International Journal of Production Research
Vol. 47, No. 7, 1 April 2009, 1989–2007

Taylor & Francis
Taylor & Francis Group

An Enhanced Grouping Genetic Algorithm for solving the cell formation problem

T. Tunnukij and C. Hicks*

Newcastle University Business School, Newcastle University, Newcastle upon Tyne NE1 7RU, UK

(*Received 1 November 2006; final version received 16 August 2007*)

Cell formation is often the first step in solving facility layout design problems. The objective is to group part families and machines so that they can be assigned to manufacturing cells. The cell formation problem is a non-deterministic polynomial (NP) complete problem which means that the time taken to produce solutions increases exponentially with problem size. This paper presents the Enhanced Grouping Genetic Algorithm (EnGGA) that has been developed for solving the cell formation problem. The EnGGA replaces the replacement heuristic in a standard Grouping Genetic Algorithm with a Greedy Heuristic and employs a rank-based roulette–elitist strategy, which is a new mechanism for creating successive generations. The EnGGA was tested using well-known data sets from the literature. The quality of the solutions was compared with those produced by other methods using the grouping efficacy measure. The results show that the EnGGA is effective and outperforms or matches the other methods.

Keywords: cell formation; cellular manufacture; clustering; Genetic Algorithms; Group Technology; meta-heuristics

1. Introduction

A well-designed manufacturing facility enhances manufacturing efficiency by reducing material flow, materials handling, work in progress and lead times. Scheduling and the control of operations may also be improved (Wemmerlov and Johnson 1997). Group Technology (GT) is a philosophy that aims to exploit similarities between parts, products and processes to achieve efficiencies (Hyer and Wemmerlov 1984). Cellular Manufacturing (CM) is the application of GT to manufacturing systems. It aims to substantially improve delivery performance and reduce work in progress, throughput time and manufacturing costs (Gallagher and Knight 1973, 1986). The implementation of CM requires parts with similar processing requirements to be grouped into part families. Manufacturing cells are clusters of dissimilar machines placed in close proximity that are dedicated to the manufacture of families of parts (Wemmerlov and Hyer 1989). Over the last three decades, a large number of clustering methods have been developed for identifying potential manufacturing cells. Many of these methods are based upon a machine–part incidence matrix (Askin and Standridge 1993). The objective is to rearrange the matrix to create a

*Corresponding author. Email: chris.hicks@ncl.ac.uk

ISSN 0020–7543 print/ISSN 1366–588X online

DOI: 10.1080/00207540701673457
http://www.informaworld.com

block diagonal structure from which families of parts and the machines required to produce them can be selected.

The optimization of the cell formation problem (CFP) has been shown to be a non-deterministic polynomial (NP) complete problem (Dimopoulos and Zalzala 2000), which means that the amount of computation increases exponentially with problem size. Even a powerful computer can take an unacceptably long time to solve a large problem due to combinatorial diffusion. Stochastic search methods are particularly suitable for solving complex combinatorial optimisation problems. They are able to search large regions of the solution space without becoming trapped in local optima. Commonly used methods include Genetic Algorithms (Holland 1975), Tabu search (Glover 1989) and Simulated Annealing (Kirkpatrick *et al*. 1983).

Genetic Algortithms (GAs) are derived from an analogy with biological evolution, in which the fitness of an individual determines its ability to survive and reproduce (Goldberg 1989). Falkenauer (1998) developed a Grouping Genetic Algorithm (GGA) that suited the structure of grouping problems. Brown and Sumichrast (2005) evaluated the performance of GGAs and suggested that GGAs are generally better than GAs for solving grouping optimisation problems because they are more computationally efficient.

The objectives of this paper are to:

- review the methods that have been used for identifying potential manufacturing cells by solving the cell formation problem;
- describe the development of the Enhanced Grouping Genetic Algorithm (EnGGA) that substitutes the replacement heuristic in a standard Grouping Genetic Algorithm with a Greedy Heuristic. It also employs the rank-based roulette–elitist strategy, which is a new mechanism for creating successive generations;
- report the results of experiments that tested the EnGGA using data sets from the literature;
- compare the quality of the solutions produced by the EnGGA with those produced by other methods.

Section 2 reviews the literature relating to the CFP. Section 3 provides an overview of GAs and GGAs for solving the CFP. Section 4 describes the development of the EnGGA algorithm. Section 5 presents the computational results obtained with datasets from the literature and compares the performance of the EnGGA with other methods. The conclusions are presented in Section 6.

2. The cell formation problem

The cell formation problem (CFP) groups machines into machine cells and parts into part families (Hu and Yasuda 2006). Well-designed manufacturing cells should maximise the machine utilisation within each machine cell and minimise the inter-cell flow of parts. Ballakur and Steudel (1987) identified three approaches to grouping employed by cell formation methods:

1. *part family grouping*, which forms part families and then groups machines into cells;
2. *machine grouping*, which forms machine cells based upon similarities in part routings and then allocates parts to cells;
3. *machine–part grouping*, which forms part families and machine cells simultaneously.

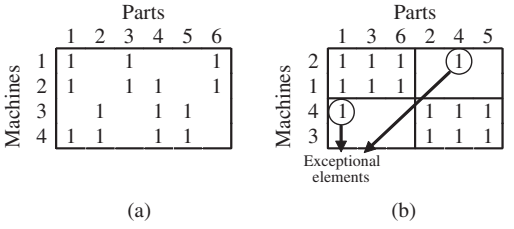

Figure 1. A machine–part incidence matrix: (a) the original matrix; (b) a rearranged matrix into block-diagonal forms.

The relationships between parts and machines may be represented as a machine–part incidence matrix (see Figure 1). For example, in Figure 1, part 1 is processed by machines 1, 2 and 4. Clustering methods based upon the machine–part incidence matrix aim to minimise the number of voids in the diagonal blocks and the number of exceptional elements (or 1's) outside the diagonal blocks, which create inter-cell flow. Kumar and Chandrasekharan (1990) proposed the grouping efficacy measure (Γ) as a quantitative criterion for measuring the quality of block diagonal forms. This measure has been widely used in the literature:

$$\Gamma = 1 - \frac{e_o + e_v}{e + e_v} \tag{1}$$

where e is the total number of operations (number of 1's in the matrix), e_o is the number of 1's in the off-diagonal blocks and e_v is the number of voids in the diagonal blocks.

Methods based upon the machine–part incidence matrix include the Bond Energy Algorithm (McCormick *et al*. 1972), the Direct Clustering Algorithm (Chan and Milner 1982), Rank Order Clustering (King 1980), MODROC (Chandrasekharan and Rajagopalan 1986), ZODIAC (Chandrasekharan and Rajagopalan 1987), GRAFICS (Srinivasan and Narendran 1991) and the Close Neighbour Algorithm (Boe and Cheng 1991). Unfortunately, these methods do not always produce solutions with the desired diagonal structure (Hicks 2004).

Methods based upon similarity coefficients have been used as an alternative approach for both part family grouping (Carrie 1973) and machine grouping (McAuley 1972, Gupta and Seifoddini 1990). A number of similarity and dissimilarity coefficients between parts and/or machines have been proposed for grouping part families and/or machine cells (Shafer and Rogers 1993a,b, Islam and Sarker 2000). Though various similarity coefficients have been proposed, no particular similarity coefficient is effective in all situations (Sarker 1996). In practice, when some large complex manufacturing systems are considered, the results produced by similarity coefficients methods may be inconclusive (Hicks 2004).

