CHAPTER 4

PROPOSED GROUPING GENETIC ALGORITHM

4.1 INTRODUCTION

The analysis of design features of parts, processing requirements, processing routes, machine types, machine capacity and other operational constraints and eventually identification of machine cells (MC) and part families (PF) is called the cell formation problem (CFP). Identification of machine-cells is one of the most important problems in the design of cellular manufacturing systems (CMS). The CFP is at the center of research in the CMS design. Cell formation consists of identifying machine groups and part families. Many cell formation methods use a part machine matrix as an input and attempt to obtain the block diagonalized form. However, perfect diagonalized matrix of parts and machines is not possible in many cases. The solution of this problem influences many aspects of the manufacturing environment such as material handling, production planning, delivery lead time, etc (Sing and Rajamani, 1996). In its simplistic form, when detailed manufacturing data such as production costs, processing times, production volumes, capacity of machines etc are not available during the design stage of the manufacturing system, then cell formation problem is modeled as a machine-part incidence matrix. In this matrix, each row corresponds to one part. Each machine cell is a set of machines denoted to a unique part family. Each part family is a set of similar parts that are generally to be processed in
one cell. Let $A = [a_{ij}]$ denote a machine–part incidence matrix. If $a_{ij}=1$, it means that part $i$ requires processing on machine $j$, otherwise $a_{ij} = 0$. As a tradition, ‘zeros’ are not shown in machine-part incidence matrix the solution approaches related to CFP aim to cluster all ‘ones’ in a block diagonal form so that the solution performs better on some criteria related to ‘goodness of solution’ such as grouping efficiency (Chandrasekharan and Rajagopalan, 1986b), grouping efficacy (Kumar and Chandrasekharan 1990), group technology efficiency (Harhalakis et al 1990), etc. Among the various PFA approaches, methodologies such as similarity coefficient-based methods, array-based methods, mathematical programming, graph theory, expert systems, neural networks, genetic algorithm, fuzzy set theory, etc have been attempted for the CFP. Machine/Part grouping problems have proven to be NP-complete and cannot be solved in polynomial time (Shanker and Agrawal 1997). Solving such problems of reasonable size often relies on heuristic approaches. Recently, several metaheuristic approaches have emerged as efficient tools for solving such problems. As discussed in chapter 2 in literature review, simulated annealing, neural networks, genetic algorithms, tabu search, fuzzy models, ant colony optimization are widely used approaches. Of these approaches, genetic algorithm based methods are among the widely attempted method to solve the CFP (Uddin and Shanker 2002). Because of its advantages over other methods, an attempt is made in this chapter to propose a grouping genetic algorithm for CFP.

Grouping genetic algorithms are suitable to solve the machine-cell formation problem. Mathematical formulations are developed and solve by method based on genetic algorithm. An objective consisting of a combined equal weightage of two objectives i.e. maximization of parts flow within the cells and minimization of intercellular movements is considered for this formulation. The number of machines, number of parts, upper and lower limit
of number of cells to be formed and volume of parts with their processing
times are given as input. The objectives of the research work are presented in
the previous chapter. This chapter provides the details of the mathematical
formulation and Pro-GGA for grouping method and device in the production
facilities. In this chapter, the results of numerical examples with comparative
studies are also discussed in detail.

4.2. A BRIEF INTRODUCTION TO GROUPING GENETIC
ALGORITHM

Group technology deals with the formation of the machine groups and
part families that make up the cells at a cellular manufacturing facility.
Specifically, the machine-part cell formation problem addresses the issues
surrounding the creation of part families based on component processing
requirements, and the identification of machine groups based on their ability to
process specific part families. When solving this problem, previous
researchers have concluded that solution methodologies for the MPCF
problem must focus attention on the block-diagonalization of the given
machine-part incidence matrix. The best solutions to MPCF problem are those
that contain a minimal number of voids (zeros in the diagonal blocks) and a
minimal number of exceptions (ones outside of the diagonal blocks). The
optimization of the cell formation (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 the problem size.
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 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). The parameter values and policies such as population size, number of generations, crossover rate, mutation rate, and replacement heuristic policy play a crucial role in successful implementation of grouping genetic algorithm. These are described in later sections.

4.3 PROBLEM DESCRIPTION

The cell formation problem in cellular manufacturing system is the identification of part families and machine cells. Given a set of parts and processing requirements on asset of machines, the aim of the cell formation problem is to obtain a satisfactory partition of parts into families and machines into cells, such that the resulting system performs well with respect to the design objectives. In this work, the machines are clustered into cells based on the objectives defines, and then the parts are assigned to part families, based on their maximum processing requirements of the part with respect to that the cell. Cell formation problem consists of non-linear objective functions with integer decision variables. 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). The applications of GA with different considerations are already discussed in chapter 2. From the study of review of literature, a grouping problem is identified. The study proposes a cell design model with the objective of maximizing the total number of parts processed within cells and minimizing intercellular movements considering the data of process plans for parts, production volume and cell size. Genetic algorithm approach is selected to solve the problem. To solve the defined problem a new grouping methodology is proposed. The crossover operator, mutation operator and replacement heuristic are used. The results are presented to compare proposed algorithm with results of other methods reported in literature. In the next section, the proposed methodology is discussed.
4.4 PROPOSED METHODOLOGY

The Proposed GGA (Pro-GGA) reported in this study is developed by improving the configuration of the standard GGA proposed by Brown and Sumichrast (2001). The Pro-GGA 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 Pro-GGA 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 Pro-GGA employs a repair process introduced by Tunnukij and Hicks (2009) that rectifies infeasible chromosomes that may be produced during the evolution process. There are several steps involved in the proposed grouping genetic algorithm (Pro-GGA). The main issues in developing a genetic algorithm are chromosome representation, initialization of the population, evaluation measure, crossover, and mutation and selection strategy. Also the genetic parameters such as population size, number of generations, probability of crossover (Pc), and probability of mutation (Pm) are determined before execution of genetic algorithm. In the following subsections, these issues and overall procedure are introduced and described.

4.4.1 Mathematical Formulations

One of the main advantages of GAs is that they only require an objective function (or ‘fitness function’) that can be evaluated numerically. The Pro-GGA uses the 0-1 machine-part incidence matrix to represent the initial configuration. Mathematical formulations are formulated as follows
\[ TG(n) = \sum_{i=1}^{N_p} (T_i \times N_i) \quad (4.1) \]

\[ FV(n) = \frac{1}{TG(n)} \quad (4.2) \]

\[ f(n) = R_{last} - R(n) \quad (4.3) \]

\[ P(n) = \frac{f(n)}{\sum_{i=1}^{N_p} f(i)} \quad (4.4) \]

\[ \sum_{k=1}^{i-1} P(k) \leq R_{nd} < f(n) \sum_{k=1}^{i} P(k) \quad (4.5) \]

where,

\[ i = 1, 2, \ldots, N_p \]

\[ n = 1, 2, \ldots, N_m \]

\[ k = 1, 2, \ldots, i \]

### 4.4.2 Notations Used

- \(T_i\): number of times which requires movement of the work between groups about part A(i)
- \(N_i\): The weight which responds to the quantity of number of parts A
TG (n) number of times TG of weighting movement (n)
FV fitness value
N_m Number of machines
N_u population
U gene arrangement
R(n) Ranking of the individual n
R_last Ranking of the lowest individual n
N_r Number of generations
f(n) selection dignity
P(n) selection probability

4.4.3 Assumptions

The model is based on the following assumptions:

A machine is allocated to only one cell. It is related with a grouping method distributing each machine to each group based on genetic information of an individual with optimal goodness of fit to a predetermined valuation basis in a final generation.

There is no empty cell.

The maximum number of cells needs to be specified by the user.

