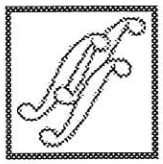
Monograph:
Dimopoulos, C. and Mort, N. (1999) A Genetic Programming Methodology for the Solution of the Cell-Formation Problem. UNSPECIFIED. ACSE Research Report 759 . Department of Automatic Control and Systems Engineering



A GENETIC PROGRAMMING METHODOLOGY FOR THE SOLUTION OF THE CELL-FORMATION PROBLEM

C Dimopoulos and N Mort

Research Report No.759

Department of Automatic Control & Systems Engineering
The University of Sheffield
Amy Johnson Building, Mappin Street, S1 3JD, U.K.

6 October 1999

A GENETIC PROGRAMMING METHODOLOGY FOR THE SOLUTION OF THE CELL-FORMATION PROBLEM

C Dimopoulos and N Mort
Department of Automatic Control & Systems Engineering
The University of Sheffield
Email: cop97cd@sheffield.ac.uk & n.mort@sheffield.ac.uk

Abstract: The problem of identifying machine cells and corresponding part families in cellular manufacturing has been extensively researched over the last thirty years. However, the complexity of the problem and the considerable number of issues involved in its solution create the need for increasingly efficient algorithms. In this report we investigate the use of Genetic Programming for the solution of a simple version of the problem. Genetic Programming is initially employed to attack individual cell formation problems. In a second stage, Genetic Programming evolves a similarity coefficient for the solution of any cell-formation problem.

1 Introduction

The nature of the globalised market has shifted the attention of companies from the mass production of single parts in dedicated transfer lines to the manufacturing of a variety of parts in medium-sized batches. Flow-line production is not flexible enough to cope with this new situation, thus the need for efficient manufacturing systems has emerged. Flexible Manufacturing Systems (FMS) have been employed extensively to provide the flexibility required in various aspects of the manufacturing procedure. The stepping stone for the implementation of an FMS is considered to be cellular manufacturing.

Cellular manufacturing is the implementation of Group Technology (GT) to the manufacturing process. GT was originally introduced by Mitrovanov (1966) and was popularised in the west by Burbidge (1963). It states that significant advantages can be achieved by grouping company elements that are bound by some form of similarity. These elements range from entire departmental units to simple machines or parts. On the plant level the aim of GT is to decompose the manufacturing process into a number of machine cells which are dedicated to the production of corresponding part families. This configuration is traditionally known as cellular manufacturing. The intuition behind cellular manufacturing is an attempt to achieve the mass-production effect of flow-line production in batch manufacturing. The implementation of cellular manufacturing has been reported to result in significant benefits for the manufacturing procedure. Some of these benefits are:

- reduced set-up times
- reduced work-in-progress inventory
- reduced throughput times
- reduced material handling costs
- simplified scheduling

- simplified flow of products
- improved quality

The problem of creating machine cells and associated part families is known as the cell-formation problem. Numerous methodologies have been proposed for its solution over the last thirty years. Many of these approaches have been successful in handling particular versions of the problem. However, the trade-off between modelling an accurate version of the manufacturing process and the resulting computational complexity of the algorithm, means that there is always a research interest in finding more efficient solution methodologies.

Genetic Programming (GP) (Koza, 1992) belongs to the family of evolutionary algorithms, a research field that has expanded rapidly over the last ten years. Evolutionary computation employs the concept of Darwinian strife for survival to guide the search for a potential solution. The probabilistic nature of evolutionary algorithms and their ability to search in parallel through the solution space, means that that they are less likely to be trapped in local optima. The difference in GP from other evolutionary algorithms is that it evolves computer programs of variable length rather than fixed-sized strings of suitably coded solutions.

While evolutionary algorithms have been extensively used for the solution of a wide range of manufacturing optimisation problems (Dimopoulos & Zalzala, 1999a), Genetic Programming has rarely been used in this context. Some notable GP applications have been reported by (Polheim & Marenback, 1996), (McKay *et al.*, 1996), (Dimopoulos & Zalzala, 1999b), and (Dimopoulos & Zalzala, 1999c). In this report we will introduce a novel Genetic Programming approach for the solution of a simple version of the cell-formation problem.

The remainder of this report is organised as follows: In section 2 we take a closer look at the cell formation problem and we discuss some of the issues involved in its solution. A literature survey of the problem is given in section 3. In section 4 we briefly describe the operation of Genetic Programming. The Genetic Programming approach for the solution of the cell-formation problem is illustrated in section 5. Results on a wide range of test problems taken from the literature and comparisons with alternative methodologies are presented in section 6. In section 7 we use groups of test problems as fitness cases for the evolution of a general-purpose similarity coefficient. The conclusions of our research are summarised in section 8.

2 Formulation of the cell-formation problem

The most important step in the development of a cellular manufacturing system is the creation of machine cells and associated part families. There are three basic approaches in designing manufacturing cells:

I. Create machine cells and assign parts to the created cells

II. Form part families and assign machines to the formulated families

III. Create machine cells and part families simultaneously

All cell-formation methods that have been reported in the literature employ one of these approaches. However, not all of them consider the same version of the cell formation problem. In fact, there are numerous alternative problem formulations depending on the

objective of optimisation and the level of manufacturing data that is incorporated in the solution procedure.

2.1 Simple binary matrix formulation

In this report we consider the simplest version of the cell formation problem which is usually illustrated with the help of the machine-component (m/c) matrix $A[n \times m]$ where

n: total number of machines in the plant

m: total number of parts in the plant

Each position in the matrix can assume two values, '0' and '1'. A positive entry indicates that the part of the corresponding column has an operation on the machine of the corresponding row. A '0' entry indicates the opposite. We will illustrate the information provided by the m/c matrix using the following example:

Assume that we have the simple case of a plant that produces 5 parts using 3 machines. By analysing information from the route cards of parts, we obtain the following m/c matrix:

	p1	**p2**	**p3**	**p4**	**p5**
m1		1	1		1
m2	1			1	
m3		1	1		1

Table 1: An example of an m/c matrix

The value of $A_{2,4}$ is equal to '1', thus part 4 needs an operation on machine 2. In contrast, part 4 does not need an operation machine 1 since $A_{1,4}$ is equal to '0'.

Once the m/c matrix has been obtained, the cell-formation problem is transformed to the problem of finding a configuration with all the positive entries arranged inside blocks along the main diagonal of the m/c matrix. A diagonalised matrix allows the easy identification of machine cells and corresponding part families. Table 2 shows the diagonalised version of our example matrix which resulted by rearranging its rows and columns:

	p4	**p1**	**p5**	**p3**	**p2**
m2	1	1			
m1			1	1	1
m3			1	1	1

Table 2: The diagonalised m/c matrix

By observing the matrix it is easy to identify two independent cells, the first one comprising of machine 2 and parts 1 and 4, and the second one comprising of machines 1 and 3 and parts 2, 3 and 5. The main objective of a cell-formation algorithm in this simple version of the problem is the construction of completely independent cells, i.e. cells where the parts included in a part family are solely processed within the corresponding machine cell. However, this is a case rarely encountered in practice.

Table 3 illustrates a situation where the cells that have been formed are not independent:

	p4	p1	p5	p3	p2
m2	1	1		1	
m1				1	1
m3			1	1	1

Table 3: m/c matrix with intercell moves

The reason for this inefficiency is part 3 which requires an operation on a machine that belongs to a different cell (machine 2). It is customary in cellular manufacturing terminology to describe part 5 as an 'exceptional part' and machine 2 as a 'bottleneck machine'. The handling of bottleneck machines and exceptional parts is a significant consideration in cellular manufacturing research. Some of the approaches that have been proposed over the years will be described in the following section.

When completely independent cells cannot be formed, the objective usually becomes the minimisation of intercell moves or the minimisation of material handling costs in general. However, in the case of binary m/c matrices it is common to employ a grouping measure to assess the quality of block diagonalisation. Several grouping measures have been introduced over the years, with grouping efficiency and grouping efficacy being the ones that have been used by the majority of researchers. A detailed explanation of these measures will be given in section 6.

2.2 Advanced formulations

The binary m/c matrix representation of the problem has been extensively used in cell-formation research, mainly because it was introduced and utilised by Burbidge in the first scientific method for creating manufacturing cells, namely Production Flow Analysis. While this representation is easy to comprehend and to work with, it suffers from serious deficiencies that limit its ability to represent realistic manufacturing environments. More specifically, only a limited amount of manufacturing data can be captured by a binary m/c matrix, and some of the missing data are critical in creating the appropriate cell configuration. This fact has lead to the development of advanced formulations of the problem (mainly mathematical programming and graph-based models) which are capable of incorporating a wide range of production data like:

- processing times
- product demands
- machine capacities
- alternative process plans for parts
- batch sizes
- limits on cell sizes and total number of cells
- operation sequences
- multiple machines of the same type

- tooling considerations

However, the larger the amount of data included in a formulation of the cell-formation problem, the more computationally intractable the model becomes, as we will see in the following section. The objective of an advanced formulation can be as simple as the minimisation of intercell moves and as complex as the minimisation of the total costs associated with the production process over a specified period of time.

3 Literature review

3.1 Introduction

The research effort for the solution of the cell-formation problem spans a period of thirty years. Several hundred papers have been published during that time making the task of surveying the field and taxinomizing the approaches extremely difficult.

In this section we will examine a number of papers that are considered to be important in cellular manufacturing literature, as well as some recent publications that are not covered in cellular manufacturing review papers. The list of publications is by no means complete. The aim to outline the state-of-the-art in cell-formation research so that the significance of the comparisons that will be made in later sections can be assessed.

There is no standard way of classifying cell-formation methods. A coarse-grained classification would result in the following three categories:

- Visual inspection methods
- Coding & classification methods
- Production-based methods

Visual inspection methods or simply 'eye-balling' methods rely on the visual identification of machine cells and part families. Considerable experience is required in the process of identification even in small problem cases. However, as the size of the problem increases the task becomes almost impossible.

In coding & classification methods the design characteristics of the parts are used for the formation of part families. Each part is assigned a multi-digit code according to its shape, size, or production requirements, and a classification system is used to group parts according to their code. While coding systems are widely used by companies, very few cell-formation methods are based on them. Notably, it was Mitrovanov (1966) that suggested the use of design characteristics for grouping similar parts. The reader interested in a thorough examination of coding & classification systems should refer to (Hyer & Wemmerlov, 1985).