Graph theoretical methods are an alternative hierarchical clustering approach based upon machine grouping (Rajagopalan and Batra 1975). A disadvantage of hierarchical methods is that they do not form part families and machine cells simultaneously; additional methods must be employed to complete the formation of cells, particularly when dealing with complex manufacturing systems (Goncalves and Resende 2004).

Mathematical programming-based methods have been used for part family grouping, machine grouping and machine–part grouping (Kusiak 1987, Won 2000).

These mathematical programming-based methods allow designers to consider a variety of objectives, however they can only be used for relatively small problems and they do not always produce desirable solutions (Joines *et al.* 1996, Hicks 2004).

Various heuristic methods have been developed to solve the CFP. They have considered production variables, such as costs, processing times and capacity utilisation, as well as exception elements, operation sequences and intra- and inter-cell flow (Askin and Subramanian 1987, Kumar and Vannelli 1987, Heragu and Gupta 1994). However, in practice, some of the production variables may be difficult to evaluate and the optimum solutions may not be robust in all situations (Singh 1993). Most of these algorithms are highly sensitive to the number of cells and the maximum number of machines or parts within each cell, which are usually predetermined in advance. Therefore, if these parameters are selected improperly, the clustering methods may produce unsatisfactory results (Tsai *et al.* 1997).

Since the CFP has been shown to be an NP-complete problem, traditional optimisation methods are incapable of finding optimal solutions to larger problems within a reasonable amount of time (Dimopoulos and Zalzala 2000, Goncalves and Resende 2004). Heuristic methods can be used for large problems, but they often become trapped in local optima (De Lit *et al.* 2000). More recently, stochastic optimisation algorithms (meta-heuristic methods) have been used for solving the CFP. They can find global or near-global optimal solutions within a reasonable amount of computation time (Goncalves and Resende 2004). Commonly used stochastic optimisation algorithms include Simulated Annealing (Kirkpatrick *et al.* 1983), Tabu search (Glover 1989) and Genetic Algorithms (Holland 1975). Simulated Annealing (SA) has been used for solving the CFP (Boctor 1991, 1996, Adil *et al.* 1996, Sofianopoulou 1999). Tabu Search (TS) has been applied to the CFP by Logendran *et al.* (1994), Aljaber *et al.* (1997), and Adenso-Diaz *et al.* (2001).

Simulated Annealing (SA) and Tabu Search (TS) are both unidirectional search methods, where the search starts from a single initial state. Genetic Algorithms (GAs) operate on a set of solutions (chromosomes) simultaneously. They use information from all the current points to direct the search towards promising regions in the solution space (Venugopal and Narendran 1992, Uddin and Shanker 2002). They are less susceptible to becoming trapped in local optima (Yasuda *et al.* 2005). GAs search in multiple directions and are more likely to search throughout large search spaces. These two features enable GAs to tackle NP-complete problems successfully (Venugopal and Narendran 1992, Uddin and Shanker 2002). Aytug *et al.* (2003) produced a comprehensive review of the use of GAs for solving a wide range of production and operations management problems including the CFP.

3. Genetic algorithms and grouping genetic algorithms

A Genetic Algorithm is a competitive method that may be used to solve large, unsmooth or noisy problems. GAs may find a 'good' solution rather than the global optimum (Mitchell 1996). One of the main advantages of GAs is that they only require an objective function (or 'fitness function') that can be evaluated numerically. They do not require a mathematical representation of the problem. GAs can be used for nonlinear problems that are defined on discrete, continuous or mixed search spaces that may be unconstrained or constrained. GAs are able to explore different regions of the solution space in parallel and direct the search towards promising regions in the space (Goldberg 1989).

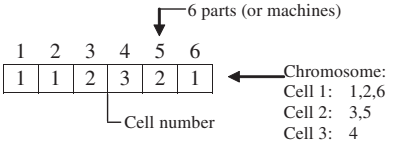

Figure 2. A general chromosome representation of GAs for the CFP.

Aytug *et al.* (2003) identified eight main components within GAs: (i) genetic representation; (ii) method for generating the initial population; (iii) evaluation function; (iv) reproduction selection scheme; (v) genetic operators; (vi) mechanism for creating successive generations; (vii) stopping criteria; and (viii) parameter settings.

Since the CFP is an NP-complete problem, GAs have been widely used to solve the problem. Venugopal and Narendran (1992) were the first researchers to apply GAs to the CFP. Their objective was to minimise the inter-cell flow and the total cell load variation within a predetermined number of manufacturing cells. Each machine corresponded to a gene. An integer in a gene indicated the cell that contained a machine, and the position of the gene within the chromosome represented the machine number. This genetic representation has been commonly used for solving the CFP by many researchers (see, for example, Gupta *et al.* (1996), Moon and Gen (1999), Plaquin and Pierreval (2000), Zolfaghari and Liang (2002)).

An example of a chromosome representation used for solving the CFP is shown in Figure 2. In this example, a chromosome represents the solution of a part (or machine) grouping problem that contains three cells. The first cell contains parts (or machines) 1, 2 and 6. The second cell contains parts (or machines) 3 and 5. The final cell contains part (or machine) 4.

Alternative approaches include: (i) the binary number representation where a gene is represented by 1 if machine j is assigned to a cell I, and 0 otherwise (Rao *et al.* 1999, Wicks and Reasor 1999); (ii) the integer representation where the integer in a gene corresponds to a machine number or a part number (Hwang and Sun 1996, Cheng *et al.* 1998); and (iii) the vector representation of real numbers (Goncalves and Resende 2004). Comprehensive reviews of the use of GAs for solving the CFP can be found in Dimopoulos and Zalzala (2000) and Pierreval *et al.* (2003).

The results provided in the literature show that GAs can outperform traditional methods. Some heuristics have been combined with GAs in order to enhance their performance. Hwang and Sun (1996) combined a GA with a Greedy Heuristic, which always chooses the best choice available (Cormen *et al.* 2001). Goncalves and Resende (2004) combined a GA with a local search heuristic. The local search heuristic aimed to improve the quality of the solutions by refining the chromosomes generated whenever possible. If the modified solution was better than the original solution, the original solution was replaced. The heuristic iterated until the quality of the new solution was no better than the quality of the previous solution. Most of these methods that have used GAs have assumed that the number of manufacturing cells is known in advance (Hu and Yasuda 2006).

Falkenauer (1998) developed a Grouping Genetic Algorithm (GGA) to optimise grouping problems efficiently. The GGA differs from classical GAs in two important aspects: (i) a special gene encoding scheme was developed to represent grouping problems

within chromosomes; and (ii) special genetic operators were developed that suited the structure of these chromosomes.

In classical GAs, the standard gene encoding scheme includes significant redundancy when representing grouping problems (Falkenauer 1998). For example, chromosomes ABAC and CACB both represent a solution where the first and third items are in the same group and the second and the fourth items are in different groups. This repetition increases the size of the search space and potentially reduces the effectiveness of the GAs. The GGA gene encoding scheme focuses upon the contents of the groups, not their ordering. An additional group portion that contains a list of the groups is added to the main portion of each chromosome. This modification to the standard gene encoding scheme allows the modified crossover and mutation operators to manipulate the group portion of the chromosome. This allows groups to be modified as a whole, rather than modifying individual members (Brown and Sumichrast 2003). The gene encoding scheme and the modified genetic operators enable the GGA to efficiently find high-quality solutions for a wide range of grouping problems (Brown and Sumichrast 2005).