A part is assigned to a cell that contributes to the maximum cumulative processing time

It is preferred that upper limit and/or a lower limit are provided in the number of the machines which can be distributed to each group.
The genetic information of a parent which makes cross is set up at random.

4.4.4 Proposed Grouping Genetic Algorithm (Pro-GGA)

In this research work, it is proposed to minimize intercell flow and maximizing the total parts flow within cells in a consideration of the processing time, production requirements and available time on machine in a given period. The general structure of Pro-GGA is shown in Figure 4.1. There are several steps are involved in the proposed grouping genetic algorithm (Pro-GGA).

Step S1: Representation and initialization
   S1-1: Setting of initial value of gene information
   S1-2: Generation of initial population

Step S2: Goodness of fit (fitness value) calculation

Step S3: Operation
   S3-1: Selection
   S3-2: Cross Over
   S3-3: Mutation

Step S4: Generate a final generation.

Step S5: Solution derivation

The proposed grouping genetic algorithm (Pro-GGA) performs the same general procedures as all GAs, such as initialization, selection, crossover
Figure 4.1 A Generic framework for Pro-GGA
operation, mutation operation, etc. For the present study, in order to enforce the algorithm to function in an efficient way, efforts are made in several steps such as a) setting out of the initial value of genetic information, b) goodness of fit (fitness) calculation, c) improvement in genetic operators such as selection, crossover and mutation and d) termination condition with generating a \((N_{r+1})\) generation's (final generation). The proposed GGA with details of these operators and the settings provided them for the present study is discussed in the following steps.

**Step S1: Representation and initialization**

**S1-1: Setting out of the initial value of genetic information**

Genetic algorithm usually starts with an initial set of solutions called population and the population at a given time is a generation. Each individual in the population is called a chromosome. When designing Pro-GGA, we selected an encoding that its naturally with the MPCF problem. Representation plays a key role in the development of GA. A problem can be solved once it can be represented in the form of a solution string (chromosomes). The bits (genes) in the chromosome could be binary, real integer numbers or combination of characters. A number of representations like adjacency representation, permutation representation, ordinal representation and precedence matrix representation have been suggested. The machines cell-part grouping problem considered in this study. The chromosome representation is shown in Figure 4.2. It consists of three sections: the component section, the machine section, and the group section.
Note that the lengths of the part and machine sections are both constant for all chromosomes and depend on the number of parts and machines, respectively, in the problem. The group section, however, may vary in length, and depends upon the number of groups into which the components and machines are divided. To understand the encoding better, consider the problem of grouping 6 parts and 4 machines into cells. Under the assumption of no outside problem constraints, one solution to this problem is the chromosome 1 1 2 3 2 1 | 3 2 1 2 | 1 2 3. 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 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. 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.

S1-2: Generation of initial population

The next important step in GAs is to initialize the population, that is, create an initial population. The initialization process can be executed with either a randomly created population or a well-adapted (seeded) population. The initial population employed by Pro-GGA is created by first randomly generating values for each gene in the machine section of the chromosome. These values are constrained to be between 1 and $G$, where $G$ is the maximum number of groups/cells for the problem, $G \leq \min (M, P)$. After creation of the machine section, random values are then generated for the components section of the chromosome. These values are not only restricted to be between 1 and $G$, but must also come from the set of values assigned to the machine section. By assigning the initial values in this manner, all of the initial chromosomes are valid solutions for the MPCF problem. Because each value contained in the set $\{p_1, p_2, p_3, \ldots, p_P\}$ is also contained at least once in the set $\{m_1, m_2, m_3, \ldots, m_M\}$, there are no groups with zero components or zero machines in the initial population of chromosomes. By ensuring these problem constraints are satisfied, and assuming no other constraints exist, Pro-GGA begins with an initial feasible population.
Step S2: Goodness-of-fit (fitness value) calculation

In GAs, a fitness function value is computed for each string in the population and the objective is to find a string with the maximum fitness function value. To improve the solutions, each chromosome is evaluated using some measures of fitness. According to the formula 4.1, number-of-times TG of weighting movement \( n \) is computed.

\[
TG (n) = \sum_{i=1}^{N_p} (T_i \times N_i)
\]

The goodness of fit \( FV (n) \) is computed as a reciprocal of number-of-times TG of weighting movement \( n \) using formula 4.2.

\[
FV (n) = \frac{1}{TG (n)}
\]

Once Pro-GGA carries out rank-based roulette-wheel selection to determine the first parent, it repeats the process to determine the second parent. However, the second parent is not permitted to be a duplicate of the first. If this happens, the second parent is re-selected until no duplication occurs.

Step S3: Operation

The type of selection that Pro-GGA employs is termed rank based roulette wheel selection. The Falkenauer’s GGA crossover operator, elimination mutation operator and division mutation operator were used with minor modifications. The Pro-GGA includes a repair process that rectifies infeasible chromosomes that may be produced during the evolution process.
S3-1 Selection

A choice method of the individual (parent individuals) made to cross. Parent individuals are chosen with the technique of having combined ranking selection and roulette selection. Individual in the population are ranked from the high order with ranking according to their fitness value $FV(n)$. That is, the procedure so far corresponds to the technique of ranking selection according to formula 4.3. Set the selection probability $P(n)$ of the each object $n$ as the value proportional to selection dignity $f(n)$ using the formula 4.4. A random number $Rnd$ ($0 <= Rnd < 1$), is generated. The individual $n$ which satisfies the conditions of the formula 4.5 is chosen as parent individuals. It corresponds to the technique of roulette selection.

$$f(n) = R_{last} - R(n)$$

Here,

$R(n)$: Ranking of the individual $n$,
$R_{last}$: Ranking of the lowest individual $n$

First, the chromosomes are ranked in order from worst (rank of 1) to best (rank of $M$). An adjusted roulette wheel is then created based on the revised fitness scores, and selection follows the strategy described below.

Set the selection probability $P(n)$ of the each object $n$ as the value proportional to selection dignity $f(n)$ using the following formula 4.4.

$$P(n) = f(n) / \sum_{i=1}^{N_p} f(i)$$

Attach the turn $i$ ($i = 1, 2, --$) to the each object $n$. The individual $n$ of the turn $i$ which satisfies the conditions of the following formula 4.5 was...
chosen as parent individuals. About the random number Rnd which was made to generate the random number Rnd (Rnd: however, 0<=Rnd<1). It corresponds to the technique of roulette selection. How to attach the turn i of the each object n is optional.

\[
\sum_{k=1}^{i-1} P(k) \leq Rnd < f(n) / \sum_{k=1}^{i} P(k)
\]

For example, the individual 1, the individual 2, the individual 3, the individual 4, and the turn i are attached to the four individuals n, and suppose that selection probability P(i) of the i-th individual n is as follows respectively at this time.

\[(P(1), P(2), P(3), P(4)) = (0.2, 0.1, 0.4, 0.3)\]

At this time, the value of the random number Rnd and the relation with the individual n chosen are as follows.

(a) The individual 1 will be chosen if it is 0<=Rnd<0.2.
(b) The individual 2 will be chosen if it is 0.2<=Rnd<0.3 (=0.2+0.1).
(c) The individual 3 will be chosen if it is 0.3<=Rnd<0.7 (=0.3+0.4).
(d) The individual 4 will be chosen if it is 0.7<=Rnd<1.0 (=0.7+0.3).

The individual of the child which is formed (the individual of the next generation) with fixed probability, it selects one gene C of gene arrangement U to random, it mutates suddenly to modify that to random. It generates the random number of 1 or less of 0, or more if the probability where it can decide the value of that random number beforehand (mutation probability Pm) it is below. Choose the parent individuals of two bodies which repeat procedure of a more than, and make them cross. However, the individual selected as parents
of the 1st body is accepted, and is kept from turning into an individual with same parents of two bodies from the selected candidate of the 2nd body. Mutation probability \( P_m \) is set to usually very little value.