The core of the cell formation algorithms fall under the category of production-based methods. In general, production based methods analyse the information found on the route cards of parts and bring together parts with similar processing requirements and and/or machines that process similar parts. The Genetic Programming methodology that is presented in this report belongs to the family of production-based methods. A fine-grained classification of these methods result in the following categories:

- Array-based methods
- Hierarchical clustering methods

- Non-hierarchical clustering methods
- Graph-based methods
- Mathematical programming methods
- Artificial Intelligence methods
- Heuristic methods

In the following paragraphs we will focus on these methodologies by discussing a number of important papers in each category. Additional information on cell-formation approaches can be found in the review papers of (Kusiak & Heragu, 1987), (Chu, 1989), (Singh, 1993), (Offodile *et al.*, 1994), and (Selim *et al.*, 1998). The classification system used in this report is illustrated in fig.1.

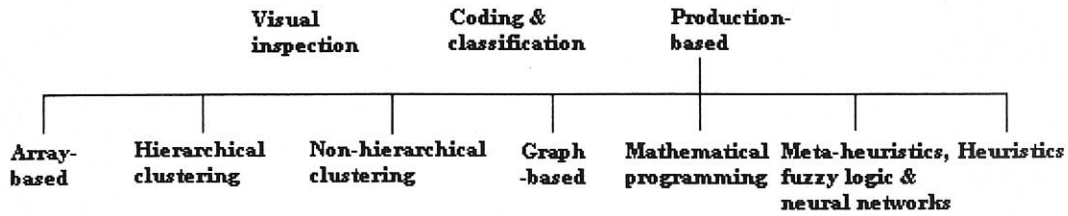


Figure 1: Classification of cell-formation approaches

3.2 Array-based methods

Array-based methods manipulate the rows and the columns of the m/c matrix aiming to obtain visible groupings of machines and components. This is usually achieved by constructing group diagonals which include as many positive entries as possible.

Cellular manufacturing bibliography indicates that the first array-based method for obtaining machine-component groups was part of the Production Flow Analysis (PFA) procedure for the implementation of a cellular manufacturing system (Burbidge, 1963), (Burbidge, 1971), (Burbidge, 1975), (Burbidge, 1977). PFA in its final form was comprised of four main steps:

- Factory flow analysis

  This step was necessary in large industries and aimed to decompose the factory in a number of independent 'major' departmental groups, making the implementation of the group analysis step easier.

- Group analysis

  This step started with the construction of the m/c matrix using information obtained from the route cards. A manual manipulation of rows and columns created machine-component groups. Burbidge believed that these groups existed naturally and it was up to the designer to unveil them. He also claimed that groups could be obtained manually even in large m/c matrices. Several researchers have criticised this claim as unrealistic. Burbidge later presented a seven-step method for obtaining cells which was based on the concept of 'nucleus' machines. The algorithm started by eliminating 'immaterial' machines from consideration, i.e. machines that performed secondary operations, such as washing etc. Then, the machines processing the smallest number of parts were identified as 'nucleus' machines and primitive

'modules' were built around them. In the latter stages of the algorithm final groups were identified by combining or dividing primitive 'modules. Burbidge proposed a number of alternative methods for the elimination of exceptional elements and the balancing of workload between the cells.

- Line analysis

  After the grouping of machines and components, the layout of machines within the cells was chosen based on the flow of parts between machines.

- Tool analysis

  During this step families of tools that processed similar parts were identified and the optimal loading sequence of tools in the machines was decided.

PFA was much more than a matrix-manipulation technique. However, it's mainly group analysis that has received considerable research interest. The obvious improvement on Burbidge's method was the creation of algorithms which were able to manipulate the rows and columns of the m/c matrix without human interaction.

El Essawy and Torrance (1972) introduced a similar procedure for implementing group technology in a manufacturing plant called Component Flow Analysis (CFA). CFA was comprised of three main stages: At the first stage the aim was to categorise components according to their production requirements. A computer program was used for the implementation of this stage. The second stage identified 'approximate' or 'rough' groupings of machines and parts. The third stage produced the final configuration of the plant by analysing the flow patterns of materials and the loads in each cell. While there are some difference between CFA and PFA, it has been indicated (King & Nakornchai, 1982) that both approaches focus on the importance of plant-specific information and the analysis of manufacturing data in every stage of the procedure.

McCormic *et al.* (1972) proposed a general clustering technique called Bond Energy algorithm aiming to achieve the desired configuration of a binary m/c matrix. A 'bond' indicated the presence of consecutive positive entries in rows or columns. The objective of the algorithm was the maximisation of the total bond energy of the matrix defined as the measure of effectiveness (ME). The authors showed that the problem was equivalent to the quadratic assignment problem, thus computationally difficult. They proposed a heuristic procedure for the maximisation of bond energy and illustrated their methodology on decomposition and analysis problems.

King (1980) devised an efficient method for diagonalisation of m/c matrices called Rank Order Clustering (ROC). ROC was based on the ranking of rows and columns according to the binary word represented by the '0' and '1' entries for each of them. Rows and columns were rearranged in decreasing order of their ranking. The process was iterative and continued until no further change could be achieved. King claimed that the algorithm always created a block-diagonal structure if one existed in only two iterations. However, Chandrasekharan & Rajagopalan (1986) depicted that the algorithm had the tendency of gathering as many positive entries as possible in the top-left hand corner of the m/c matrix, while the rest of the matrix was left highly disorganised. This tendency resulted in erroneous identification of bottleneck machines. Another drawback of ROC was its dependence on the initial configuration of the m/c matrix. In some cases the application of ROC could even disrupt an initially good solution. In addition, the algorithm was becoming computationally inefficient for large

matrix instances, since a 2^n (where n=total number of machines or parts) integer equivalent of a binary number had to be stored for each row and column.

King identified the drawbacks of ROC and proposed a series of improvements (King & Nakornchai, 1982). He noted that the problem of storing very large integer numbers could be avoided by simply ranking rows and columns using pairwise comparisons of bit values. Since the time complexity of ROC was cubic and the storage requirements were significant for that time, King & Nakornchai introduced a modified version of ROC, called ROC2. ROC2 utilised linked lists to store the data of the matrix. Linked lists enabled the use of fast and efficient sorting procedures which resulted in an overall algorithm with linear time complexity. King & Nakornchai also presented an interactive algorithm which combined ROC2 with specialised procedures for dealing with exceptional elements and bottleneck machines. Despite all these improvements, the outcome of ROC2 was still dependent on the configuration of the initial m/c matrix.

As we already discussed Chandrasekharan & Rajagopalan (1986) criticised ROC and illustrated its deficiencies. However, they noted that the application of ROC always resulted in one clear-cut machine-component group which was located on the top left-hand corner of the m/c matrix. They proposed an extension of ROC called MODROC, which took advantage of this feature. MODROC started with the execution of two iterations of ROC. Then, the group of machines and components that had been formed on the top left-hand corner of the matrix was recorded and the components were sliced off the matrix. The procedure was repeated until no more components were left. This algorithm created mutually independent part families, but the identified machine cells were most of the times intersecting. Chandrasekharan & Rajagopalan employed a hierarchical clustering procedure (see paragraph 3.3) which created the final plant configuration. While MODROC was less sensitive to the configuration of the initial m/c matrix, it was still not completely independent of it.

Chan & Milner (1982) introduced a fast and efficient method for diagonalising m/c matrices called Direct Clustering Algorithm (DCA). DCA employed a systematic procedure for the manipulation of rows and columns of the matrix. The aim of the procedure was to move the rows with the 'left-most' positive entries to the top of the matrix and the columns with the 'top-most' positive entries to the left of the matrix. The procedure was iterative and continued until no further improvement could be achieved. The converged matrix would ideally have all the positive entries placed along the main diagonal. The main advantage of this method over ROC was that the outcome was independent of the initial disposition of the matrix. This was achieved by a pre-processing stage where the columns and rows were ranked according to the number of their positive entries. In this way, the input to the main phase of the algorithm was always the same.

Garcia & Proth (1986) proposed a cross-decomposition algorithm for the diagonalisation of the m/c matrix. An initial partition of the problem was obtained using the 'nuées dynamiques' method of Diday *et al.* (1982). Then, the rows and the columns of the matrix were permuted, aiming to maximise the sum of parts produced within a given partition and minimise the number of exceptional elements. The algorithm always converged to a solution equal or better to the initial partition.

Khator & Irani (1987) criticised previous cell-formation methods and introduced the Occupancy Value (OV) algorithm which claimed that eliminated most of their deficiencies. A diagonalised matrix was created by progressively ordering components with the lowest machine usage together with their corresponding machines. Ties

between components were broken by calculating their Occupancy Value, a measure of the resulting matrix density if these components were to be selected. The procedure allowed the easy identification of bottleneck machines and exceptional parts.

A cell-formation approach that has attracted considerable attention over the years is the Cluster Identification Algorithm (CIA) of Kusiak & Chow (1987). CIA was basically a cutting algorithm, originally introduced by Iri (1968), able to identify machine cells and part families by drawing vertical and horizontal lines on the m/c matrix. The authors also introduced The Cost Analysis Algorithm (CAA), an extension of CIA which explicitly considered the cost of subcontracting parts that caused intercellular moves.

Balakrishnan(1996) employed CRAFT, an algorithm for calculating material handling costs, in an efficient procedure which transformed the problem of rearranging the rows and the columns of the m/c matrix into a facility layout problem. First, Jaccard's similarity coefficients were calculated for each pair of machines and parts in the plant. Then, two facility layout problems were solved by CRAFT, one for the flow-line of parts and the other for the flow-line of machines. The similarity coefficients of parts and machines were used as inputs to the algorithm. CRAFT rearranged the position of parts and machines in their layout having as objective the minimisation of transportation costs. In this way CRAFT was indirectly rearranging the columns and the rows of the m/c matrix. The main drawback of the algorithm was its sensitivity to the initial layout of parts and machines.

Finally, Mukhopadhyay has proposed a number of alternative methodologies for the solution of binary cell-formation problems. These methods include multidimensional scaling (Mukhopadhyay *et al.*, 1994), vector analysis (Mukhopadhyay & Gopalakrishnan, 1995a), moments-based clustering techniques (Mukhopadhyay *et al.*, 1995b) and conjoint measurement analysis (Mukhopadhyay & Bhandari, 1997). The common feature of all these methods is that they had been originally used for the solution of scientific problems not related to cellular manufacturing.

3.3 Hierarchical clustering methods

Hierarchical clustering methods usually employ a form of similarity or dissimilarity measure for machines or parts. A number of potential solutions is then created by either progressively breaking down a single cell or part family to individual machines or parts (divisive methods) or by progressively merging individual machines or parts to a single cell or family (agglomerative methods).