De Lit *et al*. (2000) used the GGA to solve the CFP with a fixed maximum cell size. Brown and Sumichrast (2001) tested the GGA using data sets from the literature. This work did not predetermine the number of manufacturing cells or the number of machines within the cell. It included a replacement heuristic that was used as part of the crossover operator, which enhanced the performance of the GGA (Brown and Sumichrast 2001, 2003). Although GGAs are generally better than GAs for solving the cell formation problem because they are more computationally efficient (Brown and Sumichrast 2005), a GA with a local search heuristic proposed by Goncalves and Resende (2004) produced better results than the standard GGA in most cases. James *et al*. (2007) combined the standard GGA with the local search heuristic proposed by Goncalves and Resende (2004) to produce a hybrid GGA. It outperformed the standard GGA and produced better solutions in all cases. It also reduced the variability amongst the solutions found. It mostly outperformed other methods, including the GA with a local search heuristic. However, the hybrid GGA required more computation time than the standard GGA due to the local search heuristic that was used to generate each chromosome.

Yasuda *et al*. (2005) used the GGA to solve multi-objective cell formation problems. Their objectives were to minimise the cell load variation and the inter-cell flows whilst considering machine capacities, part volumes and part processing times on machines. Hu and Yasuda (2006) used the GGA to solve the cell formation problem with alternative processing routes. Their objective was to minimise the total cost of material flow between cells and within the cells. They assumed that the inter-cell movement cost was directly proportional to the number of cells and that the intra-cell movement cost was inversely proportional to the number of cells. However, these assumptions may be invalid in reality because transportation costs usually depend upon how the layout and transportation system are designed, which are determined by further steps of the facilities layout problem. In addition, transportation costs are a function of the weight and size of parts.

Although the consideration of factors such as machine capacities, part processing times and alternative processing routes can be taken into account, they may make the analysis very complicated, which can be a problem for practitioners. The 0–1 machine–part incidence matrix is easier for practitioners to comprehend. It provides a representation of the initial cell formation that can form the basis for further steps of the facility layout design process. The design produced can be subsequently modified to take other factors into consideration (Cheng *et al*. 1998).

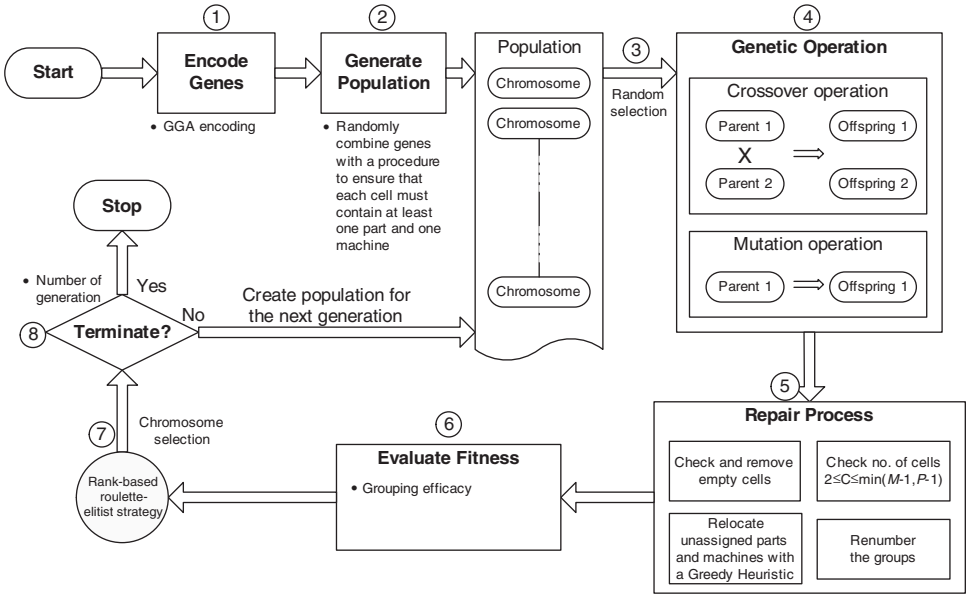

Figure 3. The general structure of the EnGGA.

4. Enhanced grouping genetic algorithm

The Enhanced GGA (EnGGA) reported in this paper was developed by improving the configuration of the standard GGA proposed by Brown and Sumichrast (2001). The EnGGA replaces the replacement heuristic in the standard GGA with a Greedy Heuristic. It employs a rank-based roulette–elitist strategy that combines the elitist strategy (Goldberg 1989) with a rank-based roulette wheel (Reeves 1995). This is a new mechanism for creating successive generations. The EnGGA uses the GGA encoding strategy proposed by Falkenauer (1998). The GGA crossover operator, elimination mutation operator and division mutation operator were used with minor modifications. The EnGGA includes a repair process that rectifies infeasible chromosomes that may be produced during the evolution process. The general structure of the EnGGA is shown in Figure 3.

The EnGGA uses the 0–1 machine–part incidence matrix to represent the initial configuration. The EnGGA can solve the CFP without predetermining the number of manufacturing cells or the number of machines and parts within each cell. However, there is no point in clustering all the machines (M) and all the parts (P) into only one cell or having only one machine in each cell. Therefore, the possible number of cells (C) is defined as $2 \leq C \leq \min(M-1,\ P-1)$.

4.1 *Genetic representation*

The first stage of the EnGGA process encodes the machine–part grouping problem into genes. The GGA encoding scheme is used. The chromosome representation (shown in Figure 4) consists of three sections: (i) the part section; (ii) the machine section; and

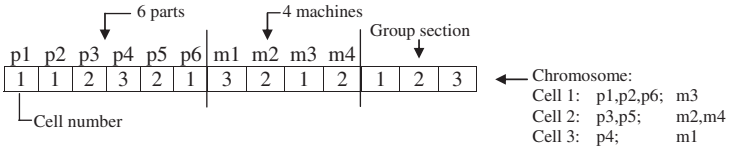

Figure 4. A chromosome representation of the EnGGA for the CFP.

(iii) the group section. Each gene in the part and machine sections contains an integer that represents the cell number. The part and machine numbers are represented by the position of the genes within the appropriate section. Note that the integers that represent cell numbers in the part and machine sections are for information only because the genetic operators only work on the group section. The length of individual chromosomes may differ because the number of cells in alternative solutions may vary. The chromosome length is therefore equal to the sum of the number of parts (P), the number of machines (M) and the number of cells (C), where C varies from 2 to $\min(M-1, P-1)$. The order in which the cells in the group section are listed does not matter. This representation allows the machine–part grouping approach to be used. It also allows the modified crossover and mutation operators to be performed on the group portion of the chromosome. As a result, the groups are modified as a whole, rather than by modifying individual members. This is a computationally efficient approach. Figure 4 illustrates this representation with a chromosome that represents a possible solution to the machine–part grouping problem shown in Figure 1(a). The group section shows that the machines and parts are allocated to three cells. The first cell contains parts 1, 2, 6 and machine 3. The second cell contains parts 3 and 5 together with machines 2 and 4. The final cell contains part 4 and machine 1.