**S3-2: Cross Over**

The crossover operator of Pro-GGA is consistent with Falkenauer's GGA. Implementation details regarding the crossover rate and the heuristic employed in step 4 are developed after preliminary experimentation.

The two parents are selected using a rank-based roulette-wheel scheme. Two cross points for each parent is then determined randomly. Crossover occurs in the manner detailed by Falkenauer, with the inclusion of a problem-specific heuristic to adapt the chromosomes that result from the crossover operation. The entire process is best explained through the use of an example.

Two parents are randomly chosen from the population; two crossover points are then randomly selected from the group section of each parent. Notice that the parent one solution has divided the components and machines up into three cells, while the parent two solutions consists of only two cells. Chromosomes for Pro-GGA may be of variable length, as this example illustrates. Figure 4.3 shows two parents and their crossover points. 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
Figure 4.3  Falkenauer’s crossover operator. (a) select crossover points; (b) injection.
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 4.3(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 Pro-GGA 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 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.

**S3-3: Mutation**

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 4.4, 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 4.4(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 4.4(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. The chromosomes produced by the genetic operations may represent infeasible solutions.

The mutation operator explored in Pro-GGA consists of eliminating one of the groups of a chromosome. The group to eliminate is chosen randomly, but intelligence is built into the reassignment heuristic. Each of the displaced
components is placed into a new group using information obtained from the problem's original MP matrix. The placement is based on which existing group has the most machines that the displaced component requires. If a tie occurs, it is broken arbitrarily. Likewise, displaced machines are reassigned based on which existing group has the most components that need it. A repair process was developed to rectify infeasible chromosomes. The repair process consists of four steps.

Figure 4.4  Falkenauer’s mutation operators. (a) elimination; (b) division mutation.
1. Checking and removing empty cells. Each cell must contain at least one part and one machine. For example, in Figure 4.4(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 4.5(a)).

2. Checking the number of cells. The possible number of cells \( G \) is defined \( G \leq \min(M, P) \).

   (a) 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;
   (b) 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 4.5(a) illustrates this procedure. Child1 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 was
Figure 4.5  The Pro-GGA replacement heuristic processes.  
(a) remove the empty cells;  (b) relocate unassigned parts and / or machine by the Greedy Heuristic; (c) renumber the groups.
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 4.5(b) shows the solution after relocating unassigned parts and machines using the Greedy Heuristic. The replacement heuristic in the standard GGA would replace 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 Pro-GGA with the Greedy Heuristic.

4. Renumbering the groups to simplify interpretation. This is illustrated by Figure 4.5 (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.

**Step S4: Generate a final generation**

Number-of-times Nr repeats the alternation of generations by the procedure of the above-shown step S2 and Step S3, and generate a final generation over again. Here, the number of times Nr sets, as it asks for the
number of times whose goodness of fit of a best individual will not improve experimentally. Pro-GGA uses a generational replacement strategy. This means that with the exception of the best solution, the entire current population is replaced with offspring at the end of each generation. With Pro-GGA, the best chromosome from each generation is always passed to the next generation. The remainder of the chromosomes chosen to replace their predecessors is selected using the rank-based roulette-wheel strategy described in section S3-2-1.

**Step S5: Solution derivation**

After generating a final generation (Nr+1) group by number-of-times Nr repeating the procedure of the above-shown step S2 and Step S3, the individual n with the highest goodness of fit FV (n) is chosen as a solution in it. The most straightforward stopping criteria for a Pro-GGA is the number of generations. Unfortunately, there is no a priori method for determining how many generations is enough. However, as Radcliffe points out, using a population size of s with a GA that runs for t generations requires st evaluations. If st is not significantly less than N, where N is the size of the search space, then one may as well employ enumeration to solve the problem (Brown and Sumichrast 2001). For this reason, and also because preliminary tests indicate Pro-GGA is capable of finding high quality solutions quickly, the limit on the number of generations is set to 50. Pro-GGA monitors the quality of its solutions and may also stop once (Nr+1) consecutive generations pass by with no improvement. The Pro-GGA terminates when a fixed number of generations have been completed. The cell formation configuration associated with the highest fitness chromosome is then shown.
4.4.5 Pro-GGA Parameters

The Pro-GGA was tested using datasets from the literature. A full factorial experiment considered the parameter settings shown in Table 4.1. 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 Pro-GGA was tested with datasets 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.

**Table 4.1 Experimental parameter settings.**

<table>
<thead>
<tr>
<th>Parameter Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size ( N_u )</td>
</tr>
<tr>
<td>100 (data 1-9), 1000 (data 10-24)</td>
</tr>
<tr>
<td>Probability of crossover ( P_c )</td>
</tr>
<tr>
<td>0.6, 0.7, 0.8, 0.9, 1.0</td>
</tr>
<tr>
<td>Probability of mutation ( P_m )</td>
</tr>
<tr>
<td>0.1, 0.2, 0.3, 0.4</td>
</tr>
<tr>
<td>No. of generations</td>
</tr>
<tr>
<td>50</td>
</tr>
</tbody>
</table>

The values assigned to various genetic algorithm parameters can have a significant impact on the algorithm's performance. When designing Pro-GGA, we employ additional experimentation in order to determine appropriate values for the parameters of population size, crossover rate, and mutation rate. A small set of simulated problems is created by randomly generating MP matrices of varying size and sparsity. Experimentation is performed using three different population sizes (50, 100, 200), two potential crossover rates (1.0, 0.6), and two possible mutation rates (0.1, 0.3). For sample problems
with small MP matrices (10 x 10), most combinations of parameter values find the best solution in less than 10 generations. For larger problems (25 x 40, 25 x 100, 40 x 100), however, the combination of population size 100, crossover rate 0.9 and mutation rate 0.1 results in the highest quality solutions.

4.4.6 Evaluation methodology

The best solution produced by the machine-part incidence matrix-based methods minimizes 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 study employed the grouping efficiency and grouping efficacy as the objective function for measuring the quality of block diagonal forms.

4.4.6.1 Measures of effectiveness

The quality of a solution to the grouping problem depends on the degree of minimization in the volume of intercellular movements, which reduces the material handling cost, simplifies the shop control significantly, and reduces mean flow time and work in process (WIP) (Seifoddini and Wolf 1986). For this reason, a number of grouping measures have been developed to evaluate the efficiency of the machine part grouping.

4.4.6.2 Grouping efficiency

The cell formation problem (CFP) groups machines into machine cells and parts into part families (Hu and Yasuda 2006). Well designed
manufacturing cells should maximize the machine utilisation within each machine cell and minimize the inter-cell flow of parts.

Ballakur and Steudel (1987) identified three approaches to grouping employed by cell formation methods:

(i) part family grouping, which forms part families and then groups machines into cells;
(ii) machine grouping, which forms machine cells based upon similarities in part routings and then allocates parts to cells;
(iii) machine-part grouping, which forms part families and machine cells simultaneously.

The relationships between parts and machines may be represented as a machine-part incidence matrix (see Figure 4.6). For example, in Figure 4.6, 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. In order to gauge the performance of Pro-GGA, two measures of effectiveness are employed. The first, grouping efficiency is formally defined by Chandrasekharan and Rajagopalan. They develop this measure to provide a quantitative standard on a rational scale for comparing different solutions to the same problem (Chandrasekharan and Rajagopalan 1986a). For the MPCF problem, the quality of a solution depends on machine utilization and intercell movement.