McAuley (1972) published a pioneering paper in the field of cellular manufacturing. He was the first researcher to employ similarity coefficients for the solution of the cell formation problem. His methodology was comprised of two main stages. First, a similarity coefficient was calculated for each pair of machines that were available in the plant. The value of the coefficient represented the similarity of the machines in terms of the common operations performed. McAuley employed Jaccard's similarity coefficient (Jaccard, 1908) which was calculated by dividing the number of common operations between the machines to the number of operations that were performed on either machine. The matrix of similarity coefficients was used as an input to the second stage of the algorithm, the Single Linkage Cluster Analysis (SLCA) procedure (Sneath, 1957). SLCA generated a dendrogram which linked individual machines or group of machines according to the values of their similarity coefficients. The dendrogram represented a tree of potential solutions. The user could specify a solution by selecting a

similarity threshold which corresponded to a particular grouping of machines. The objective of McAuley's algorithm was the minimisation of the sum of intercell and intracell costs. The calculation of intracell costs was based on a prespecified layout of machines within cells (straight line, rectangle or square). In SLCA the similarity between a machine and a group of machines is calculated by taking the highest similarity coefficient between this machine and the machines included in the group. For this reason, SLCA suffered from the problem of 'chaining', i.e. its application sometimes lead to the grouping of machines that were quite dissimilar in terms of the parts that they were processing. McAuley suggested the use of Average Linkage Cluster Analysis (AVCA) as a cure for this problem. The same similarity coefficient was employed by Carrie (1973) for the calculation of similarities between parts. SLCA was again used as the clustering procedure for the creation of part families. An illustartion of the SLCA procedure is given in section 5.

Seifoddini &Wolfe (1987) extended the use of McAuley's algorithm by introducing some enhancing modifications. First, as suggested by McAuley, they employed AVCA as the clustering method, thus avoiding the effect of 'chaining'. In AVCA the linkage of a machine to a group of machines is calculated by taking the average value of the similarity coefficients between the machine and the machines included in the group. Next, a procedure for eliminating exceptional elements was introduced, based on the duplication of bottleneck machines. For each bottleneck machine in the plant the number of intercellular moves was calculated. The machine that caused the more severe bottleneck was duplicated. It is obvious that this procedure implied the existence of unlimited capital available for duplicating machines, a condition not always true in real manufacturing environments. Finally, the authors introduced a bit-level data storage and analysis technique which simplified the procedure of creating machine cells from the m/c matrix.

The same authors (Seifoddini & Wolfe, 1987) addressed the problem of selecting an optimal value for the threshold level when using a hierarchical clustering procedure. They proposed the evaluation of solutions for several threshold values, aiming to minimise the sum of intercellular and intracellular moves, in other words the overall material handling costs.

Gupta & Seifoddini (1990) extended the applicability of the coefficient-based hierarchical clustering methods by introducing an enhanced version of McAuley's similarity coefficient. The authors argued that the main disadvantage of the simple similarity coefficient was the limited amount of manufacturing information included in it. They introduced the production-based similarity coefficient which explicitly considered the production volume of parts, the part routing sequences and the processing times. Complete Linkage Clustering Analysis (CLCA) was employed for the clustering of machines. In CLCA the similarity between a machine and a group of machines is calculated by taking the lowest similarity coefficient between this machine and the machines included in the group. Machine cells were created iteratively for all possible values of the threshold level. Parts were assigned to machine cells based on the number of operations performed within each cell. During the assignment phase the sequence of operations was explicitly considered. The evaluation of the solutions was based on the minimisation of intercell moves and the minimisation of the total within-cell workload for all parts. The latter criterion was also employed for the identification of bottleneck machines which were candidates for duplication. The superiority of the production-based similarity coefficient over Jaccard's similarity coefficient was illustrated on test problems taken from the literature. Gupta (1993) later introduced an

improved version of the coefficient which explicitly considered the existence of alternative process plans for the parts produced. Seifoddini & Djassemi (1995) also compared the performance of these two coefficients and concluded that the production-based similarity coefficient was able to reduce total material handling costs more efficiently than Jaccard's similarity coefficient.

Another well-reported hierarchical clustering procedure is that of Waghodekar & Sahu (1984). They employed three alternative coefficients for the calculation of similarities between machines. The first coefficient was an alternative version of Jaccard's coefficient, the second one utilised the same parameters but in a product-type rather than an additive-type form and the third one considered the total flow of common components for each machine. A simple hierarchical clustering procedure was used for the identification of manufacturing cells. The authors reported that all coefficients produced similar results on some test problems taken from the literature.

Mosier & Taube (1985) illustrated the deficiencies of ROC and McAuley's SLCA and proposed an extension of the simple Jaccard's coefficient which explicitly considered the relative importance of each part (weight). The weight was equal to the volume of part that was to be produced. The authors presented an additive and a multiplicative form of the coefficient and compared its performance with ROC and McAuley's SLCA on randomly generated test problems. Results indicated that the use of weighted similarity coefficient yielded substantial benefits in many cell-formation problems.

Vakharia & Wemmerlov (1987), (1990) presented a methodology for the creation of manufacturing cells which was based on the identification of part families rather than machine cells. A similarity measure for parts was introduced which considered not only the machines visited by each part, but the operation sequences as well. The merging procedure was interactive, with the designer having the power to approve or disapprove merging based on the information about the resulted skip moves, backtracking moves, etc. After the part families had been created, the designer dealt with backtracking and single-operation parts which had been initially removed from consideration, and decided where to allocate 'key' equipment (equipment required by many parts in the plant). On a final stage more general objectives were considered, like the minimisation of investment and the respect of cell-size constraints. The procedure was highly interactive since it required decisions and experience from a human operator.

Wei & Kern (1989) adopted a modified version of the matching similarity coefficient introduced by Kusiak (1987) which considered not only the number of common machines used by a pair of parts, but also the number of common machines that are not used by the pair of parts. Since the total number of machines in the plant is usually lower than the toatl number of parts and for reasons of computational efficiency the authors employed the modified metric for the calculation of similarities between pairs of machines rather than parts. A simple hierarchical procedure was consequently used for the creation of machine cells.

Tam (1990) introduced a similarity measure for parts which considered not only the commonality of processing requirements but the proximity of operation sequences as well. He employed the K-Nearest-Neighbour clustering procedure (Wong, 1983) for the formation of part families and discussed in detail the advantages of the proposed methodology over the existing clustering procedures.

Luong (1993) presented an innovative cell-formation scheme based on the concept of the cellular similarity coefficient. He introduced a new coding procedure for all parts

which considered their processing requirements rather than their design parameters. The cellular similarity coefficient measured the similarity between two alternative machine cells. Initial cells were created around individual parts, thus they were not disjoint. If the similarity coefficient of two cells was equal to '1', the two cells were merged. The same procedure was continued until a prespecified similarity level. The duplication of bottleneck machines was an internal part of the procedure, thus completely independent cells could be formed by just a single-pass of the algorithm.

Akturk & Balkose (1996) presented a comprehensive multi-objective non-linear integer programming formulation of the problem which simultaneously considered the problems of machine-cell formation and part-family grouping. The model aimed to minimise machine investment and number of skippings while balancing the intercell and intracell workload. The computational intractability of the model was dealt with the introduction of a multiobjective hierarchical clustering procedure which explicitly considered both the design characteristics of the parts and their processing requirements.

Ho & Moodie (1996) considered the case of a flexible manufacturing system where alternative process plans are available for each part. An acyclic network representation was employed as a model for the problem. Each node corresponded to a part operation and carried information about the machines on which the operation could be performed, the processing times and the operation costs. Alternative operations were denoted using OR-branches. Based on this information an operation-based similarity coefficient was calculated for each pair of parts. The formation of part families was accomplished by a hierarchical clustering algorithm suitably modified to allow for human interaction. Finally, machines were assigned to part families according to three alternative scenaria in the plant:

- only existing machines where allowed to form cells
- purchase of new machines was allowed
- all machines were bought from scratch

Mathematical programming was used for the modelling of each of these cases with the objective of minimising total costs including operation costs, annual fixed costs and costs of intercell operations. The models were computationally intractable, thus the authors proposed relaxed versions which resulted in mixed-integer programming problems. Still, the suitability of the method for large-sized problems is questionable.

The hierarchical clustering methods described so far employed agglomerative procedures for the creation of machine cells or part families: individual machines or parts were merged into progressively larger cells or families until a single cell or family was obtained or until a size constraint had been reached. Stanfel (1985) proposed a simple hierarchical divisive procedure for the creation of machine cells which was not based on the calculation of similarity coeficients. The algorithm started with an initial cell comprised of all available machines. An iterative procedure followed with each machine leaving the cell to either form a new cell or to join one that had already been formed. All the solutions were evaluated in terms the resulting intercell moves and the number of extraneous transitions caused by the presence of machines within a cell that were not processing all the family parts.

3.4 Non-hierarchical clustering methods

Non-hierarchical clustering methods also employ a measure of similarity or dissimilarity for the grouping of machines or parts. An initial partition, or a number of seed points are then selected and used for the classification of machines or parts. The main drawback of these methods is that they usually require the specification of the total number of manufacturing cells in advance.

Chandrasekharan & Rajagopalan (1986b) introduced a three-step non-hierarchical clustering algorithm for the solution of the cell-formation problem. A modified version of MacQueen's k-means method (MacQueen, 1967) was employed for the initial clustering of machines and components. Since the k-means method required the prespecification of the total number of clusters, a formula was derived for the calculation of the maximum number of independent cells that could be formed for a certain problem. During the second stage part families were allocated to machine groups according to their efficiency factor, an indicator of the within-cell utilisation for each part-family. The output of this stage was used for the determination of ideal-seed clustering points in a perfectly diagonalised matrix. These points initialised a new run of the k-means clustering algorithm which resulted in the elimination of singleton clusters. In the same paper Chandrasekharan & Rajagopalan introduced *grouping efficiency*, a qualitative measure of matrix diagonalisation. While grouping efficiency has been criticised of being inadequate to assess the performance of an algorithm, it has been used by a considerable number of researchers over the years.

The same authors (Chandrasekharan & Rajagopalan, 1987) introduced an extension to the ideal-seed clustering algorithm called ZODIAC (Zero-One Data: Ideal-seed Algorithm for Clustering). The algorithms were quite similar with the exception of the initialisation phase where ZODIAC considered a choice of different seeding procedures. The problem of seeding was discussed in depth and it was concluded that the use of 'natural' seeds resulted in better clustering of data. The concept of *limiting efficiency* was also introduced, an indication of the best grouping efficiency that could be achieved for a given matrix. The ratio of grouping efficiency to limiting efficiency was defined as *relative efficiency*. Relative efficiency provided a realistic indication of the performance of the cell-formation algorithm.

Chandrasekharan & Rajagopalan (1989) employed ZODIAC for their investigation of the characteristics of binary data matrices. A set of progressively more difficult problems were solved by ZODIAC and the values of grouping efficiency were recorded and associated to the parameters of the matrices. The mean value and the standard deviation of the pairwise similarities of rows and columns as measured by Jaccard's similarity coefficient were used as the discriminating characteristics of the matrices. The authors concluded that the value of standard deviation was a good indication of the groupability of the matrix. However, since other discriminating parameters, like the number of rows and columns, were not included in the study, it was suggested that this conclusion should not be regarded as nothing more than an indication of the actual relationship between the structure of the matrix and its groupability.