4.2 *Method for generating the initial population*

The initial population of chromosomes is generated randomly. This process is as follows:

1. C cells are randomly generated, where C is a random positive integer where $2 \leq C \leq M-1$ if $M < P$, otherwise $2 \leq C \leq P-1$;
2. C parts and C machines are randomly selected; the parts and machines are then assigned to cells so that each cell contains at least one part and one machine;
3. the remaining parts and machines are randomly allocated into the cells;
4. steps 1–3 above are repeated until a population of the required size (Pop) is produced.

4.3 *Reproduction selection scheme*

Chromosomes are randomly selected for the crossover and mutation operations; all chromosomes have an equal probability of selection. The probabilities of crossover (P_c) and mutation (P_m) are pre-specified experimental parameters.

4.4 *Genetic operators*

There are two types of genetic operators: (i) crossover, the 'focusing operator', which helps the GA move towards a local optimum by exploiting the current neighbourhood; and

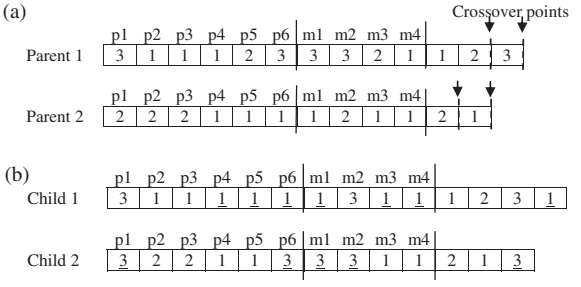

Figure 5. Falkenauer's crossover operator. (a) select crossover points; (b) injection.

(ii) mutation, the 'exploration operator', which tends to randomly move the search to a new neighbourhood in order to avoid becoming trapped in a local optimum (Aytug *et al.* 2003). Crossover tends to make the chromosomes within the population more similar, whereas mutation tends to make them more diverse (Holland 1975, Goldberg 1989).

In this research, Falkenauer's (1998) crossover, elimination mutation and division mutation operators were adopted (with minor modifications). They were integrated with a new repair process that rectifies infeasible chromosomes produced by genetic operations. The crossover operator includes two steps, which are shown in Figure 5.

(a) Two parents are randomly chosen from the population; two crossover points are then randomly selected from the group section of each parent. Figure 5(a) shows two parents (that both represent possible solutions to the machine–part grouping problem shown in Figure 1(a)) and their crossover points.
(b) All the genes from the first parent are initially copied to the first child. Likewise, all the genes from the second parent are initially copied to the second child. The section within the crossover points of the second parent is appended to the first child; likewise, the section within the crossover points of the first parent is appended to the second child. When genetic information is copied from the second parent to the first child, or from the first parent to the second child, it is shown in underlined text. All the parts and machines that belong to the cells within the appended section are inherited by the child. For example, in Figure 5(b), the first child has inherited cell 1 from the second parent. This cell contains parts 4, 5 and 6 together with machines 1, 3 and 4; they are all inherited by the first child, which replace the genes initially inherited from the first parent.

If the cell formations represented by the two parents are the same, Falkenauer's crossover operator will produce children that are identical to the parents. This phenomenon will trap the search into a local optimum. Therefore, in the EnGGA the two selected parents are compared before they are processed by the crossover operator. If they are the same, a parent that has a different cell formation will be randomly chosen from the population to replace one of the parents. Unfortunately, there is a problem that may arise from this procedure. When the results produced by the algorithm are nearly convergent, the population will include a lot of duplicated chromosomes. As a result, the algorithm may not be able to find two parents that represent different solutions. An alternative approach, proposed by Yasuda *et al.* (2005), is to clone one of the parents to produce one child and create another child randomly. However, this approach may prevent convergence. In this research, the algorithm attempts to randomly choose a parent

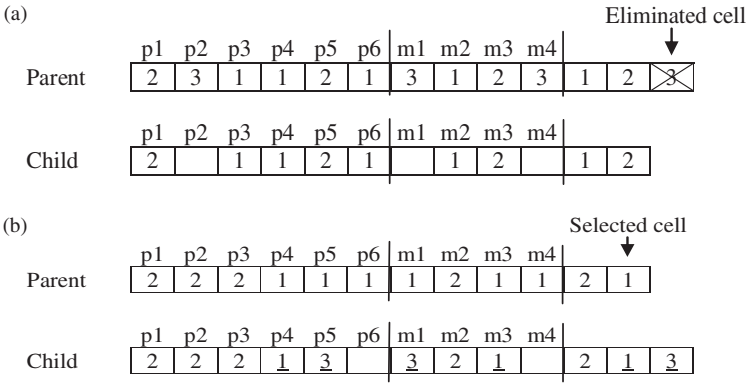

Figure 6. Falkenauer's mutation operators. (a) elimination; (b) division mutation.

that has a different cell formation. If the randomly chosen chromosomes are the same, the process is repeated until either a different chromosome has been chosen, or until 30% of the population has been sampled.

The standard GGA elimination mutation operator and division mutation operator (Falkenauer 1998) were used with minor modifications. The mutation steps, which are shown in Figure 6, are as follows:

1. a parent is chosen from the population randomly;
2. the number of cells is checked:

   (a) if the number of cells is more than two, the standard elimination mutation operator will be used. One of the cells in the group section is randomly selected and all of its elements are eliminated. The remaining elements are inherited by the child (see Figure 6(a));

   (b) if the number of cells is two or less, the modified division mutation operator will be used. With the modified division mutation operation, a cell that contains at least two parts and two machines is randomly selected and then divided into two new cells. Two parts and two machines within the selected cell are randomly selected and are split between the two new cells. This ensures that each new cell contains at least one part and one machine. Figure 6(b) illustrates this process. In this case, cell 1 has been randomly selected as it contains at least two parts and two machines. Cell 1 is then divided into cell 1 and cell 3. The underlined cell numbers indicate that the cells have been created by the division mutation. The next step is to randomly select two parts and two machines from cell 1 to be assigned to cells 1 and 3. In this case, part 4 and machine 3 have been assigned to cell 1, whilst part 5 and machine 1 have been allocated to cell 3. The remaining unassigned elements (part 6 and machine 4) are allocated by the repair process.

4.5 *Repair process*

The chromosomes produced by the genetic operations may represent infeasible solutions. A repair process was developed to rectify infeasible chromosomes. The repair process consists of four stages.

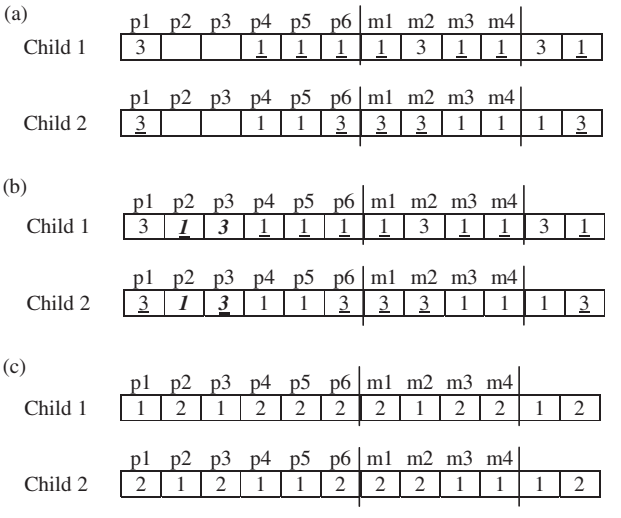

Figure 7. The EnGGA repair process. (a) remove the empty cells; (b) relocate unassigned parts and/or machine by the Greedy Heuristic; (c) renumber the groups.