Many solution techniques attempt to diagonalize the MP matrix in an effort to maximize machine utilization and minimize intercell traffic simultaneously. When a matrix representation of a solution has zeros on the
diagonal, this indicates a machine is not employed by one of the components in the group. When the solution matrix has ones of the diagonal, this represents intercell movement. Chandrasekharan and Rajagopalan (1986a) propose a measurement of efficiency, \( \eta \) that is based on these concepts.

(a)

(b)

Figure 4.6. A machine-part incidence matrix: (a) the original matrix; (b) a rearranged matrix into block-diagonal forms
The grouping efficiency, \( \eta \), is calculated as a weighted average of \( \eta_1 \) and \( \eta_2 \), using the following formula

\[
\eta = q \eta_1 + (1 - q) \eta_2
\]

where,

\[0 \leq q \leq 1\] and

\[
\eta_1 = \frac{e_d}{k} \sum_{r=1}^{M_r N_r}
\]

\[
\eta_2 = 1 - \frac{e_o}{k \cdot mn - \sum_{r=1}^{M_r N_r}}
\]

Where,

- \( e_d \) total number of ones in the diagonal blocks,
- \( e_o \) total number of ones in the off-diagonal blocks,
- \( k \) limiting number of groups,
- \( m \) number of machines (rows),
- \( n \) number of components (columns),
- \( M_r \) number of machines in the \( r^{th} \) cell,
- \( N_r \) number of components in the \( r^{th} \) cell.

The formula for \( \eta_1 \) expresses the ratio of the number of non-zero elements in the diagonal blocks to the total number of elements in the diagonal blocks. The
closer $\eta_1$ is to 1.0, the more likely that the machine utilization in the cell is close to 100%. The expression for $\eta_2$ represents the ratio of the number of zeros in the off-diagonal sections to the total number of elements in the off-diagonal sections. Values of $\eta_2$ that are close to 1.0 indicate minimal intercell traffic. Incorporation of the weighting factor $q$ enables the analyst to alter the emphasis between utilization and intercell movement, depending on the specific requirements of the given problem.

4.4.6.3 Group Efficacy

Among the algorithms that seek block diagonalization of the zero-one matrix are production flow analysis (Burbidge 1963), single linkage clustering (McAuley 1972), numerical taxonomy (Carrie 1973), rank order clustering (King and Nakornchai 1982), MODROC and ideal seed algorithm (Chandrasekharan and Rajagopalan 1986a, b) and ZODIAC (Chandrasekharan and Rajagopalan 1987). In addition there are many heuristic methods demonstrated for very small matrices and certain specific problems. Although researchers generally claim success in solving specific problems, such claims are not generally based on any quantitative criterion. Chandrasekharan and Rajagopalan (1986 b) have introduced a concept called grouping efficiency which is a quantitative criterion for comparing the block diagonal forms of zero-one matrices on a common scale. Kumar and Chandrasekharan (1990) developed another method as a quantitative criterion for measuring the quality of block diagonal forms. This measure has been widely used in the literature to a given MPCF problem. Their measure, termed grouping efficacy, considers the 'number of operations' to be the number of ones in the original MP matrix. With this in mind, $\Psi$ is defined as the ratio of exceptional elements to the
number of operations, and $\varphi$ is defined as the ratio of voids to the number of operations.

The formula for the group efficacy is given as:

$$\Gamma = \frac{1 - \Psi}{1 + \varphi} = \frac{1 - e_v/e}{1 + e_o/e} = \frac{e - e_o}{e + e_v} = \frac{e_o + e_v}{e + e_v}$$

where

- $e$ - number of operations,
- $e_v$ - number of voids,
- $e_o$ - number of exceptions.

Both the above two measures are used to evaluate the Pro-GGA.

Whereas the formula for grouping efficiency allowed for equal weighting of the ratio involving exceptional elements and the ratio involving voids (by setting $q = 0.5$), the formula for grouping efficacy does not allow for this. 'As the efficacy function is not symmetric with references to $\Psi$ and $\varphi$, the influence of exceptions and voids are not identical' (Kumar and Chandrasekharan 1990). This is an intentional variation from the efficiency formula because these authors believe that, in real life, exceptional elements are much more significant than voids.
4.4.7 Numerical Study

The proposed GGA was compiled in C language and the numeric examples were tested on PC. Pro-GGA was tested with datasets of two simple examples that have been published in the literature.

4.4.7.1 Analysis of performance using data obtained from the literature

A full factorial experiment considered the parameter settings shown in Table 4.1. In this research, the sum of probabilities of crossover ($P_c$) and mutation ($P_m$) was defined as $P_c + P_m \leq 1$. The Pro-GGA was tested with a set of 9 datasets that have been published in the literature and have been widely used in many comparative studies. All the data sets were transcribed form the original articles.

4.5 RESULTS AND DISCUSSION

The purpose of the experimental study is to determine how the proposed grouping genetic algorithm performs under various circumstances. The proposed GGA was compiled in C language and the numeric examples were tested on PC.

In order to demonstrate the proposed method, a simple problem was introduced and then the result is certified from previous literatures that have been published. By comparison with benchmark examples, it showed that the proposed method is capable of dealing with grouping devices and methods
more efficient and flexible. Numerical examples have been provided to verify the approach proposed in previous sections.

4.5.1 Sample Example 1

In order to demonstrate the proposed method, a simple problem was introduced and then the result is certified from Yasuda et al (2005). The first example is a CFP with five machines and eight parts, and the result after cell formation is shown in Figure 4.7. To solve the problem using the proposed genetic algorithm, the genetic parameters are set as follows: maximum generation, $N_r = N_m$, population size, $N_u = 20$; crossover probability, $P_c = 0.6$; mutation probability, $P_m = 0.2$. From the result of the obtained chromosome, it implies that machines 1, 4 and 5, parts 1, 4 and 7 are grouped into the same cell and machines 2 and 3, parts 2, 3, 5, 6 and 8 are grouped into another cell, and the total number of cells is two for the optimal solution. The results are summarized in Table 4.2.

Efficiency calculation:

In this example, the block diagonal matrix has no exceptional elements and no voids. Thus the both efficiency and efficacy are 100%

4.5.2 Sample Example 2

The second problem with 8 machines and 20 parts was introduced and then the result is certified from Nair and Narendran (1998). The machine part incidence matrix is given in Figure 4.8. To solve the problem using the proposed genetic algorithm, the genetic parameters are set as follows: maximum generation, $N_r = N_m$, population size, $N_u = 200$; crossover
Figure 4.7  5 x 8 CFP before and after grouped – Sample Example 1

Table 4.2  Group configuration with their corresponding products and Machines –Sample Example 1

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,4,5</td>
<td>1,4,7</td>
</tr>
<tr>
<td>2</td>
<td>2,3</td>
<td>2,3,5,6,8</td>
</tr>
</tbody>
</table>
probability, $P_c = 0.5$; mutation probability, $P_m = 0.3$. For the case of 3 groups with the group number $N_g$ is equal to 4, the best chromosomes is obtained by proceeding the Pro-GGA. From the result of the obtained chromosome, it implies that machines 1, 2 and 5 are assigned to cell 1, machines 3, 6, 7 and 8 to cell 2 and machines 4 to cell 3, respectively. The results are summarized in Table 4.3.