Srinivasan & Narendran (1991) illustrated some of the deficiencies of ZODIAC in terms of the choice of initial seeds and the use of the city block distance as the clustering criterion. They introduced a new non-hierarchical clustering procedure called GRAFICS (GRouping using Assignment method For Initial Cluster Seeds). GRAFICS identified initial machines for seeding by solving the assignment problem as introduced by Srinivasan *et al.* (1990). The maximum density rule was employed as the clustering

criterion. The main algorithm progressed by alternatively clustering machines and parts until no improvement could be made in terms of the number of exceptional elements and voids (zero's inside the block diagonal matrices). GRAFICS did not allow the existence of singleton clusters. GRAFICS was tested against ZODIAC on a considerable number of problems taken from the literature and was found to be superior in most cases. Srinivasan (1994) later extended GRAFICS by using a minimum-spanning tree algorithm for the creation of initial seeds. The modified GRAFICS algorithm performed better than simple GRAFICS and ZODIAC on a wide range of problems.

Gu (1991) indicated the difficulty of representing the operations of flexible NC machines in binary m/m matrices. He introduced a novel representation which was based on two binary matrices. The first matrix associated parts with operations, while the second one associated operations with machines. This type of representation allowed the consideration of machines that performed multiple operations, machines with multiple units and parts with alternative process plans. A non-hierarchical clustering procedure was employed for the formation of part families.

Nair & Narendran (1998) presented a non-hierarchical clustering method for the creation of manufacturing cells called CASE (Clustering Algorithm for SEquence Data). They introduced a new similarity metric which explicitly considered the sequence of operations for each part, multiple visits to machines, and part demands. The metric was used for the identification of initial seeds in a non-hierarchical clustering algorithm. Nair & Narendran (1999) later presented an enhanced version of CASE called ACCORD (A bicriterion Clustering algorithm for Cell formation using Ordinal and Ratio-level Data). The difference of ACCORD was that it combined the similarity coefficient used in CASE with a new similarity coefficient that captured the workload similarity between a pair of machines.

3.5 Graph-based approaches

Graph-based methods employ a graph or network representation of the cell-formation problem and use relevant techniques to create manufacturing cells.

One of the first graph-based approaches for the solution of the cell-formation problem was introduced by Rajagopalan & Batra (1975). Their method combined graph theory and similarity coefficients. The problem was modelled with the help of a vertex-edge map. Machines in the plant were represented by vertices. Jaccard's similarity coefficients were calculated for each pair of machines. Vertices were connected by edges if the value of the similarity coefficient for the corresponding pair of machines was larger than a prespecified threshold level. All the cliques (complete maximal subgraphs) were identified and they were used to form hybrid machine cells. At this point machines could be present in more than one cell, thus a procedure was needed for the creation of mutually independent cells. A new graph was constructed with each vertex representing a cell and each connecting edge representing intercellular moves between the hybrid cells. The graph was partitioned with the help of a standard graph-partitioning procedure introduced by Kernighan & Lin (1970). The objective of the algorithm was the minimisation of the total intercell moves. The resulted partitions corresponded to the final configuration of cells in the plant.

Sun *et al.* (1995) utilised a similar graph representation but in a different context. They argued that important information for the design of manufacturing cells was not

included in the m/c matrix. Instead, they constructed a flow-matrix which contained information about the amount of parts that were processed in sequence by any pair of machines. Construction of the flow-matrix required knowledge of the processing sequences of parts and their required quantity. Based on the information found in the flow-matrix a graph was constructed with the nodes representing the machines and the weighted arcs representing the flow of products between them. The authors indicated that the partitioning of that graph was a problem equivalent to the quadratic assignment problem. The authors proposed a tabu search methodology for the solution of the problem, enhanced with a look-ahead evaluation method for the selection of moves. The framework was tested on a series of randomly generated test problems.

Rath *et al.* (1995) also employed Rajagopalan & Batra's representation to model the cell formation problem. The difference of their representation was that the weight of the connecting arc was determined by the number of common processing operations and not by Jaccard's similarity coefficient. The graph colouring approach was used as the method of obtaining disconnected subgraphs.

(De Witte, 1980) combined the hierarchical clustering procedure of McAuley with the graph-partitioning approach of Rajagopalan & Batra (1975) (see graph-based methodologies). The machines in the plant were initially divided in three main types: primary machines (machine types of which only one unit was available), secondary machines (machine types of which multiple units were available) and tertiary machines (machine types of which enough units were available to cover every cell in the plant). A machine-to-machine combination matrix was then created based on the routings of parts and their required quantity. This matrix was utilised for the calculation of three different similarity coefficients which were fed as input to Rajagopalan & Batra's graph partitioning procedure for the creation of manufacturing cells. Primary, secondary and tertiary cells were created in a sequential manner. Finally, secondary and tertiary cells were added to primary cells to obtain the final design.

Kumar *et al.* (1986) discussed the group technology problem within the context of flexible manufacturing systems. Their aim was the creation of cells which incorporated not only machines and parts but important FMS components as well such as fixtures and pallets. They presented a grouping procedure which was based on a bipartite graph representation of the problem. Machines and parts were represented by vertices, while their potential relationship was depicted by the existence of a connecting edge. The aim was to find an optimal k-decomposition of the graph which minimised the weighted interconnection between the k subgraphs. A 0-1 quadratic programming mathematical model of the problem was developed, together with a two-phased methodology for its solution. The first phase created a starting k-decomposition by solving a linear transportation approximation of the problem. A linear transportation grouping procedure was then employed for the creation of solutions.

Vannelli & Kumar (1986) focused on the development of a method for finding the minimal number of bottleneck machines or parts when creating manufacturing cells. They showed that the problem was equivalent to finding the minimal cut-nodes of a bipartite graph while disconnecting it to a number of subgraphs. Since the problem was NP-complete, a heuristic procedure (Lee *et al.*, 1979) was employed for its solution. The same authors later extended their methodology by introducing the concept of weighted graphs (Kumar & Vannelli, 1987). The improved model was able to tackle cost-based problems by simply assigning costs as weights for each part in the graph.

Askin & Chiu (1990) presented an elegant mathematical programming formulation of the problem which considered several elements of the production process, like fixed period machine and group costs, part demand, machine capacity, tooling costs and multiple machine of the same type. They described the associated problem as the economic Group Technology Configuration Problem (GTCP). The four-part objective function aimed to minimise machine overhead, group overhead, tooling costs and intercell material handling costs. The large number of variables in the model lead to the creation of two submodels, one for the part-to-machine assignment problem and one for the machine grouping problem. A heuristic solution procedure was introduced, comprising of four stages: batch sizing, assignment of parts to machines when multiple machines of the same type existed, assignment of machines to cells and intergroup handling reduction. The second and third steps of the procedure were solved using the graph-partitioning procedure of Kernighan & Lin (1970) suitably accommodated and enhanced for each case. The most interesting feature of the methodology was the complete automation of all stages. Even in the case of bottleneck machines no human interaction was required to decide on potential duplications.

Askin *et al.* (1997) also addressed the concept of flexibility in the design of a cellular manufacturing system. Flexibility was considered explicitly as an objective of the design in terms of machine types, routings, parts volume and parts mix. A four-step algorithm was proposed for the solution of the model. The first stage considered the problem of assigning operations to machine types. A mathematical model was introduced with the objective of minimising the total annual operating cost of the assignments and the annual procurement cost of machines. Since the problem could be reduced to a generalised assignment problem, a suitable heuristic was employed for its solution. The second stage of the algorithm was identical to the one already described in (Askin & Chiu, 1990). However, a different graph partitioning procedure was used to assign part operations to specific machines. The third stage identified candidate machine cells using the same algorithm. The objective was the minimisation of intercell material flow and the maximisation of flexibility in terms of adapting to changes in processing requirements. The final stage was concerned with potential modifications to the preliminary design which could increase the volume and routing flexibility of solutions.

Al-Qattan (1990) described an interesting methodology for the solution of the cell-formation problem which was based on network analysis. Cells were created by branching on machines and bounding on their corresponding parts. The selection of initial machines was based on their utilisation, i.e. how likely it was that they would end up as bottleneck machines. The resulted network diagrams provided the designer with a wide range of alternative solutions.

Askin *et al.* (1991) proposed a diagonalisation method based on the hamiltonian path representation of the problem. Initially, the distance matrix for all machine and parts in the plant was calculated. A suitably modified version of Jaccard's similarity coefficient was employed as the distance metric. The problem of rearranging the rows and columns of the m/c matrix was modelled as a graph-based Travelling Salesman Problem (TSP) with the objective of finding the shortest tour of all vertices. TSP required a cyclic solution, thus the associated Hamiltonian Path Problem (HPP) had to be considered since it did not require a return tour to the starting vertice. Graph heuristic procedures where used for the solution of both problems.

Wu & Salvendy (1993) presented an acyclic graph representation of the problem and proposed the use of a standard network optimisation technique for its partitioning into a number of disconnected subgraphs. They also introduced a novel heuristic partitioning procedure which utilised information about intercell material flow in a similar way to a hierarchical clustering procedure. The authors indicated that their model could be extended to consider the processing times of parts.

Vohra *et al.* (1993) also proposed an acyclic graph representation of the problem and employed the graph-partitioning procedure of Gomory & Hu (1971) for the creation of disconnected subgraphs. The objective of their model was the minimisation of the total amount of processing time that resulted from intercell moves.

Ng (1993) introduced an minimum spanning tree methodology for the solution of the problem. The nodes of the tree represented the rows of the matrix and the connecting arcs denoted the distance between them, in other words the level of their dissimilarity. K machine cells were obtained by deleting the (k-1) largest arcs from the tree. A procedure was also presented for the re-assignment of parts to machine cells aiming to improve the derived partitions. A worst-case analysis of the algorithm was performed in terms of the grouping efficiency and grouping efficacy (Kumar & Chandrasekharan, 1990) measures. Ng illustrated the deficiencies of these grouping measures and proposed the *weighted grouping efficacy* measure for the evaluation of solutions.

The concept of minimum spanning trees was also employed by Lin *et al.* (1996) in an attempt to solve a more detailed version of the cell-formation problem. The objective of their cost-based mathematical model was the minimisation of the sum of intercell processing costs, intracell processing costs and total cell-balance delay costs. Since the formulation was computationally intractable, a minimum spanning tree heuristic was employed for its solution. Disconnected subgraphs were created by iteratively deleting arcs until no configuration could be found that resulted in lower overall costs. The method was compared with some classic array-based methods on test problems taken from the literature and produced excellent results. In addition a case-study application of the method was presented for a company that manufactured irrigation products.