1. *Checking and removing empty cells*. Each cell must contain at least one part and one machine. For example, in Figure 5(b), children 1 and 2 contain empty cells. Cell 2 in child 1 has no machines or parts, whilst cell 1 has two parts, but no machines. Likewise, cell 2 in child 2 has parts 2 and 3, but no machines. The repair process identifies and then removes the empty cells (see Figure 7(a)).
2. *Checking the number of cells*. The possible number of cells (C) is defined as $2 \leq C \leq \min(M-1,\ P-1)$:
   - if the number of cells within the child produced after step 1 is one, a new cell number will be inserted and unassigned parts and machines will be relocated into the new cell;
   - if the number of cells is more than $\min(M-1,\ P-1)$, a cell will be randomly selected and eliminated until the number of cells is equal to $\min(M-1,\ P-1)$. Unassigned parts and machines will then be relocated into the existing cells by the Greedy Heuristic.
3. *Greedy Heuristic*. Unassigned parts and machines are assigned to the existing cells by a Greedy Heuristic, which is used as an alternative to the replacement heuristic in the standard GGA proposed by Brown and Sumichrast (2001). The Greedy Heuristic evaluates the fitness of all the possible chromosomes that could be produced by all the alternative allocations of unassigned parts and machines. Fitness is measured in terms of the grouping efficacy. Figure 7(a) illustrates this procedure. Child 1 represents a cell formation where cell 1 contains parts 4, 5, 6 and machines 1, 3 and 4; cell 3 contains part 1 and machine 2. However, parts 2 and 3 are unassigned and need to be relocated into either cell 1 or 3. If the original machine–part incidence matrix shown in Figure 1(a) was rearranged to reflect this configuration and part 2 was relocated into cell 1, the grouping efficacy would be 42.10. If part 2 was relocated into cell 3, the grouping efficacy would be 31.58. Therefore, the Greedy Heuristic would place part 2 into cell 1 because that would

generate the highest grouping efficacy. After relocating part 2 into cell 1, part 3 would then be relocated into cell 3 because that would generate the highest grouping efficacy of 50.00 rather than placing it into cell 1 which would generate a grouping efficacy of 42.86. Figure 7(b) shows the solution after relocating unassigned parts and machines using the Greedy Heuristic. The replacement heuristic in the standard GGA would place an unassigned part into the cell that contains the most machines on its routing. In this example, the replacement heuristic would randomly allocate part 3 to cell 1 or cell 3 because part 3 requires one machine in each cell. Thus the replacement heuristic may not select the solution with the highest grouping efficacy. The standard GGA with the replacement heuristic may therefore produce inferior results to the EnGGA with the Greedy Heuristic.

4. *Renumbering the groups* to simplify interpretation. This is illustrated by Figure 7(c). In this example, cell 3 in the first child has been renumbered as cell 1, whilst cell 1 has been renumbered as cell 2. Likewise for the second child, cell 3 has become cell 2 and cell 1 is unchanged.

4.6 *Evaluation criteria*

The best solution produced by the machine–part incidence matrix-based methods minimises the number of voids (zeros) in the diagonal blocks and the number of exceptions (1's outside the diagonal blocks) which represent inter-cell flows. This paper employed the grouping efficacy (Γ) as the objective function for measuring the quality of block diagonal forms.

4.7 *Mechanism for creating successive generations*

Three selection mechanisms have been widely used for creating successive generations: (1) the roulette wheel approach; (2) the tournament approach; and (3) the elitist strategy. With the roulette wheel approach (Goldberg 1989), also known as biased roulette wheel selection, the fitness of a particular chromosome determines the size of its segment on the roulette wheel. The roulette wheel is then 'spun' repeatedly to produce a new population, with the same number of chromosomes as the initial population. With this approach, the chromosomes with low fitness values still have a small probability of being selected for the next generation. However, if there is only a small difference between the highest fitness and the lowest fitness chromosomes, the roulette wheel selection may not always allow the fittest chromosomes to survive (Brown and Sumichrast 2001). Reeves (1995) proposed an algorithm to solve this problem by determining an alternative fitness score for each chromosome as follows. First, the chromosomes were ranked in order from the worst (rank of 1) to the best (rank of N). Then, a chromosome of rank r was assigned a fitness score of $2r/N(N+1)$, where N was the number of chromosomes ranked. However, if a large number of chromosomes were ranked, the difference between the highest fitness chromosomes and the low fitness chromosomes may be small. For example, the best chromosome from 1000 chromosomes is assigned 0.2% of the wheel whilst the 500th chromosome ranked is assigned 0.1% of the wheel; therefore, there is only a 0.1% difference. As a result, with a large population the rank-based roulette wheel selection may not always allow the fittest chromosomes to survive.

The tournament approach and the elitist strategy (Goldberg 1989) are more likely to allow the fittest chromosomes to be replicated in the next generation. However, they may be dominated by a small number of fit chromosomes which may reduce the amount of search. Higher probabilities of mutation can be employed to prevent the solution from becoming trapped in a local optimum, but this may also prevent convergence.

The EnGGA was tested with seven alternative selection mechanisms: (1) the roulette wheel approach (Goldberg 1989); (2) the rank-based roulette wheel approach (Reeves 1995); (3) the tournament approach (Goldberg 1989); (4) the stochastic remainder sampling without replacement approach (Goldberg 1989); (5) the elitist strategy (Goldberg 1989); (6) the roulette–elitist strategy; and (7) the rank-based roulette–elitist strategy. The first five selection mechanisms are well established selection mechanisms. The last two are new selection mechanisms that were developed in this research. The roulette–elitist strategy combines the elitist strategy with the roulette wheel approach. The elitist strategy is used to select successive chromosomes by copying the best chromosomes from the previous generation to the next generation (the percentage copied is an experimental parameter) and the roulette wheel approach (Goldberg 1989) is then used to select other successive chromosomes. The rank-based roulette–elitist strategy also employs the elitist strategy to select the fittest chromosomes, but it uses the rank-based roulette wheel (Reeves 1995) to select other successive chromosomes. It was found that the rank-based roulette–elitist strategy with 15% of the best chromosomes surviving to the next generation produced the best results. It was therefore chosen for the EnGGA.

4.8 *Stopping criteria*

The EnGGA terminates when a fixed number of generations have been completed. The cell formation configuration associated with the highest fitness chromosome is then shown.

5. Analysis of performance using data obtained from the literature

The EnGGA was tested using datasets from the literature. A full factorial experiment considered the parameter settings shown in Table 1. In this research, the sum of probabilities of crossover (P_c) and mutation (P_m) was defined as $P_c + P_m \leq 1$. Therefore, if P_c was fixed at 1.0, there was no mutation.

The EnGGA was tested with a set of 24 problems that have been published in the literature and have been widely used in many comparative studies. All the data sets were transcribed from the original articles. The sources of the problems are shown in Table 2. The EnGGA was written in C and was tested on a laptop with a 1.66 GHz processor.

Table 1. Experimental parameter settings.