**Calculation:**

(i) The grouping efficiency, $\eta$, is calculated as follows;

$$e_d = 51; e_0 = 10; e = 61; e_v = 1; q = 0.5;$$

$$\eta = q \eta_1 + (1-q) \eta_2$$

where $0 \leq q \leq 1$, and

$$\eta_1 = \frac{e_d}{\sum_{r=1}^{M_r N_r} M_r N_r} = \frac{51}{52} = 0.98077$$

$$\eta_2 = 1 - \frac{e_0}{\sum_{r=1}^{M_r N_r} M_r N_r} = 1 - \frac{10}{160 - 52} = 0.90741$$

$$\eta = 0.5 \times 0.98077 + (1-0.5) \times 0.90741 = 0.94409$$

$$= 94.409 \%$$
(ii) The grouping efficacy \( \Gamma \), is calculated as follows;

\[
\Gamma = \frac{1 - \Psi}{1 + \Psi} = \frac{1 - e_0/e}{1 + e_0/e} = \frac{e - e_0}{e + e_0} = 1 - \frac{e_0 + e_v}{e_0 + e_v}
\]

\[
\frac{61 - 10}{61 + 1} = \frac{51}{62} = 0.8225806
\]

4.5.3 Comparative Studies

In order to test the applicability of Pro-GGA for large-scale problems, it was tested with the datasets of benchmark examples from the published literature. However, all these problems are only flow charts of 0-1 matrices denoting the manufacturing relationship between machines and parts. Other important design factors such as processing time, production requirements, and available time on machine in a given period are not considered. For a comparison and an ability test of Pro-GGA, it is assumed that all \( w_{ij} \) of workloads on machine \( i \) induced by part \( j \) are 1 s. Using this assumption, the mathematical model defined in the proposed methodology can be adopted for the machine-part flow charts of 0-1 matrices with a number of additional iterations. A full factorial experiment considered the parameter settings shown in Table 4.1. In this research, the sum of probabilities of crossover \( (P_c) \) and mutation \( (P_m) \) was defined as \( P_c + P_m \leq 1 \). As the final group formation is obtained by Pro-GGA method in the form of gene sequence, the results are summarized in tabular form and then grouping efficiency and grouping efficacy are measured, which are usually introduced when comparing examples.
Figure 4.8 Machine part incidence matrix for 8 x 20 CFP before and after grouped – Sample Example 2

Table 4.3 Group configuration with their corresponding Parts and Machines – Sample Example 2

<table>
<thead>
<tr>
<th>Groups</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3, 1</td>
<td>2, 8, 9, 11, 13, 14, 16, 17, 19</td>
</tr>
<tr>
<td>2</td>
<td>4, 7, 8, 2</td>
<td>3, 4, 6, 7, 18, 20</td>
</tr>
<tr>
<td>3</td>
<td>5, 6</td>
<td>1, 5, 10, 12, 15</td>
</tr>
</tbody>
</table>
The details of the data set of 9 benchmark problems along with the results and calculation are presented here as follows:

4.5.3.1 Data set 1

KN1: 5 x 7 (King and Nakronchai 1982)

![Machine part incidence matrix for 5 x 7 CFP before and after grouped – Data set 1](image)

(a) 5x7 CFP before grouped

(b) 5x7 CFP before grouped

Figure 4.9 Machine part incidence matrix for 5 x 7 CFP before and after grouped – Data set 1
Table 4.4  Group configuration with their corresponding Parts and Machines – Data set 1

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3 ,2,5</td>
<td>1,3,7</td>
</tr>
<tr>
<td>2</td>
<td>1,4</td>
<td>2,4,6,5</td>
</tr>
</tbody>
</table>

Calculation:

(i) The grouping efficiency, $\eta$, is calculated as follows;

$$e_d = 14;\ e_0 = 0;\ e = 14;\ e_v = 3;\ q = 0.5;$$

$$\eta = q \eta_1 + (1 - q) \eta_2$$

where $0 \leq q \leq 1$, and

$$\eta_1 = \frac{e_d}{k} = \frac{14}{17} = 0.8235$$

$$\eta_2 = 1 - \frac{e_0}{k} = 1 - \frac{0}{35 - 17} = 1$$
(ii) The grouping efficacy \( \Gamma \), is calculated as follows;

\[
\Gamma = \frac{1 - \Psi}{1 + \frac{e_o}{e}} = \frac{1 - e_o/e}{1 + e_o/e} = \frac{e - e_o}{e + e_o} = 1 - \frac{e_o + e_v}{e + e_v}
\]

\[
= \frac{14 - 0}{14 + 3}
\]

\[
= 0.8235294
\]

4.5.3.2 Data set 2

BC1  20 x 35  (Boe and Cheng 1991)

Table 4.5  Group configuration with their corresponding parts and machines – Data set 2

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7,1,11,16,19,15,12</td>
<td>30,32,6,34,35,4,9,11,21,28,33</td>
</tr>
<tr>
<td>2</td>
<td>8,17,3</td>
<td>1,5,15,17,20,25,3,29</td>
</tr>
<tr>
<td>3</td>
<td>2,14,18,4,13</td>
<td>31,2,10,12,13,18,24,27,7</td>
</tr>
<tr>
<td>4</td>
<td>9,20,10,6,5</td>
<td>8,14,19,23,26,16,22</td>
</tr>
</tbody>
</table>
Calculation:

(i) The grouping efficiency, $\eta$, is calculated as follows;

$$e_d = 123; \; e_0 = 30; \; e = 153; \; e_v = 58; \; q = 0.5;$$

$$\eta = q \eta_1 + (1 - q) \eta_2$$

where $0 \leq q \leq 1$, and

$$\eta_1 = \frac{e_d}{\sum_{r=1}^k M_r N_r} = \frac{123}{181} = 0.67956$$

$$\eta_2 = 1 - \frac{e_0}{\sum_{r=1}^k M_r N_r} = 1 - \frac{30}{700 - 181} = 0.9421965$$

(ii) The grouping efficacy $\Gamma$, is calculated as follows;

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

$$\Gamma = \frac{153 - 30}{153 + 58} = \frac{0.58294}{153 + 58} = 58.294\%$$
### 4.5.3.3 Data set 3

SR1 16 x 30 (Srinivasan et al 1990)

<table>
<thead>
<tr>
<th>P</th>
<th>M</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) 16x30 CFP after grouped

<table>
<thead>
<tr>
<th>P</th>
<th>M</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) 16x30 CFP after grouped

Figure 4.10. Machine part incidence matrix for 16 x 30 CFP (a) before grouped and (b) after grouped – Data set 3
Table 4.6 Group configuration with their corresponding parts and machines – Data set 3

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1, 4, 7, 8, 11, 12</td>
<td>2, 4, 7, 9, 12, 18, 22, 30</td>
</tr>
<tr>
<td>2</td>
<td>2, 13</td>
<td>1, 3, 10, 16, 17, 20</td>
</tr>
<tr>
<td>3</td>
<td>3, 6, 9, 15</td>
<td>5, 13, 19, 23, 25, 27, 28, 29</td>
</tr>
<tr>
<td>4</td>
<td>5, 10, 14, 16</td>
<td>6, 8, 11, 14, 15, 21, 24, 26</td>
</tr>
</tbody>
</table>

Calculation:

(i) The grouping efficiency, $\eta$, is calculated as follows;

\[
e_d = 98; \quad e_0 = 18; \quad e = 116; \quad e_v = 26; \quad q = 0.5;
\]

\[
\eta = q \eta_1 + (1 - q) \eta_2
\]

where $0 \leq q \leq 1$, and

\[
\eta_1 = \frac{e_d}{\sum_{r=1}^{M} \sum_{N_r} M_r N_r} = \frac{98}{124} = 0.79033
\]

\[
\eta_2 = 1 - \frac{e_0}{\sum_{r=1}^{M} \sum_{N_r} M_r N_r} = 1 - \frac{18}{480 - 124} = 0.94944
\]

\[
\eta = 0.5 \times 0.79033 + (1 - 0.5) \times 0.94944
\]

\[
= 0.869884
\]
(ii) The grouping efficacy $\Gamma$, is calculated as follows;

$$\Gamma = \frac{1 - \Psi}{1 + \sigma/e} = \frac{1 - e_o/e}{1 + e/o/e} = \frac{e - e_o}{e + e_o} = 1 - \frac{e_o + e_v}{e + e_v}$$

$$= \frac{116 - 18}{116 + 26} = \frac{98}{142} = 0.690141$$

4.5.3.4 Data set 4

CR1 24X 40 (Chandrasekharan and Rajagopalan 1989)