Lee & Garcia-Diaz (1993) modelled the problem using a capacitated circulation network which illustrated the machines available in the plant and the level of dissimilarity between them. Dissimilarity was quantified with the help of Hamming distances. Based on this model, a linear programming formulation of the model was presented. The solution methodology was based on the identification of closed loops in the network using a relaxation method suggested by Bertsekas & Cheng (1988). The algorithm proved to be computationally efficient even for large-sized instances of the problem.

Wu (1998) examined the case of multiple machines of the same type being available in the plant. Cell-formation methods usually dealt with this issue after the initial cell configuration had been obtained. Wu illustrated how this sequential procedure could lead to the erroneous identification of bottleneck machines. He introduced a model which accommodated the concurrent formation of cells and assignment of identical machines to individual cells. His model was based on a graph-representation of the problem where an identical machine was denoted by a complex node that contained a number of multiple units. Machine types for which only a single machine was available were denoted by simple nodes that contained no units. Complex nodes could be broken down to simpler structures containing fewer units. Arcs were allowed to pass through complex nodes, enabling in that way the partitioning of the graph. An arc indicated a

successive operation of a part on the machines represented by the connected nodes. Two graph-partitioning procedures that had been originally introduced by Wu, were employed for the creation of cells. The objective of the algorithm was the minimisation of total transportation costs.

Kandiller (1998) developed an interesting graph-theoretic approach for the solution of the cell formation problem which was based on a hypergraph representation of the manufacturing environment. A hypergraph is an extension of a normal graph where an edge (defined as hyperdge in this case) can be incident to more than two vertices. The hypergraph of the cell formation problem was constructed by vertices which represented machines and by hyperedges which represented parts. A hyperedge connected all machines (vertices) that were needed for a part to be produced. Partitioning of hypergraphs is not an easy task, thus Kandiller proposed a methodology which was based on the approximation of a hypergraph using either cliques or star graphs. A maximum flow – minimum cut tree was then created based on a methodology proposed by Gomory & Hu (1971). Information taken from the cut tree allowed the iterative bi-partitioning of cells until the desirable value of the objective defined by the designer had been reached. The performance of the framework on various objectives and size constraints was investigated using a series of randomly generated test problems. In addition, it was tested against six well-known cell-formation methodologies, producing one of the leading performances.

3.6 Mathematical Programming

As it has already been discussed mathematical programming formulations of the cell formation problem are capable of considering a wide range of manufacturing data. Several types of integer programming formulations have been proposed over the years and especially the last decade. Most of these models suffer from computational intractability and require the a priori specification of the total number of manufacturing cells.

Kusiak (1987) was one of the first researchers to propose the use of mathematical programming for the solution of the cell formation problem. He introduced a p-median zero-one integer programming model for the formation of part families. The objective of the model was the maximisation of similarity of parts within the part families in terms of the common machines used. A standard integer programming package (LINDO) was employed for the solution of the problem. Kusiak additionally presented a generalised zero-one integer programming formulation which considered the existence of alternative process plans for each part. A Generalised Assignment Problem (GAP) formulation which was equivalent to Kusiak's p-median formulation was proposed by Shtub (1989).

Choobineh (1988) presented a two-stage procedure for the design of a cellular manufacturing system. First, a hierarchical clustering algorithm was employed for the creation of part families. The similarity measure between parts was an enhanced version of Jaccard's similarity coefficient. The existence of alternative process plans was also addressed during that stage of the procedure. After the creation of part families, a linear integer programming formulation of the problem was presented with the objective of minimising the sum of production costs and the costs of acquiring and maintaining machine tools.

Another early attempt to employ mathematical programming for the solution of the cell formation problem was due to Co & Araar (1988). They considered the case where multiple units of machines of the same type were available in the plant. A mathematical model was introduced with the objective of balancing the workload between identical machines. On a second stage the resulted m/c matrix was used as input to a modified version of ROC. Finally, the output of ROC was partitioned using a direct search heuristic procedure.

Wei & Kern (1990) introduced a zero-one integer programming formulation which explicitly considered the available capacity of machines. The objective of the model was the minimisation of opportunity costs which resulted either in the form of material handling costs for intercellular moves, or in the form of subcontracting costs for exceptional moves. The authors indicated a number of alternative objectives that could be used in conjunction with the proposed mathematical formulation.

Nagi *et al.* (1990) presented a linear integer programming formulation which explicitly considered the existence of alternative process plans for each part and the available capacity of machines. The objective of the model was the minimisation of the total intercell traffic within the system. The authors proposed a decomposition of the problem since it was too complex to be solved directly. The first subproblem determined the process plan for each part out of the set of available process plans. The second subproblem identified machine cells. Any standard linear programming solver could be used for the solution of the first problem but the complexity of the second one enforced the use of the intercell traffic minimisation heuristic (Harhalakis *et al.*, 1990). The main disadvantage of the procedure was its sensitivity to the initial partition that was required as input for the solution of the first subproblem. Since the algorithm was relatively fast, the authors suggested its initialisation from various random partitions.

Rajamani *et al.* (1990) also studied the cell formation problem under the existence of alternative process plans for the parts manufactured. They presented integer programming models for the solution of three separate instances of the problem. The first model assigned machines to parts considering the alternative routings available. The resulted m/c matrix could be used as an input to any of the block diagonalisation algorithms already discussed. The second model assumed that part families had already been formed and machines had been assigned to the associated cells. The third model aimed to simultaneously form part families and machine cells. The objective of all models was the minimisation of capital cost. The LINDO package was used for their solution. However, the authors indicated that as the size of the problem increased, the models were becoming computationally intractable.

Boctor (1991) presented a simple zero-one integer programming formulation of the problem which considered only the data available from the binary m/c matrix. The objective was the minimisation of the total number of exceptional elements. An efficient procedure for the linearisation of the objective function was proposed. In addition, Boctor showed that a large number of integrality constraints could be relaxed without affecting the binary outcome of the solution. Even with these modifications the model was computationally intractable for large problem instances. Boctor proposed the use of simulated annealing for these cases.

Gunasingh & Lashkari (1991) presented a detailed mathematical formulation of the problem which incorporated a considerable number of production data such as the number of identical machines available, fixed costs of machines, production volumes, processing times etc. The model aimed to group machine and parts simultaneously by

computing a compatibility measure between them. The value of the measure was based on the tooling requirements of parts and the tooling capabilities of machines. The model was comprised of two zero-one integer programming formulations. The first one grouped machines and parts having as objective the minimisation of the sum of compatibility measures. The second one formed the final groups while seeking a trade-off between the cost of duplicating machines and the cost of accepting intercellular moves. After the objectives of both problems had been linearised, a standard integer programming solver was employed for their solution. For large-sized problems the authors proposed the use of an approximate solution for these cases, based on the decomposition of the non-linear models into separate submodels, one for the machine allocation and one for the part allocation problem.

Shafer & Rogers (1991) discussed in detail the mathematical programming approaches that had been employed up to that time for the solution of the cell formation problem and highlighted their deficiencies. They introduced a goal programming formulation comprising of a p-median model for the identification of part families and a Travelling Salesman Problem formulation for the identification of within-cell sequences that minimised set-up times. Additional objectives of the formulation were the minimisation of intercell moves, the minimisation of investment on new equipment and the respect of machine utilisation constraints. The authors presented three versions of the model, depending on the design of the system being made from scratch, with existing equipment, or with a mixture of new and old equipment. The model was computationally intractable, thus a heuristic procedure was proposed for its solution. The design of part families was dealt by a modified p-median formulation of the associated problem, while the optimal within-cell processing sequence of parts was found by solving a zero-one integer programming formulation of the TSP.

Jain *et al.* (1991) presented a zero-one integer programming model which jointly considered the cell formation problem and the tool provisioning problem in a flexible manufacturing system. The tool provisioning problem determines the number of tools that need to be purchased for each type used in the manufacturing procedure. The authors claimed that the algorithm was ideal for the case of small-sized flexible manufacturing cells since the computational requirements are not extensive.

Logendran (1992) focused on the problem of duplicating bottleneck machines in a cellular manufacturing environment. He criticised previous approaches for assuming unlimited capital availability and for not considering the sequence of operations during the calculation of intercell moves. He proposed a two-phased algorithm for duplicating bottleneck machines within a given budget. The algorithm required that that an initial assignment of machines and parts had already been implemented. The first phase defined the binary variables and the constraints of a zero-one integer programming formulation that was solved during the second phase. The variables were identified by calculating savings in material handling costs and amortised costs from duplicating machines. A standard integer programming solver was employed for the solution of the model.

Shafer *et al.* (1992) discussed the handling of exceptional elements after the initial cell configuration had been created and given that no further process plan modification could reduce their number. Exceptional parts can be either subcontracted, or their respective machines can be duplicated. A third alternative is to allow the existence of intercell moves and take no further action. Each of these alternatives incurs a different cost on the production process. The authors constructed an integer programming model

which aimed to achieve the right balance between all the alternative actions by minimising the sum of incurred costs on a yearly basis. Some of the integrality constraints of the variables were relaxed and the model was solved using the LINDO package.

Dahel & Smith (1993) noted that the creation of pure manufacturing cells decreased the process flexibility of the plant. Two zero-one integer programming models were proposed for the solution of the problem. The first one created machine cells and part families simultaneously with the objective of minimising the total number of intercell moves. The second model explicitly considered the issue of process flexibility by introducing a multiobjective function for the minimisation of intercell moves and the maximisation of machine routing flexibility. The constrained method was employed for the generation of non-dominated solutions. The authors suggested that GT research should focus on the development of analytic methods that provided optimal solutions. However, they did not address the issue of computational intractability as the size of the problem increases.

Suresh *et al.* (1995) argued that previous cell formation approaches had failed to address sufficiently the issues of large problem instances, alternative process plans and multiple objectives. He proposed a comprehensive hierarchical procedure which combined efficiently different optimisation methods for the creation of manufacturing cells. Initially, a fuzzy ART neural network was employed for quick identification of part families. Then, a mixed integer goal programming model was constructed for the grouping of machines to cells. The weighted objective function of the model included the maximisation of cell inter-dependence, the minimisation of purchasing of new machines, the maximisation of release of extra machines and the maximisation of routing flexibility. Finally, another goal programming model was created for the minimisation of intercell moves. The framework contained a number of additional features such as the identification of alternative part families during the first phase and the reallocation of parts to these families during the second phase. Human interaction was also required in various stages. However, even for such an impressive framework, the prespecification of the total number of cells remained an unresolved issue. Suresh proposed as a remedy the iterative execution of the second phase of the algorithm for several values of the required number of cells.