Parameter	Levels
Population size (*Pop*)	100 (data 1–9), 1000 (data 10–24)
Probability of crossover (P_c)	0.6, 0.7, 0.8, 0.9, 1.0
Probability of mutation (P_m)	0.1, 0.2, 0.3, 0.4
No. of generations	50

Table 2. A comparison of the results obtained by 13 clustering algorithms.

Problem source	Size	ZODIAC	GRAFICS	MST	TSP–GA	GP–GA	Zolfaghari & Liang's SA	Zolfaghari & Liang's GA	Zolfaghari & Liang's TS	EA–GA	CF–GGA	MOGGA	HGGA	EnGGA Max sol.	EnGGA Pop	EnGGA Time(s)	EnGGA Gen.
1. King and Nakornchai (1982)	5 × 7	73.68	73.68							73.68[b]			82.35	82.35	100	<1	2
2. Waghodekar and Sahu (1984, Figure 4a)	5 × 7	56.52	60.87		68.00					62.50			69.57	69.57	100	<1	2
3. Seifoddini (1989)	5 × 18				77.36					79.59			79.59	79.59	100	<1	3
4. Kusiak and Cho (1992)	6 × 8				76.92					76.92			76.92	76.92	100	<1	3
5. Kusiak and Chow (1987)	7 × 11	39.13	53.12		46.88		58.62	58.62	58.62	53.13			60.87	60.87[a]	100	<1	6
6. Boctor (1991, Figure 1)	7 × 11				70.37					70.37			70.83	70.83[a]	100	<1	3
7. Seifoddini and Wolfe (1986)	8 × 12	68.30	68.30				68.29	68.29	68.29	68.30[b]			69.44	69.44[a]	100	<1	5
8. Chandrasekharan and Rajagopalan (1986b)	8 × 20	85.24	85.24	85.24	85.24	85.20	85.25	85.25	85.25	85.25	85.25		85.25	85.25	100	<1	4
9. Chandrasekharan and Rajagopalan (1986a)	8 × 20	58.33	58.13	58.72	58.33	58.70				58.72	56.88		58.72	58.72	100	<1	3
10. McCormick *et al.* (1972)	16 × 24	32.09	45.52	48.70						52.58[b]			52.75	53.26[a]	1000	21	10
11. Srinivasan *et al.* (1990)	16 × 30	67.83	67.83	67.83						67.83[b]			68.99	68.99[a]	1000	21	8
12. King (1980)	16 × 43	53.76	54.39	54.44	53.89		53.93	53.63	52.25	54.86	53.70	55.43	57.53	57.53[a]	1000	27	15
13. Carrie (1973)	18 × 24	41.84	48.91	44.20						54.46[b]	52.38		57.73	57.73[a]	1000	25	9
14. Carrie (1973)	20 × 35	75.14	75.14	75.14	75.28	76.70	75.14	70.33	62.37	76.22[b]	77.91		77.91	77.91[a]	1000	30	6
15. Boe and Cheng (1991)	20 × 35					56.80	56.17	54.29	51.33	58.07[b]			57.98	57.98[a]	1000	26	8
16. Chandrasekharan and Rajagopalan (1989) Data 1	24 × 40	100.0	100.0	100.0	100.0	100.0	100.0	80.14	71.97	100.0	100.0	100.0	100.0	100.0	1000	28	6
17. Chandrasekharan and Rajagopalan (1989) Data 2	24 × 40	85.11	85.11	85.11	85.11	85.10	85.11	68.32	66.46	85.11	85.11	85.11	85.11	85.11	1000	27	7
18. Chandrasekharan and Rajagopalan (1989) Data 3	24 × 40		73.51	73.51	73.03	73.50	73.51	61.11	62.05	73.51	73.29	73.03	73.51	73.51	1000	23	10
19. Chandrasekharan and Rajagopalan (1989) Data 5	24 × 40	20.42	43.27	51.81	49.37	53.30				51.88	48.98	48.98	53.29	53.29[a]	1000	23	12
20. Chandrasekharan and Rajagopalan (1989) Data 6	24 × 40	18.23	44.51	44.72	44.67	47.90				46.69	46.81	45.00	48.95	48.95[a]	1000	22	15
21. Chandrasekharan and Rajagopalan (1989) Data 7	24 × 40	17.61	41.67	44.17	42.50	43.70				44.75	44.14	41.90	47.26[c]	46.58[a]	1000	23	17
22. McCormick *et al.* (1972)	27 × 27	52.14	47.37	51.00						54.27[b]			54.02	54.82[a]	1000	22	8
23. Kumar and Vannelli (1987)	30 × 41	33.46	55.43	55.29	53.80	60.70	42.75	39.64	37.28	58.11[b]			63.31	63.31[a]	1000	24	14
24. McCormick *et al.* (1972)	37 × 53	52.21	52.21							56.42[b]			60.64	60.64[a]	1000	38	8

[a]Solutions where singletons appear.
[b]Data reported in Goncalves and Resende (2004) was inconsistent with data in the original reference.
[c]Value reported in James *et al.* (2007) was inconsistent with the value calculated from the block diagonal matrix for the solution.

The EnGGA was compared with the other methods from the literature listed in Table 2. These methods included: (i) ZODIAC (Chandrasekharan and Rajagopalan 1989); (ii) GRAFICS (Srinivasan and Narendran 1991); (iii) MST-Clustering Algorithm (Srinivasan 1994); (iv) TSP-GA (Cheng *et al*. 1998); (v) GP-GA (Dimopoulos and Mort 2001); (vi) Zolfaghari and Liang's Simulated Annealing (SA) (Zolfaghari and Liang 2002); (vii) Zolfaghari and Liang's GA (Zolfaghari and Liang 2002); (viii) Zolfaghari and Liang's Tabu Search (TS) (Zolfaghari and Liang 2002); (ix) EA-GA (Goncalves and Resende 2004); (x) CF-GGA (Brown and Sumichrast 2001); (xi) MOGGA (Yasuda *et al*. 2005); and (xii) HGGA (James *et al*. 2007). The results of ZODIAC and GRAFICS were obtained from Srinivasan and Narendran (1991); otherwise the results were obtained from the original articles.

There were several issues that needed to be considered when interpreting results. Dimopoulos and Mort (2001) only reported their results to one decimal place. Some of the data sets reported in Goncalves and Resende (2004) were inconsistent with data in the original references. In problem 21, shown in Table 2, the grouping efficacy reported by James *et al*. (2007) was inconsistent with the grouping efficacy calculated from the block diagonal solution matrix that they provided. These inconsistencies are marked in the table. Zolfaghari and Liang (2002) used a fixed computational time of 10 s to obtain the solutions; it is possible that better solutions could have been achieved with more computational time. ZODIAC, GRAFICS and EA-GA did not allow the presence of singletons (cells containing only one machine or one part) which may have reduced the quality of the solutions produced by these algorithms. TSP-GA, Zolfaghari and Liang's algorithms, GP-GA, CF-GGA, MOGGA, HGGA, and EnGGA all allowed singletons. In Table 2, the best solutions including singletons found by the EnGGA are shown. The computational time in seconds and the generation when the best solution was found are also reported.