Table 4.7 Group configuration with their corresponding parts and machines – Data set 4

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,13,21,22</td>
<td>1,9,16,17,33</td>
</tr>
<tr>
<td>2</td>
<td>2,5,11,19</td>
<td>10,13,14,22,35,36</td>
</tr>
<tr>
<td>3</td>
<td>3,20</td>
<td>2,11,12,15,23,24,31,34</td>
</tr>
<tr>
<td>4</td>
<td>4,16</td>
<td>8,19,21,28,37,38,39</td>
</tr>
<tr>
<td>5</td>
<td>6,8,12,15,18</td>
<td>4,5,18,26,27,30</td>
</tr>
<tr>
<td>6</td>
<td>7,14,23,24</td>
<td>3,25,32</td>
</tr>
<tr>
<td>7</td>
<td>9,10,17</td>
<td>6,7,20,29,40</td>
</tr>
</tbody>
</table>
Calculation:

(i) The grouping efficiency, \( \eta \), is calculated as follows;

\[
e_d = 131; \ e_0 = 0; \ e = 131; \ e_v = 0; \ q = 0.5;
\]

\[
\eta = q \eta_1 + (1 - q) \eta_2
\]

where \( 0 \leq q \leq 1 \), and

\[
\eta_1 = \frac{e_d}{\sum_{r=1}^{k} \frac{M_r N_r}{131}} = \frac{130}{131} = 1
\]

\[
\eta_2 = 1 - \frac{e_0}{\sum_{r=1}^{k} \frac{M_r N_r}{960 - 131}} = 1 - \frac{0}{960 - 131} = 1
\]

\[
\eta = 0.5 \times 1 + (1-0.5) \times 1 = 1 = 100\%
\]

(ii) The grouping efficacy \( \Gamma \), is calculated as follows;

\[
\Gamma = \frac{1 - \Psi}{1 + \varphi} = \frac{1 - e_o/e}{1 + e_v/e} = \frac{e_0/e}{e + e_v} = \frac{e_o + e_v}{e + e_v}
\]

\[
= \frac{1}{\frac{131}{131+0}} = 1
\]
4.5.3.5 Data set 5

CR2  24 X 40 (Chandrasekharan & Rajagopalan 1989)

Table 4.8  Group configuration with their corresponding parts and machines – Data set 5

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,13,21,22</td>
<td>1,9,16,17,33</td>
</tr>
<tr>
<td>2</td>
<td>3,20</td>
<td>2,11,12,15,23,24,31,34</td>
</tr>
<tr>
<td>3</td>
<td>4,16</td>
<td>8,19,21,28,37,38,39</td>
</tr>
<tr>
<td>4</td>
<td>2,5,11,19</td>
<td>10,13,14,22,35,36</td>
</tr>
<tr>
<td>5</td>
<td>6,8,12,15,18</td>
<td>4,5,18,26,27,30</td>
</tr>
<tr>
<td>6</td>
<td>7,14,23,24</td>
<td>3,25,32</td>
</tr>
<tr>
<td>7</td>
<td>9,10,17</td>
<td>6,7,20,29,40</td>
</tr>
</tbody>
</table>

Calculation:

(i) The grouping efficiency, $\eta_1$, is calculated as follows;

$$e_d = 120; \quad e_0 = 10; \quad e = 130; \quad e_v = 11; \quad q = 0.5;$$

$$\eta_1 = q \eta_1 + (1 - q) \eta_2$$

where $0 \leq q \leq 1$, and

$$\eta_1 = \frac{e_d}{\sum_{r=1}^{k} M r N r} \quad \eta_1 = \frac{120}{131} \quad \eta_1 = 0.916031$$
\[
\eta_2 = 1 - \frac{e_o}{\sum_{r=1}^{k} M r N r} = 1 - \frac{10}{960 - 131} = 0.98788
\]

\[
\eta = 0.5 \times 0.916031 + (1-0.5) \times 0.98788
\]

\[
= 0.98788
\]

\[
= 951955 \%
\]

(ii) The grouping efficacy \( \Gamma \), is calculated as follows;

\[
\Gamma = \frac{1 - \Psi}{1 + \psi} = \frac{1 - e_o/e}{1 + e_o/e} = \frac{e - e_o}{e + e_o} = \frac{e_o + e_v}{e + e_v}
\]

\( e \) - number of operations,

\[
130 - 10
\]

\[
= \frac{130 - 10}{130 + 11}
\]

\[
= 0.851064
\]
4.5.3.6 Data set 6

CR3 24 X 40 (Chandrasekharan & Rajagopalan 1989)

Table 4.9 Group configuration with their corresponding parts and machines – Data set 6

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,13,21,22</td>
<td>1,9,16,17,33</td>
</tr>
<tr>
<td>2</td>
<td>2,5,11,19</td>
<td>10,13,14,22,35,36</td>
</tr>
<tr>
<td>3</td>
<td>3,20</td>
<td>2,11,12,15,23,24,34</td>
</tr>
<tr>
<td>4</td>
<td>4,16</td>
<td>8,19,21,28,37,38,39</td>
</tr>
<tr>
<td>5</td>
<td>6,8,12,15,18</td>
<td>4,5,18,26,27,30</td>
</tr>
<tr>
<td>6</td>
<td>7,14,23,24</td>
<td>3,25,32</td>
</tr>
<tr>
<td>7</td>
<td>9,10,17</td>
<td>6,7,20,29,40,31</td>
</tr>
</tbody>
</table>

Calculation:

(i) The grouping efficiency, $\eta_1$, is calculated as follows;

$$ e_d = 112; \quad e_0 = 20; \quad e = 132; \quad e_v = 20; \quad q = 0.5; $$

$$ \eta = q \eta_1 + (1 - q) \eta_2 $$

where $0 \leq q \leq 1$, and

$$ \eta_1 = \frac{e_d}{\sum_{r=1}^{k} M_r N_r} = \frac{112}{132} = 0.8485 $$
\[ \eta_2 = 1 - \frac{e_o}{\kappa} = 1 - \frac{20}{960 - 132} = 0.9758 \]

\[ \eta = 0.5 \times 0.8485 + (1 - 0.5) \times 0.9758 \]
\[ = 0.91215 \]
\[ = 91.215\% \]

(ii) The grouping efficacy \( \Gamma \), is calculated as follows;

\[ \Gamma = \frac{1 - \Psi}{1 + \phi} = \frac{1 - e_o/e}{1 + e_v/e} = \frac{e - e_o}{e + e_v} = 1 - \frac{e_o + e_v}{e + e_v} \]
\[ = \frac{132 - 20}{132 + 20} \]
\[ = 0.736842 \]
\[ = 73.6842\% \]
4.5.3.7 Data set 7

CR5 24X 40 (Chandrasekharan & Rajagopalan 1989)

Table 4.10 Group configuration with their corresponding parts and machines – Data set 7

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,21,2,19,23</td>
<td>9,33,3,13,14,36</td>
</tr>
<tr>
<td>2</td>
<td>3,20</td>
<td>2,12,15,23,34,31</td>
</tr>
<tr>
<td>3</td>
<td>4,16</td>
<td>8,19,21,28,37,38,39</td>
</tr>
<tr>
<td>4</td>
<td>5,11</td>
<td>10,22,35</td>
</tr>
<tr>
<td>5</td>
<td>7,9,14</td>
<td>40,29,32</td>
</tr>
<tr>
<td>6</td>
<td>12,18,6,10,15,8</td>
<td>18,37,6,24,27,26,20,39,4,5,30</td>
</tr>
<tr>
<td>7</td>
<td>13,22,17,24</td>
<td>1,16,7,17</td>
</tr>
</tbody>
</table>