Atmani *et al.* (1995) presented yet another mathematical programming procedure for the solution of the cell formation problem with alternative process plans. They proposed a model which simultaneously considered the allocation of operations to machines and the allocation of machines to cells. A standard integer programming solver was employed for the solution of the resulted zero-one integer programming formulation. The objective of the model was the minimisation of the sum of operation costs, refixturing costs and transportation costs.

Zhu *et al.* (1995) introduced a zero-one integer programming formulation of the problem with the objective of maximising the opportunity costs associated with all the parts manufactured within the system, i.e. the parts that were not needed to be subcontracted. The authors showed that their formulation resulted in less number of variables and constraints and in faster computational times than the corresponding formulation of Wei & Gaither.

Cheng *et al.* (1996) proposed a simple zero-one quadratic assignment formulation of the problem based on the information available from the binary m/c matrix. The objective of the model was the minimisation of the sum of Hamming distances between

machines within the cells. A truncated-tree heuristic algorithm was employed for the solution of the problem. The authors extended their formulation to allow for the existence of multiple machines of the same type.

Hwang & Ree (1996) also focused on the development of a methodology for solving the cell formation problem when alternative process plans existed for some parts. They proposed an algorithm which addressed the problems of selecting a process plan and forming the part families in a sequential manner. First, a quadratic integer programming formulation of the process plan selection problem was introduced, with the objective of maximising the sum of compatibility coefficients between process plans. These coefficients were customly designed to award route pairs with high similarity and to punish route pairs with mediocre similarity. The intuition behind this procedure was that route pairs requiring a limited number of similar machines should not be promoted. After the optimal process plans had been selected, Kusiak's p-median integer programming formulation was employed for the creation of part families. The methodology was compared with the Kusiak's generalised GT formulation of the same problem and results showed that its performance was highly competitive. The LINDO package was employed for the solution of both integer programming models.

Amirahmadi & Choobineh (1996) presented a framework for identifying and handling exceptional elements and bottleneck machines in binary m/c matrices. In contrast with conventional approaches their methodology had the ability to identify exceptional parts and bottleneck machines either independently or simultaneously during the diagonalisation of the matrix. The authors proposed three alternative cost-based integer programming models for dealing with the identified exceptional parts and/or bottleneck machines.

Adil *et al.* (1997) presented a non-linear zero-one integer programming formulation of the problem which aimed to minimise the weighted sum of the total number of voids and the total number of exceptional elements. Given the non-linearity of the model, two heuristic methods were proposed for its solution, the Assignment Allocation Algorithm (AAA) and a simulated annealing algorithm. AAA decomposed the problem into two subproblems, one for the assignment of machines to cells and the other for the allocation of parts to cells. The solution of each problem determined the values of the respective parameters in the mathematical model. The sensitivity of the procedure to the initial partition lead to the utilisation of a simulated annealing algorithm. Both algorithms were tested on a wide variety of large and ill-structured matrices producing very good results.

Taboun *et al.* (1998) noted that the concept of flexibility of product demand during a planning horizon had never been addressed in the models of the cell formation problem that had been proposed over the years. Product demand and product mix was always assumed to be unchanged, a condition which is unrealistic in modern manufacturing systems. An elaborate mixed integer programming mathematical model was introduced which attempted to create manufacturing cells over multiple planning periods where product demand changed according to forecasts or according to prespecified life-cycles. The multi-part cost-based objective function of the model aimed to minimise the sum of cell configuration costs, machine capital investment costs, machine procurement and salvage costs, idle time costs, intercell movement costs and part subcontracting costs. The model itself was computationally intractable, thus the authors produced a two-stage methodology for its solution which they claimed that reduced the computational requirements substantially. The first stage of the procedure was a heuristic algorithm

for the creation of part families based on a similar algorithm originally introduced by Taboon. The total number of cells in the plant had to be prespecified at this stage. Once the part families had been formed, the constraints associated with them could be dropped from the original model resulting in a much more tractable mathematical formulation.

Finally, Deutch *et al.* (1998) proposed a modified p-median formulation of the problem with the objective of minimising the sum of dissimilarities of machines that belonged to the same group. This modification lead to a fully linearised version of the problem solved by a vertex substitution heuristic approach introduced by Teitz & Bart (1968). A worst-case analysis on the running time of the heuristic resulted in an overall time complexity of $O(n^2)$ where n is equal to the total number of parts.

3.7 Meta-heuristics, fuzzy logic & neural networks

3.7.1 Evolutionary algorithms

The application of evolutionary algorithms (EA's) to the cell formation problem has been described in detail in (Dimopoulos & Zalzala, 1999a). For the interested reader we note that the most significant approaches have been presented by Venugopal & Narendran (1992), Gupta *et al.* (1996), Billo *et al.* (1996), Joines *et al.* (1996), Hwang & Sun (1996) and Hsu & Su (1998).

There are also some methodologies that have been proposed quite recently which are not included in (Dimopoulos & Zalzala, 1999a), for example:

Cheng *et al.* (1998) noted that the reorganisation of rows and columns in a binary m/c matrix can be described as a permutation problem equivalent to the TSP, where the objective is the minimisation of some kind of distance measure between columns or rows. They employed a real-coded EA with path-representation for the solution of the problem. The Minkowski metric was used as an indication of the distance between a pair of machines or parts. The performance of the algorithm was compared with that of ZODIAC on a wide range of problems taken from the literature and was found to be superior in most cases.

Gravel *et al.* (1998) considered the version of the cell formation problem which allows the existence of alternative process plans for the parts. A double-loop EA was employed for the solution of the problem with the objective of minimising the volume of intercell moves and balancing the load within cells. The external loop of the EA used Venugopal & Narendran's coding for the assignment of machine to cells. A second internal loop which determined the allocation of process plans to parts was used for the evaluation of solutions created in the external loop. Different multiobjective optimisation approaches were tested, including the epsilon-constraint approach and the weighted-sum approach.

Lee *et al.* (1996) combined similarity coefficients and genetic algorithms in a multi-stage cell-formation procedure. Their methodology explicitly considered the production volume of parts, the processing sequence of parts and the alternative routings available. EA's were used as the optimisation method in all stages of the procedure. The basis of the algorithm was the calculation of the machine-chain similarity coefficient (MCSC) between all pairs of machines. MCSC represented the level of direct or indirect movement between two machines. A series of EA's were then employed for the creation of manufacturing cells. The methodology was impressive in terms of the

amount of manufacturing data considered. The use of multiple EA's was computationally intensive, however, the author indicated that if parallel computers were to be used, even large-sized problems could be solved.

3.7.2 Simulated annealing

Sofianopoulou (1997) introduced an efficient mathematical programming formulation of the problem which did not require the prespecification of the total number of cells in the plant. The objective of the model was the minimisation of the total intercell traffic. A simulated annealing algorithm was employed for its solution. Sofianopoulou (1999) later extended the use of simulated annealing to cell-formation problems with alternative process plans and duplicate machines. She presented non-linear mathematical programming formulations for the machine allocation and part allocation problems respectively. Since the models were computationally intractable, a novel simulated annealing procedure was proposed for their solution. The procedure had the ability to move simultaneously in two different search dimensions. The algorithm started with a random allocation of parts to process plans. Then, the algorithm searched for the machine-cell configuration that minimised the number of intercell moves, given the part-process allocation. Once the termination criterion had been reached, a new part-process allocation was randomly created and the same procedure was repeated until a global termination criterion was reached. The efficiency of the proposed methodology was illustrated on several test problems taken from the literature.

3.7.3 Tabu search

Aljaber *et al.* (1997) modelled the cell formation problem using a pair of shortest spanning path problems, one for the machines (rows) and one for the parts (columns) of the m/c matrix. A modified version of Jaccard's similarity coefficient was employed for the calculation of distances between pairs of machines or parts. The authors introduced a tabu search methodology for the solution of both problems. The algorithm was able to accommodate the consideration of additional manufacturing data with a suitable modification of the distance measure used.

Vakharia & Chang (1997) presented both a simulated annealing and a tabu search methodology for the solution of a detailed version of the cell formation problem. A large problem of manufacturing data were included in the formulation of the problem, like transportation costs, product demands and processing times. The objective was the minimisation of machines investment and material handling costs. A comparison of the algorithms on a number of test problems showed that simulated annealing outperformed tabu search both in terms of solution quality and computational complexity.

3.7.4 Fuzzy logic

Chu & Hayya (1991) indicated there was a degree of uncertainty in the allocation of a part to a specific part family. This uncertainty could be expressed with the help of fuzzy sets. For each part processed a degree of membership was defined in relation to each part family. The authors employed the generalised fuzzy c-means algorithm for the clustering of parts. The advantage of this methodology was that it provided the designer with a number of alternative solutions.

3.7.5 Neural networks

Kao & Moon (1998) employed the concept of memory association for the solution of the cell-formation problem. The intuition behind their approach was to simulate the association procedure that takes part in the memory of a production engineer who is

faced with the task of creating manufacturing cells. The methodology was comprised of two main stages: First, an autoassociative neural network formed part families by considering the characteristics (features) of the parts. Then, a heteroassociative neural network created machine cells by considering the relation between machines and part features. In addition, an extension of the latter network was introduced which was able to create groups for other important GT domains, like tool sets, canned cycles etc.

3.8 Heuristics

Askin & Subramanian (1987) were among the first researchers to introduce a cost-based model for the evaluation of cell-formation solutions. Their model explicitly considered a number of cell-related costs like set-up costs, variable production costs, material handling costs and fixed machine costs. The authors suggested a procedure for the calculation of these costs and proposed a three-step heuristic algorithm for the design of cells based on the cost model. The first step was a simple run of the ROC algorithm. During the second step potential merging of cells was considered with the help of the cost model. Further merging was attempted at the third step, based on machine capacity constraints and possible cost savings. While the proposed cost model proposed by the authors was not in their own words 'ultimate', it provided a realistic approximation of costs involved in a modern manufacturing environment.

Steudel & Ballakur (1987) introduced a similarity measure called Cell Bond Strength which utilised information about the processing times and production requirements of parts in order to determine a level of similarity between a pair of machines. Dynamic Programming (Held & Karp, 1962) was employed for the identification of sequences of machines that possessed the highest sum of cell bond strengths between them. The sequences were divided into a number of cells by using two alternative heuristic procedures.

Harhalakis *et al.* (1990) proposed a two-step heuristic solution procedure which explicitly considered the operation sequences of parts. The first stage of the procedure clustered the machines aggregately based on the calculation of the 'Normalised Intercell Traffic' measure. The second step further refined the solutions aiming to maximise the intercell traffic of the proposed configuration. The authors additionally introduced three new measures for the evaluation of cell-formation solutions, *global efficiency*, *group efficiency* and *group technology efficiency*.

