computers & Industrial Engineering* Vol 22 (1992) 469–480

22 **Chan KC and Tansri H** 'A study of genetic crossover operations on the facility layout problem', *Computers & Industrial Engineering* Vol 26 No 3 (1994) 537–550

23 **Wesolowsky GO and Love RF** 'A non-linear approximation method for solving a generalized rectangular distance weber problem', *Man. Sci.* Vol 18 (1972) 656–663

24 **DeJong KA** 'Analysis of the behavior of a class of genetic adaptive systems', *Ph.D. thesis, The University of Michigan*, (1975).

25 **Brindle A** 'Genetic algorithms for function optimization', *Ph.D. thesis, University of Alberta*, (1981).

26 **Goldberg DE and Lingle R** 'Alleles, loci, and the traveling salesman problem', *In: Proceedings of an International Conference on Genetic Algorithms and Their Application*, (1985), 154–159.

27 **Davis L** 'Applying adaptive algorithms to epistatic domains', *In: Proceedings of the International Joint Conference on Artificial Intelligence*, (1985), 162–164.

28 **Oliver IM, Smith DJ and Holland JRC**. 'A study of permutation crossover operators on the traveling salesman problem', *In: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, (1987), 224–230.