Calculation:

(i) The grouping efficiency, $\eta$, is calculated as follows;

$$e_d = 101; \ e_0 = 30; \ e = 131; \ \ e_v = 53; \ q = 0.5;$$

$$\eta = q \eta_1 + (1 - q) \eta_2$$

where $0 \leq q \leq 1$, and

$$\eta_1 = \frac{e_d}{\sum_{r=1}^{k} M_r N_r} = \frac{101}{154} = 0.655844$$
\[
\eta_2 = 1 - \frac{e_o}{k} = 1 - \frac{30}{960 - 154} = 0.9627792
\]

(ii) The grouping efficacy \( \Gamma \), is calculated as follows;

\[
\Gamma = \frac{1 - \Psi}{1 + \psi} = \frac{1 - \frac{e_o}{e}}{1 + \frac{e_o}{e}} = \frac{e - e_o}{e + e_o} = 1 - \frac{e_o + e_v}{e + e_v}
\]

\[
\frac{131 - 30}{131 + 53} = 0.5489 = 54.89 \%
\]

**4.5.3.8 Data set 8**

CR6 24X 40 (Chandrasekharan & Rajagopalan 1989)

Table 4.11  Group configuration with their corresponding parts and machines – Data set 8

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20,3,11,24</td>
<td>2,15,23,34</td>
</tr>
<tr>
<td>2</td>
<td>1,17</td>
<td>17,31</td>
</tr>
<tr>
<td>3</td>
<td>2,5,19</td>
<td>13,14,22,35,36,9,5,10,13,7,29</td>
</tr>
<tr>
<td>4</td>
<td>9,10,8,6</td>
<td>40,6,39,26,30</td>
</tr>
<tr>
<td>5</td>
<td>21,15,22,13,14,23</td>
<td>1,12,16,3</td>
</tr>
<tr>
<td>6</td>
<td>12,18</td>
<td>20,24,27,4,18</td>
</tr>
<tr>
<td>7</td>
<td>7,4,16</td>
<td>25,38,11,8,19,21,28,32,37</td>
</tr>
</tbody>
</table>
Calculation:

(i) The grouping efficiency, $\eta_1$, is calculated as follows;

$$e_d = 97; \quad e_0 = 39; \quad e = 136; \quad e_v = 66; \quad q = 0.5;$$

$$\eta_1 = q \eta_1 + (1 - q) \eta_2$$

where $0 \leq q \leq 1$, and

$$\eta_1 = \frac{e_d}{\sum_{r=1}^{M_rNr}} = \frac{97}{134} = 0.72388$$

$$\eta_2 = 1 - \frac{e_o}{\sum_{r=1}^{M_rNr}} = 1 - \frac{39}{960 - 134} = 0.9528$$

$$\eta = 0.5 \times 0.72388 + (1-0.5) \times 0.9528$$

$$= 0.8384 = 83.84\%$$

(ii) The grouping efficacy $\Gamma$, is calculated as follows;

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

$$= \frac{136 - 39}{136 + 66} = 0.4802 = 48.02\%$$
4.5.3.9 Data set 9

CR7 24X 40 (Chandrasekharan & Rajagopalan 1989)

Table 4.12  Group configuration with their corresponding parts and machines – Data set 9

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,21</td>
<td>9,33,17,1</td>
</tr>
<tr>
<td>2</td>
<td>2,19</td>
<td>13,14,40,36</td>
</tr>
<tr>
<td>3</td>
<td>11,5</td>
<td>10,22,23,15</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>35,8,11,21,38,6</td>
</tr>
<tr>
<td>5</td>
<td>10,18,12</td>
<td>5,24,4,18,27,20</td>
</tr>
<tr>
<td>6</td>
<td>7,20,17,16,24,15</td>
<td>28,2,25,32,7,29,39,37,31,19</td>
</tr>
<tr>
<td>7</td>
<td>8,6,9,14,22,3,13,23</td>
<td>26,30,16,34,12,3</td>
</tr>
</tbody>
</table>

Calculation:

(i) The grouping efficiency, $\eta$, is calculated as follows;

$e_d = 96; \quad e_0 = 35; \quad e = 131; \quad e_v = 52; \quad q = 0.5; \quad \eta = q \eta_1 + (1 - q) \eta_2$

$\eta |_{\text{M r Nr}} = \frac{e_d}{k} \sum_{r=1}^{M r Nr} 96 = \frac{146}{0.657534}$

where $0 \leq q \leq 1$, and
\[ r_2 = 1 - \frac{e_o}{\sum_{r=1}^{k} M_r N_r} = 1 - \frac{35}{960 - 146} = 0.957 \]

\[ r_1 = 0.5 \times 0.657534 + (1-0.5) \times 0.957 \]

\[ = 0.807266 = 80.7266 \% \]

(ii) The grouping efficacy \( \Gamma \), is calculated as follows;

\[ \Gamma = \frac{1 - \Psi}{1 + \phi} = \frac{1 - e_o/e}{e + e_v} = \frac{e - e_o}{e + e_v} = 1 - \frac{e_o + e_v}{e + e_v} \]

\[ = \frac{131 - 35}{131 + 52} = 0.52459 \]

\[ = \frac{0.957}{146} \]

4.5.4 Application of a Pro-GGA to MPCF problems from the literature

Initially, Pro-GGA is applied to a small set of randomly generated problems so that fine-tuning of algorithmic parameters can occur. Pro-GGA’s success on these few problems from the literature provides confidence that the algorithm can perform well across a spectrum of example MPCF problems. Six data sets are taken from an article by Chandrasekharan and Rajagopalan (1989). For each of these data sets, the authors report the best grouping efficiency that their algorithm ZODIAC (Chandrasekharan and Rajagopalan 1987) obtains. Pro-GGA achieves an efficiency score of 100% for the first data set, and this matches the score obtained by Chandrasekharan and
Rajagopalan (1987). For the remaining five data sets, however, Pro-GGA outperforms ZODIAC in all cases. The efficiency scores reported for ZODIAC and those obtained with Pro-GGA are given in Table 4.13. In all cases, Pro-GGA obtains its best grouping efficiency score in 25 generations or less. With grouping efficiency as the measure of effectiveness, Pro-GGA is very successful, obtaining solutions that are as good as, or equal, than those presented in the literature. To test the algorithm further, the objective function measure is changed to grouping efficacy, but no other changes are made to the algorithm or its parameters. The Pro-GGA was compared with the other methods from the literature listed below:

(i) ZODIAC (Chandrasekharan and Rajagopalan 1989);
(ii) GRAFICS (Srinivasan and Narendran 1991);
(iii) CF-GGA (Brown and Sumichrast 2001);
(iv) Joines et al (1996b)
(v) EA-GA (Goncalves and Resende 2004);
(vi) HGGA (James et al. 2007);
(vii) EnGGA (Tunnukij and Hicks 2009).

Table 4.13 Grouping efficiency comparison of ZODIAC and PRO-GGA.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Size</th>
<th>ZODIAC %</th>
<th>PRO-GGA %</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR1</td>
<td>24 x 40</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>CR2</td>
<td>24 x 40</td>
<td>95.20</td>
<td>95.20</td>
</tr>
<tr>
<td>CR3</td>
<td>24 x 40</td>
<td>91.14</td>
<td>97.58</td>
</tr>
<tr>
<td>CR5</td>
<td>24 x 40</td>
<td>77.31</td>
<td>80.93</td>
</tr>
<tr>
<td>CR6</td>
<td>24 x 40</td>
<td>72.43</td>
<td>83.84</td>
</tr>
<tr>
<td>CR7</td>
<td>24 x 40</td>
<td>69.33</td>
<td>80.72</td>
</tr>
</tbody>
</table>
The results of ZODIAC and GRAFICS were obtained from Srinivasan and Narendran (1991); others were obtained from Tunnukij and Hicks (2009) and Brown and Sumichrast 2001). Table 4.14 presents the results for one data set (KN1) from King and Nakornchai (1982), one data set (SR1) from Srinivasan et al (1990), one data set from (BC1) Boe and Cheng (1991) and 6 data sets (CR1, CR2, CR3, CR5, CR6 and CR7) from (Chandrasekharan and Rajagopalan 1989).