In terms of the grouping efficacy measure, the EnGGA produced results that were equal to, or better than, all the other methods. For problem 15, EA-GA apparently produced a better solution than the EnGGA but this was due to an error within the data used by Goncalves and Resende (2004) (the data used was different from the original reference). With problem 21, the result of the HGGA apparently outperformed the EnGGA, but the grouping efficacy reported in James *et al*. (2007) was inconsistent with that calculated from the block diagonal solution matrix provided, which was only 45.27 (the EnGGA result was 46.58). For problems 10 and 22, the EnGGA found the best solutions. When the data and calculation errors in the literature are taken into account, the EnGGA produced the best solutions in all cases. The EnGGA also performed better than other GGAs including the standard GGA (Brown and Sumichrast 2001), the MOGGA (Yasuda *et al*. 2005), and the HGGA (James *et al*. 2007) that combined the standard GGA with a local search heuristic (Goncalves and Resende 2004).

The computational time required to run the EnGGA with 50 generations was less than 40 s, even for the large population size. For problems 1–9, the small problems, the EnGGA took less than 1 s to run, even with the large population size of 100. The best solution for each problem was found within 20 generations. In terms of parameter settings, the results showed that the combination of a P_c of 0.6–0.7 together with a P_m of 0.1–0.3 and the combination of a P_c of 0.9 together with a P_m of 0.1 produced the highest-quality solutions.

6. Conclusions

A large number of methods have been developed to solve the cell formation problem (CFP). Since the CFP has been shown to be an NP-complete problem, meta-heuristic methods or stochastic optimization algorithms have been widely used because they can produce global or near-global optimal solutions within a reasonable amount of computation time.

This paper has presented the Enhanced Grouping Genetic Algorithm (EnGGA) that can solve the CFP without predetermining the number of manufacturing cells or the number of machines and parts within each cell. The EnGGA replaces the replacement heuristic in a standard Grouping Genetic Algorithm with a Greedy Heuristic and employs a rank-based roulette–elitist strategy, which is a new mechanism for creating successive generations. The EnGGA was tested using well-known data sets from the literature. The quality of the solutions was compared with other methods using the grouping efficacy measure. The results show that the EnGGA is effective and outperforms all the other methods considered. The program required less than 1 min computational time in all situations, even with the large population size.

References

Adenso-Diaz, B., Lozano, S., Racero, J. and Guerrero, F., 2001. Machine cell formation in generalized Group Technology. *Computers & Industrial Engineering*, 41, 227–240.

Adil, G.K., Rajamani, D. and Strong, D., 1996. Cell formation considering alternate routings. *International Journal of Production Research*, 34, 1361–1380.

Aljaber, N., Baek, W. and Chen, C.-L., 1997. Tabu search approach to the cell formation problem. *Computers & Industrial Engineering*, 32, 169–185.

Askin, R.G. and Subramanian, S.P., 1987. Cost-based heuristic for Group Technology configuration. *International Journal of Production Research*, 25, 101–113.

Askin, R.G. and Standridge, C.R., 1993. *Modeling and analysis of manufacturing systems*. New York: Wiley.

Aytug, H., Khouja, M. and Vergara, F.E., 2003. Use of Genetic Algorithms to solve production and operations management problems: a review. *International Journal of Production Research*, 41, 3955–4009.

Ballakur, A. and Steudel, H.J., 1987. A within-cell utilization based heuristic for designing cellular manufacturing systems. *International Journal of Production Research*, 25, 639–665.

Boctor, F.F., 1991. A linear formulation of the machine-part cell formation problem. *International Journal of Production Research*, 29, 343–356.

Boctor, F.F., 1996. Minimum-cost, machine–part cell formation problem. *International Journal of Production Research*, 34, 1045–1063.

Boe, W.J. and Cheng, C.H., 1991. Close neighbour algorithm for designing cellular manufacturing systems. *International Journal of Production Research*, 29, 2097–2116.

Brown, E.C. and Sumichrast, R.T., 2001. CF-GGA: a Grouping Genetic Algorithm for the cell formation problem. *International Journal of Production Research*, 39, 3651–3669.

Brown, E.C. and Sumichrast, R.T., 2003. Impact of the replacement heuristic in a Grouping Genetic Algorithm. *Computers & Operations Research*, 30, 1575–1593.

Brown, E.C. and Sumichrast, R.T., 2005. Evaluating performance advantages of Grouping Genetic Algorithms. *Engineering Applications of Artificial Intelligence*, 18, 12.

Carrie, A.S., 1973. Numerical taxonomy applied to Group Technology and plant layout. *International Journal of Production Research*, 11, 399–416.

Chan, H.M. and Milner, D.A., 1982. Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing Systems*, 1, 65–74.

Chandrasekharan, M.P. and Rajagopalan, R., 1986a. Ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research*, 24, 451–464.

Chandrasekharan, M.P. and Rajagopalan, R., 1986b. MODROC: an extension of Rank Order Clustering for Group Technology. *International Journal of Production Research*, 24, 1221–1233.

Chandrasekharan, M.P. and Rajagopalan, R., 1987. ZODIAC—an algorithm for concurrent formation of part-families and machine-cells. *International Journal of Production Research*, 25, 835–850.

Chandrasekharan, M.P. and Rajagopalan, R., 1989. GROUPABILITY: an analysis of the properties of binary data matrices for Group Technology. *International Journal of Production Research*, 27, 1035–1052.

Cheng, C.H., Gupta, Y.P., Lee, W.H. and Wong, K.F., 1998. TSP-based heuristic for forming machine groups and part families. *International Journal of Production Research*, 36, 1325–1337.

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., 2001. *Introduction to algorithms*. Cambridge, MA: MIT Press.

De Lit, P., Falkenauer, E. and Delchambre, A., 2000. Grouping Genetic Algorithms: an efficient method to solve the cell formation problem. *Mathematics and Computers in Simulation*, 51, 257–271.

Dimopoulos, C. and Zalzala, A.M.S., 2000. Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. *IEEE Transactions on Evolutionary Computations*, 4, 93–113.

Dimopoulos, C. and Mort, N., 2001. A hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems. *International Journal of Production Research*, 39, 1–19.

Falkenauer, E., 1998. *Genetic algorithms and grouping problems*. New York: Wiley.

Gallagher, C.C. and Knight, W.A., 1973. *Group technology*. London: Butterworth.

Gallagher, C.C. and Knight, W.A., 1986. *Group technology production methods in manufacture*. New York: Wiley.

Glover, F., 1989. Tabu search—Part I. *ORSA Journal of Computers*, 1, 190–206.

Goldberg, D.E., 1989. *Genetic algorithms in search, optimisation and machine learning*. Reading, MA: Addison-Wesley.

Goncalves, J.F. and Resende, M.G.C., 2004. An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering*, 47, 247–273.

Gupta, T. and Seifoddini, H., 1990. Production data based similarity coefficient for machine–component grouping decisions in the design of a cellular manufacturing system. *International Journal of Production Research*, 28, 1247–1269.

Gupta, Y., Gupta, M., Kumar, A. and Sundaram, C., 1996. A Genetic Algorithm-based approach to cell composition and layout design problems. *International Journal of Production Research*, 34, 447–482.

Heragu, S.S. and Gupta, Y.P., 1994. Heuristic for designing cellular manufacturing facilities. *International Journal of Production Research*, 32, 125–140.

Hicks, C., 2004. A Genetic Algorithm tool for designing manufacturing facilities in the capital goods industry. *International Journal of Production Economy*, 90, 199–211.

Holland, J.H., 1975. *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Hu, L. and Yasuda, K., 2006. Minimising material handing cost in cell formation with alternative processing routes by Grouping Genetic Algorithm. *International Journal of Production Research*, 44, 2133–2167.