Boe & Cheng (1991) presented a simple but efficient heuristic procedure for the clustering of machines and parts in a binary m/c matrix. A measure of similarity for each pair of machines in the plant was calculated based on the 'closeness' of their part routings. The rows of the m/c matrix were then reorganised by bringing 'similar' machines closer together. The intermediate matrix was diagonalised by rearranging the columns using a simple heuristic procedure. The authors compared the performance of the algorithm with some well-known clustering procedures on a number of test problems taken from the literature and concluded that its performance was always equal or better in terms of the grouping efficiency measure.

Kusiak (1991) introduced three branch & bound algorithms for the solution of binary cell formation problems. The procedure was based on the use of the CIA algorithm for the creation of primitive cells. If CIA was not able to create mutually independent cells, child nodes where created by removing columns of the m/c matrix. A child node was selected through a purpose-based procedure and CIA was re-applied to that node. The

algorithm continued in the same fashion until a child node with clearly visible cells was identified. Kusiak suggested extensions to this branching scheme which accommodated the existence of cell-size constraints and the identification and screening of bottleneck machines on a pre-processing stage.

Kusiak & Cho (1992) introduced two similarity coefficients for the identification of manufacturing cells when alternative process plans are available. The first coefficient assigned a zero-one similarity to process plans according to their processing requirements. The similarity matrix was employed by a graph-based branching procedure for the determination of cells. This coefficient was suitable for cases where complete disjoint cells could be formed. For alternative cases the authors proposed another zero-one similarity coefficient which considered the level of similarity between two process plans. The similarity matrix was then fed to CIA which produced the final configuration.

Okogbaa *et al.* (1992) presented a cell-formation heuristic based on the calculation of the flow of parts between machines. Initially, a number of seed machines were identified, and machine cells were formed around them. Inter-machine flow was used for the calculation of a safety combination rate which determined the admission of a machine to a certain cell. Parts were then allocated to cells and the design was refined by considering the objective of the optimisation procedure. The authors illustrated the performance of the algorithm on some test problems taken from the literature, but most interestingly they compared the benefits of applying different layout designs on a manufacturing case study. A GT layout, a classic process layout and a modified GT layout which explicitly considered the balancing of load between machines of the same type were used in the comparison. The results showed that while the GT layout decreased set-up times and average distance travelled per move, the process layout produced shorter throughput times. However, the modified GT layout outperformed both alternative layouts showing that the non-consideration of load balancing during the cell-formation procedure could result in suboptimal designs.

Heragu & Gupta (1994) developed a heuristic procedure for the solution of a constrained version of the cell-formation problem. A number of design constraints that reflected practical considerations had to be respected by a potential solution. These constraints included the available capacity of machines, safety requirements and cell-size bounds. The algorithm started by calculating the number of machines required for each type in order to satisfy the technological requirements. Then, a four-step heuristic procedure was employed for the design of manufacturing cells. The authors illustrated the performance of the algorithm on some problems taken from the literature and introduced the *flow efficiency* measure for the evaluation of solutions.

Beaulieu *et al.* (1997) also presented a two-stage algorithm for the solution of a cell formation problem with alternative process plans for parts. The first stage of the procedure was a cell aggregation algorithm which employed information about machine flexibility and alternative part routings for the creation of independent cells. The second stage introduced intercell moves by reassigning machines to different cells in an attempt to minimise the annual operating costs. The authors illustrated the computational efficiency of the algorithm but no comparisons were presented with alternative solution procedures.

Another multi-stage heuristic algorithm for the creation of manufacturing cells was introduced by Lee & Chen (1997). The authors considered a multiobjective version of the problem which aimed to simultaneously minimise the number of intercell moves

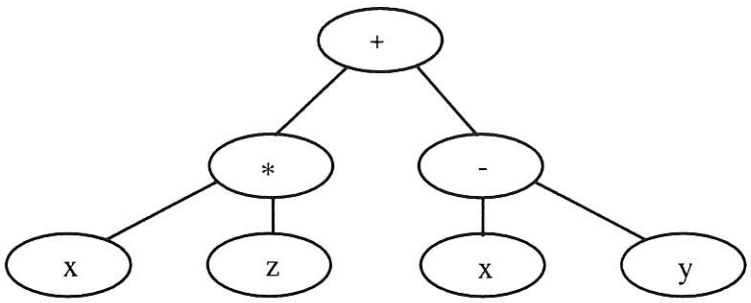
and balance the load on duplicated machines. A weighted-sum approach was employed for the optimisation of both objectives.

In a recent paper Da Silveira (1999) presented a methodology for the practical implementation of a cellular manufacturing system. The procedure was based on an application that took part in a toy manufacturing plant in Brasil. The benefits of the implementation in terms of reduction in scrap, rework, work-in-progress, stock, and delivery times were quite significant. Boe & Cheng's Close Neighbour Algorithm was used for the creation of machine cells and part families.

4 Brief Introduction to Genetic Programming

Genetic Programming (Koza, 1991), is an evolutionary computation method that evolves solutions in the form of computer programs, i.e. structures that can be compiled directly or with slight modifications by a computer. In that sense GP is a form of automatic programming.

The intuition behind Genetic Programming (GP) is that every solution to a problem can be represented by a computer program which takes certain inputs and produces the required outputs. In GP terminology, inputs are usually described as 'terminals'. The user specifies a number of 'functions' that manipulate terminals. The outputs can either be a direct result of this manipulation or a side effect, depending on the nature of the problem.



$$= x * z + (x - y)$$

Figure 2: An example of a GP parse tree and its interpretation

Once the appropriate coding of the solution has been defined, GP operates in a similar way to other evolutionary algorithms. A population of programs is randomly created and a fitness value is assigned to each of them according to their performance on the problem considered. Genetic operators are applied to programs which have been probabilistically selected based on their fitness, and a new generation of programs is created. The same procedure is repeated until a termination criterion has been reached.

The length of computer programs is not fixed and can change considerably during the evolutionary procedure. This feature provides GP with a considerable amount of flexibility in comparison with other EA's where the solution is usually coded as a fixed-length string of binary or real numbers.

The genetic operators employed by the majority of GP researchers are subtree crossover and subtree mutation. Since computer programs in GP are usually represented as parse-trees (fig.2), subtree crossover works by exchanging randomly selected branches of the program tree. In subtree mutation, a randomly selected branch

of the tree is deleted and replaced by a randomly created new branch. The application of genetic operators increases the length of computer programs considerably. A size constraint is usually imposed on the maximum depth of evolved trees.

For a thorough description of GP and a more detailed discussion on the recent developments in the field, the interested reader should refer to Banzhaf *et al.* (1998).

5 A Genetic Programming-based methodology for the solution of the cell-formation problem

5.1 Introduction

Genetic Programming applications for the solution of manufacturing optimisation problems have rarely been reported. In this section we present a novel GP methodology for the solution of the cell-formation problem. Since the proposed methodology is based on McAuley's Single Linkage Cluster Analysis (SLCA) algorithm, we will start by illustrating the application of SLCA on a small example problem.

5.2 McAuley's SLCA Algorithm

As we discussed earlier McAuley (1972) introduced the first hierarchical clustering algorithm for the solution of the cell-formation problem. The algorithm was based on the calculation of a similarity measure for each pair of machines in the plant, which was used for the creation of a pictorial representation of the solutions in the form of a 'dendrogram'. We will describe the operation of the algorithm using the m/c matrix shown in Table 4:

	p1	**p2**	**p3**	**p4**	**p5**
m1	1		1		
m2		1		1	1
m3	1		1		
m4	1	1		1	

Table 4: Example m/c matrix for the illustration of SLCA

The algorithm starts with the calculation of similarity coefficients for each pair of machines. McAuley employed Jaccard's similarity coefficient which, for this particular problem, was defined as follows:

$$S_{ij} = \frac{a_{ij}}{a_{ij} + b_{ij} + c_{ij}}$$

where:

S_{ij} : similarity between machines i and j

a_{ij} : number of parts processed by both machines i and j

b_{ij} : number of parts processed by machine i but not by machine j

c_{ij} : number of parts processed by machine j but not by machine i

The value of the similarity coefficient ranges from 0 to 1. For the above example the similarities are calculated as follows:

$$S_{1,3} = \frac{2}{2+0+0} = 1 \qquad S_{1,4} = \frac{1}{1+1+1} = 0.33 \qquad S_{1,2} = \frac{0}{0+2+3} = 0$$

$$S_{2,4} = \frac{2}{2+1+1} = 0.5 \qquad S_{3,4} = \frac{1}{1+1+2} = 0.25 \qquad S_{2,3} = \frac{0}{0+3+2} = 0$$

Thus, there is total similarity between machines 1 and 3, and no similarity between machines 1 and 2. The above values are used for the construction of the similarity matrix:

	m1	**m2**	**m3**
m2	0	*	*
m3	1	0	*
m4	0.33	0.5	0.25

Table 5: Similarity matrix for the example problem

The calculated similarity matrix is used for the creation of a pictorial representation of solutions which is known as a 'dendrogram'. Single Linkage Cluster Analysis is employed for the construction of the dendrogram. SLCA assumes that all machines are initially ungrouped. Then, the pair of machines having the highest value in the similarity matrix is recorded and is grouped at this level of similarity. In our example, machines 1 and 3 are grouped at the similarity level of 1. The next highest similarity level is found and the associated machines are merged at this level. In our case, machines 2 and 4 are merged at the similarity level of 0.5. At this point the highest similarity level is 0.33 between machines 1 and 4. Since both machines have already been grouped, their associated groups are merged as well. Thus, at the similarity level of 0.33 all machines have formed a single cell and consequently there is no reason for examining the remaining similarity coefficient values. The constructed dendrogram is illustrated in figure 3:

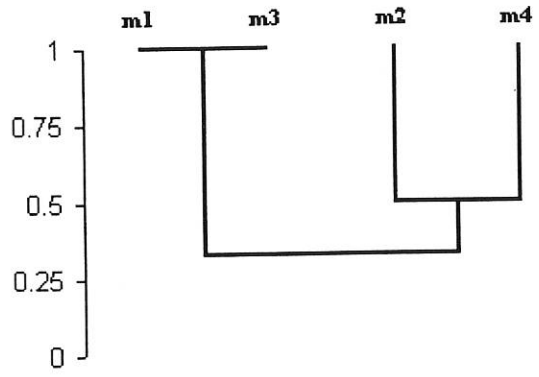


Figure 3: Dendrogram of solutions for the example problem

The above dendrogram contains a number of alternative solutions, depending on the value of the threshold level (T) that we wish to choose and the objective of the algorithm. More specifically, there are four alternative solutions:

- **Solution 1** (initial) *cell 1*: m_1
  cell 2: m_2
  cell 3: m_3
  cell 4: m_4
- **Solution 2** (T=1) *cell 1*: m_1, m_3
  cell 2: m_2
  cell 3: m_4
- **Solution 3** (T=0.5) *cell 1*: m_1, m_3
  cell 2: m_2, m_4
- **Solution 4** (T=0.33) *cell 1*: m_1, m_2, m_3, m_4

McAuley used as an objective the minimisation of the sum of material handling costs which is calculated by adding the intercellular and intracellular handling costs under a prespecified layout. However, since the output of SLCA is a partition of machines into a number of cells, it can be used in conjunction with any desired objective.