4.5.5 Analysis of performance

Using the grouping efficiency measure, Pro-GGA outperforms ZODIAC on five of the six data sets, and matches its performance on the sixth. The average percentage improvement of Pro-GGA over ZODIAC is 5%. The solutions created by Pro-GGA using efficiency as the objective function were analysed to gain an understanding of why they were superior to solutions created by ZODIAC. In terms of the grouping efficiency measure, the Pro-GGA produced results that were equal to, or better than, all the other methods. With the efficiency measure, voids and exceptional elements are weighted equally by setting \( q \) much smaller than the proportion where exceptions may occur, in most situations. Pro-GGA realizes this and finds optimal solutions by creating \( \eta_1 \) value very close to 1.0. It does this by ensuring the diagonal blocks contain as many ones as possible, without worrying about an increase in exceptional elements. Since the off-diagonal portion of the MP matrix is generally much larger than the diagonal portion, the number of exceptions will be divided by a larger denominator. This means that a single exceptional element has less of an impact on the efficiency rating than does a single void.
Kumar and Chandrasekharan recognized this shortcoming of the efficiency measurement and created grouping efficacy to gauge more accurately the solution quality for MPCF problems. With Pro-GGA, the traditional grouping genetic algorithm crossover is the only operator utilized, indicating the potential robustness of Pro-GGA. The previous methods such as 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. The other methods such as CF-GGA, HGGA and EnGGA all allowed singletons. Pro-GGA also allowed singleton. Pro-GGA is success in achieving high quality solutions so quickly can be attributed, in part, to the intelligence built in to the replacement

Table 4.14 Grouping Efficacy comparisons with benchmark examples

<table>
<thead>
<tr>
<th>No.</th>
<th>Dataset</th>
<th>Size</th>
<th>ZODIAC</th>
<th>GRAFICS</th>
<th>CF-GGA Joines</th>
<th>EA-GA</th>
<th>HGGA</th>
<th>EnGGA</th>
<th>Pro-GGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KN1</td>
<td>5X7</td>
<td>73.68</td>
<td>73.68</td>
<td>-</td>
<td>73.68b</td>
<td>82.35</td>
<td>82.35</td>
<td>82.35</td>
</tr>
<tr>
<td>2</td>
<td>SR1</td>
<td>16X30</td>
<td>67.83</td>
<td>67.83</td>
<td>-</td>
<td>67.83b</td>
<td>68.99</td>
<td>68.99a</td>
<td>69.01a</td>
</tr>
<tr>
<td>3</td>
<td>BC1</td>
<td>20X35</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>58.07b</td>
<td>57.98</td>
<td>57.98a</td>
<td>58.29a</td>
</tr>
<tr>
<td>4</td>
<td>CR1</td>
<td>24X40</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>CR2</td>
<td>24X40</td>
<td>85.11</td>
<td>85.11</td>
<td>85.11</td>
<td>85.11</td>
<td>85.11</td>
<td>85.11</td>
<td>85.11</td>
</tr>
<tr>
<td>6</td>
<td>CR3</td>
<td>24X40</td>
<td>73.51</td>
<td>73.51</td>
<td>73.29</td>
<td>73.51</td>
<td>73.51</td>
<td>73.51</td>
<td>73.51</td>
</tr>
<tr>
<td>7</td>
<td>CR5</td>
<td>24X40</td>
<td>20.42</td>
<td>43.27</td>
<td>48.98</td>
<td>-</td>
<td>51.88</td>
<td>53.29</td>
<td>53.29a</td>
</tr>
<tr>
<td>8</td>
<td>CR6</td>
<td>24X40</td>
<td>18.23</td>
<td>44.51</td>
<td>46.81</td>
<td>-</td>
<td>46.69</td>
<td>48.95</td>
<td>48.95a</td>
</tr>
<tr>
<td>9</td>
<td>CR7</td>
<td>24X40</td>
<td>17.61</td>
<td>41.67</td>
<td>44.14</td>
<td>-</td>
<td>44.75</td>
<td>47.26</td>
<td>46.58a</td>
</tr>
</tbody>
</table>

a Solutions where singletons appear.
b Data reported in Gonclaves and Resende (2004) was inconsistent with data in the original reference.
c Value reported in James et al. (2007) was inconsistent with value calculated from the block diagonal matrix for the solution.
heuristic. The crossover operator may disrupt a group, leaving it with zero machines or zero components. When this occurs, the entire disrupted group is eliminated and the displaced components or machines are reinserted into existing groups. By placing displaced machines with components that need them or displaced components with machines they require, the heuristic speeds up the work of the algorithm by moving it in the right direction through intelligent replacement.

In all the benchmark problems, the grouping efficacy of the solution obtained by the proposed algorithm is either better than that of the other method or its equal. These results are highlighted. There were several issues that needed to be considered when interpreting results. Some of the data sets reported in Goncalves and Resende (2004) were inconsistent with data in the original references. In problem 9, shown in Table 4.14, 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. In terms of the grouping efficacy measure, the Pro-GGA produced results that were equal to, or better than, all the other methods. The Pro-GGA produced the best solutions in all cases. The Pro-GGA also performed equal to other GGAs including the standard GGA (Brown and Sumichrast 2001), EnGGA (Tunnukij and Hicks 2009) and the HGGA (James et al. 2007) that combined the standard GGA with a local search heuristic (Goncalves and Resende 2004). With efficacy as the measure of effectiveness, Pro-GGA shows an average percentage improvement of 39% over ZODIAC and 11% over GRAFICS. Although not applied to the entire set of test problems, EnGGA outperforms Pro-GGA by 5.94% on one data set (CR6). Pro-GGA requires many more generations to reach this solution.
The proposed algorithm performs as good as the algorithms of other methods. Some of the authors have reported time computation and number of iterations required for solving the problem with a set of parameters. Due to difference in the computational facility, various methods adopted and parameters used, evaluating on grouping measure is focused. The computational time required to run the Pro-GGA with 50 generations was less than 40 s, even for the large population size. For the small problems 1 and 2, the Pro-GGA 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.

4.6 CONCLUSION

In this chapter, more efficient and flexible 0-1 integer programming model was formulated for grouping machines into machine cells and parts into part families. The objective of the model is to maximize the parts flow between machines within a same cell and minimize the intercellular movements between cells. For this, a set of machines having high relationship should be included in a same cell. The computation of parts flow between machines is based on the process plans, production volumes, material handling time etc. A grouping genetic algorithm approach (Pro-GGA) was developed to solve the model efficiently. The numerical examples and comparisons show that the Pro-GGA is effective and outperforms all the other methods considered. The program required less than 1 minute computational time in all situations, even with the large population size. From the results of the study presented in this Chapter, it is concluded that Pro-GGA is a practical tool. It requires only three parameters and is not highly sensitive to the choice of
values for these parameters. For this reason, Pro-GGA is easy to set up and use. In addition, Pro-GGA works with limited reliance on specialized operators developed for a narrow problem range. Another advantage of Pro-GGA is its ability to be adapted to different performance measures. Altering the performance measure may done without changing the steps or parameters of the algorithm.