Hwang, H. and Sun, J.-U., 1996. Genetic-algorithm-based heuristic for the GT cell formation problem. *Computers & Industrial Engineering*, 30, 941–955.

Hyer, N.L. and Wemmerlov, U., 1984. Group Technology and productivity. *Harvard Business Review*, 62, 140–149.

Islam, K.M.S. and Sarker, B.R., 2000. A similarity coefficient measure and machine–parts grouping in cellular manufacturing systems. *International Journal of Production Research*, 38, 699–720.

James, T.L., Brown, E.C. and Keeling, K.B., 2007. A hybrid Grouping Genetic Algorithm for the cell formation problem. *Computers & Operations Research*, 34, 2059–2079.

Joines, J.A., Culbreth, C.T. and King, R.E., 1996. Manufacturing cell design: an integer programming model employing Genetic Algorithms. *IIE Transactions (Institute of Industrial Engineerings)*, 28, 69–85.

King, J.R., 1980. Machine-component grouping in production flow-analysis – an approach using a Rank Order Clustering-Algorithm. *International Journal of Production Research*, 18, 213–232.

King, J.R. and Nakornchai, V., 1982. Machine-component group formation in Group Technology – review and extension. *International Journal of Production Research*, 20, 117–133.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., 1983. Optimization by Simulated Annealing. *Science*, 220, 671–680.

Kumar, C.S. and Chandrasekharan, M.P., 1990. Grouping efficacy—a quantitative criterion for goodness of block diagonal forms of binary matrices in Group Technology. *International Journal of Production research*, 28, 233–243.

Kumar, K.R. and Vannelli, A., 1987. Strategic subcontracting for efficient disaggregate manufacturing. *International Journal of Production Research*, 25, 1715–1728.

Kusiak, A., 1987. Generalized Group Technology concept. *International Journal of Production Research*, 25, 561–569.

Kusiak, A. and Chow, W.S., 1987. Efficient solving of the group technology problem. *Journal of Manufacturing Systems*, 6, 117–124.

Kusiak, A. and Cho, M., 1992. Similarity coefficient algorithms for solving the Group Technology problem. *International Journal of Production Research*, 30, 2633–2646.

Logendran, R., Ramakrishna, P. and Sriskandarajah, C., 1994. Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans. *International Journal of Production Research*, 32, 273–297.

McAuley, J., 1972. Machine grouping for efficiency production. *Product Engineering*, 51, 53–57.

McCormick, W.T., Schweitzer, P.J. and White, T.W., 1972. Problem decomposition and data reorganization by a clustering technique. *Operations Research*, 20, 993–1009.

Mitchell, M., 1996. *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.

Moon, C. and Gen, M., 1999. A Genetic Algorithm-based approach for design of independent manufacturing cells. *International Journal of Production economy*, 60, 421–426.

Pierreval, H., Caux, C., Paris, J.L. and Viguier, F., 2003. Evolutionary approaches to the design and organization of manufacturing systems. *Computers & Industrial Engineering*, 44, 339–364.

Plaquin, M.-F. and Pierreval, H., 2000. Cell formation using evolutionary algorithms with certain constraints. *International Journal of Production Economy*, 64, 267–278.

Rajagopalan, R. and Batra, J.L., 1975. Design of cellular production systems: a graph-theoretic approach. *International Journal of Production Research*, 13, 567–579.

Rao, H.A., Pham, S.N. and Gu, P., 1999. Genetic Algorithms-based approach for design of manufacturing systems: an industrial application. *International Journal of Production Research*, 37, 557–580.

Reeves, C.R., 1995. Genetic Algorithm for flowshop sequencing. *Computers & Operations Rcesearch*, 22, 5–13.

Sarker, B.R., 1996. Resemblance coefficients in Group Technology: a survey and comparative study of relational metrics. *Computers & Industrial Engineerings*, 30, 103–116.

Seifoddini, H., 1989. Note on the similarity coefficient method and the problem of improper machine assignment in group technology applications. *International Journal of Production Research*, 27, 1161–1165.

Seifoddini, H. and Wolfe, P.M., 1986. Application of the Similarity Coefficient Method in Group Technology. *IIE Transactions*, 18, 271–277.

Shafer, S.M. and Rogers, D.F., 1993a. Similarity and distance measures for cellular manufacturing. Part I. A survey. *International Journal of Production Research*, 31, 1133–1142.

Shafer, S.M. and Rogers, D.F., 1993b. Similarity and distance measures for cellular manufacturing. Part II. An extension and comparison. *International Journal of Production Research*, 31, 1315–1326.

Singh, N., 1993. Design of cellular manufacturing systems: an invited review. *European Journal of Operational Research*, 69, 284–291.

Sofianopoulou, S., 1999. Manufacturing cells design with alternative process plans and/or replicate machines. *International Journal of Production Research*, 37, 707–720.

Srinivasan, G., 1994. A clustering-algorithm for machine cell-formation in Group Technology using minimum spanning-trees. *International Journal of Production Research*, 32, 2149–2158.

Srinivasan, G., Narendran, T.T. and Mahadevan, B., 1990. An assignment model for the part-families problem in Group Technology. *International Journal of Production Research*, 28, 145–152.

Srinivasan, G. and Narendran, T.T., 1991. GRAFICS. A nonhierarchical clustering algorithm for Group Technology. *International Journal of Production Research*, 29, 463–478.

Tsai, C.-C., Chu, C.-H. and Barta, T.A., 1997. Modeling and analysis of a manufacturing cell formation problem with fuzzy mixed-integer programming. *IIE Transactions (Institute of Industrial Engineerings)*, 29, 533–547.

Uddin, M.K. and Shanker, K., 2002. Grouping of parts and machines in presence of alternative process routes by Genetic Algorithm. *International Journal of Production Economy*, 76, 219–228.

Venugopal, V. and Narendran, T.T., 1992. Genetic Algorithm approach to the machine–component grouping problem with multiple objectives. *Computers & Industrial Engineering*, 22, 469–480.

Waghodekar, P. H. and Sahu, S., 1984. Machine-component cell formation in group technology: MACE. *International Journal of Production Research*, 22, 937–948.

Wemmerlov, U. and Hyer, N.L., 1989. Cellular manufacturing in the US industry: a survey of users. *International Journal of Production Research*, 27, 1511–1530.

Wemmerlov, U. and Johnson, D.J., 1997. Cellular manufacturing at 46 user plants: implementation experiences and performance improvements. *International Journal of Production Research*, 35, 29–49.

Wicks, E.M. and Reasor, R.J., 1999. Designing cellular manufacturing systems with dynamic part populations. *IIE Transactions (Institute of Industrial Engineerings)*, 31, 11–20.

Won, Y., 2000. Two-phase approach to GT cell formation using efficient p-median formulations. *International Journal of Production Research*, 38, 1601–1613.

Yasuda, K., Hu, L. and Yin, Y., 2005. A Grouping Genetic Algorithm for the multi-objective cell formation problem. *International Journal of Production Research*, 43, 829–853.

Zolfaghari, S. and Liang, M., 2002. Comparative study of Simulated Annealing, Genetic Algorithms and Tabu search for solving binary and comprehensive machine-grouping problems. *International Journal of Production Research*, 40, 2141–2158.