The main disadvantage of SLCA is the occurrence of the 'chaining' phenomenon during the grouping procedure. Since the existence of a single linkage between machines or groups of machines is enough for the approval of merging, the algorithm may bring together machines with low similarity. In our example problem, machines 1 and 2 are grouped together at the similarity level of 0.33, while their similarity coefficient is 0. McAuley, as well as other researchers (Gupta & Seifoddini, 1990) have proposed the use of alternative clustering methods like Average Linkage Cluster Analysis (ALCA) and Complete Linkage Cluster Analysis (CLCA) as a cure for this problem. ALCA calculates the average of similarity coefficients between groups of machines. CLCA works in the opposite way of SLCA by assigning the lowest and not the highest similarity coefficient between groups of machines. However, both ALCA and CLCA require the recalculation of the similarity matrix after each individual merging step, resulting in greater computational complexity.

5.3 Genetic Programming methodology

Finding an optimal solution to the cell-formation problem is not an easy task. Garcia-Diaz & Lee (1993) indicate that the number of p non-empty partitions of n objects is equal to :

$$S(n,p) \approx \frac{p^n}{p!}$$

where $S(n,p)$ is the Stirling number. For n=20 the number of non-empty subsets of size 5 is approximately 7.94×10^{11}. If the number of clusters is not prespecified, the total number of partitions becomes equal to:

$$\sum_{j=1}^{n} S(n,j)$$

This formula rules out a complete enumeration of solutions. The advantage of methods that employ similarity coefficients is that they narrow down the search by focusing on regions of the solution space that are most likely to contain an optimal solution.

However, it has been indicated (Sarker, 1986) that the performance of similarity coefficients depends heavily on the characteristics of the problem and the grouping measure that is used for the evaluation of solutions. Jaccard's similarity coefficient is less likely to find an optimal solution in the case of ill-structured matrices, since the grouping of machines becomes less straightforward. In addition, SLCA is a 'blind' method, i.e. it produces the same dendrogram irrespective of the optimisation objective. Our methodology employs Genetic Programming for the evolution of similarity coefficients that are used by SLCA for the construction of the dendrogram of solutions. The operation of GP-SLCA in algorithmic form is the following:

Procedure Main

initialise population of randomly created similarity coefficients
run procedure SLCA for each coefficient
loop
    loop
        select individuals for crossover or mutation
        apply genetic operators and form new coefficients
    until a new generation has been formed
    run procedure SLCA for each coefficient
until termination criterion is true

Procedure SLCA

compute similarity matrix
construct dendrogram
loop
    create machine cells for the highest level of similarity coefficient
    assign parts to machine cells
    calculate the fitness value of the cell configuration
    if solution is the best recorded so far, best=current solution
    delete the value of similarity coefficient from the matrix
until a single cell has been formed
assign the best solution as the fitness of the individual

It is interesting to note that GP indirectly evolves a dendrogram of solutions since given the SLCA algorithm, each evolved coefficient corresponds to a particular dendrogram. It is also obvious that as the size of the problem increases the evaluation function becomes computationally expensive. SLCA was employed in order to reduce the computational requirements, since the use of ALCA or CLCA would result in additional overhead.

The methodology described in this section belongs to the category of cell-formation methods that group machines into cells and not parts into families (see section 2). Thus, for each machine-cell configuration created from the dendrogram, the corresponding part families must be formed in order to calculate the value of the objective function. Since no information is available about the sequencing of operations (backtracking, skips, etc.), parts are usually assigned to the cell where the majority of their processing takes part. In case of a tie, the part is assigned to the smallest of the candidate cells. In that way we ensure that the number of voids created by the assignment procedure is

minimal. If there is still a tie, the part is assigned randomly to one of the candidate cells. If the allocation of parts to machine cells results in the creation of an empty cell (a cell that processes no parts), then the fitness of this solution is set to 0. However, there is no limit on the size of machine cells, and consequently no limit on the total number of cells in the plant. If required, the algorithm has the ability to explicitly consider size constraints, by just assigning a zero fitness to each solution that violates the constraints.

The characteristics of the GP algorithm used in the above formulation are the following:

Functions: The four standard arithmetic operations {+, -, ×, %} were used for the manipulation of terminals. Note that the division operator '%' corresponds to the protected division function. This function returns the value of '1' if the denominator of the division is equal to '0', so that the closure property of the set is not violated.

Terminals: The following four terminals are used for the construction of the similarity coefficient:

a_{ij} : number of parts processed by both machines i and j

b_{ij} : number of parts processed by machine i but not by machine j

c_{ij} : number of parts processed by machine j but not by machine i

d_{ij} : number of parts processed by neither machine i nor machine j

Note that with the exception of d_{ij} the same variables were used for the calculation of Jaccard's similarity coefficient. McAuley did not include this variable since its value is usually quite high, thus it would create very small values of the coefficient. In our case, the structure of the evolved similarity coefficient is not known in advance and neither is the significance of d_{ij} in the construction of a fit coefficient. In theory, the evolutionary procedure is robust enough to leave out of the final solution any terminal (variable) that is irrelevant to the solution of the problem. The range of values for the evolved similarity coefficient is not known in advance. However, this does not change the operation of the hierarchical clustering procedure, since it does nor require a specific range of similarity values. For ease of illustration we have included a function that normalises the values of the similarity matrix within the region 0 to 1, using the minimum and maximum values of coefficients. The operation of SLCA on both matrices yields the same result

Operators: Subtree crossover and subtree mutation were employed for the creation of genetic diversity in the population of coefficients. These operators were applied with a probability of 90% and 10% respectively in each generation.

Fitness measure: Two different fitness measures were used for the evaluation of the similarity coefficients. Both measures assess the quality of block diagonalisation and have been extensively used by researchers to illustrate the performance of cell-formation algorithms. The following notation is essential for understanding the calculation of the fitness measure (notation taken from Ng (1993)):

n: total number of columns (parts)

m: total number of rows (machines)

e: total number of non-zero entries in the m/c matrix

e_1: total number of non-zero entries inside the diagonal blocks

e_0: total number of non-zero entries outside the diagonal blocks (exceptional elements)

e_v: total number of zero entries inside the diagonal blocks (voids)

d_1: total number of elements inside the diagonal blocks

d_0: total number of elements outside the diagonal blocks

The grouping efficiency, η, of a diagonalised matrix is calculated using the following formula (Chandrasekharan & Rajagopalan, 1986b):

$$\eta = q \cdot (e_1/d_1) + (1-q) \cdot \left[1 - \left(\frac{e_0}{d_0}\right)\right]$$

where: $0 \le q \le 1$

In the case of a single cell configuration d_1 is equal to 0, thus the following formula should be used:

$$\eta = \frac{q \cdot e}{(m \cdot n) + (1-q)}$$

Grouping efficiency has two main drawbacks:

- If the value of the weight q used is equal to 0.5, it can be shown that the elimination of voids becomes much more important than the elimination of exceptional elements. However, in practical situations, exceptional elements are more costly to handle. A remedy that has been suggested is to set the value of the weight to 0.2, however, the majority of reported results have been taken with q=0.5. In order to be able to perform meaningful comparisons, we employed the same value of 0.5 in our experimental set-up.
- The value of grouping efficiency is always greater than 75%, independent of the structure of the diagonalised matrix. It is thus not a good reflection of the real quality of diagonalisation.

The results reported in the following section indicate that the maximisation of grouping efficiency does not necessarily result in an efficient solution from the cellular manufacturing point of view.

The grouping efficacy measure, Γ, is calculated using the following formula (Kumar & Chandrasekharan, 1990):

$$\Gamma = 1 - \frac{e_v + e_0}{e + e_v} = \frac{e - e_0}{e + e_v}$$

Grouping efficacy has been used by a considerable number of researchers. It is a much better grouping measure than grouping efficiency. However, as Ng (1993) showed, it assigns excessive importance to the elimination of exceptional elements. He suggested a weighted version of the measure (weighted grouping efficiency, γ), which is calculated using the following formula:

$$\gamma = \frac{q \cdot (e - e_0)}{q \cdot (e + e_v - e_0) + (1 - q) \cdot e_0}$$

where: $0 \leq q \leq 1$

If the weight is set to 0.5, then $\gamma = \Gamma$. Ng showed that if the weight is set to 0.2, γ assigns realistic importance to the existence of exceptional elements and voids in the m/c matrix.

While these two grouping measures were employed for the evaluation of solutions, any other grouping measure could have been used in conjunction with the algorithm (for a review of the grouping measures in cellular manufacturing, see (Sarker & Mondal, 1999). One of the advantages of this methodology is that it is flexible enough to work with any objective function chosen by the user. The only consideration should be that available information is enough for the calculation of the objective value.

Parameters of the run: Apart from the values of crossover & mutation probability, a number of additional parameters should be defined when using Genetic Programming. The values of these parameters are illustrated in fig.4, which is commonly referred to as the Koza-tableau.

Parameters	Values
Objective:	maximisation of the grouping efficiency or grouping efficacy of a diagonalised matrix
Terminal set:	a, b, c, d (defined earlier)
Function set:	+, -, ×, %
Population size:	500
Subtree crossover probability:	.9
Subtree mutation probability:	.1
Selection:	Tournament selection, size 7
Number of generations:	50
Maximum depth for crossover:	17
Initialisation method:	Ramped half and half

Figure 4. Koza tableau for the GP-SLCA methodology

6 Application of the GP-SLCA algorithm for the solution of individual cell-formation problems

6.1 Experimental set-up

Finding a set of problems for the evaluation of an optimisation method is always a difficult task. The main requirements that a representative set of test problems should fulfil are the following:

- Different instances of the problem should be included in terms of size, difficulty or any other parameter that can be varied
- Results from alternative solution methods should be available, so that meaningful comparisons can be made.

In the case of cell-formation problems, there is no formal definition of the difficulty of a particular instance of the problem. As we discussed in section 3.4, Chandrasekharan & Rajagopalan (1989) investigated the characteristics of binary m/c matrices but their conclusions were based on matrices of fixed size. The lack of a parameter-based estimation of the difficulty of the problem means that the creation of randomly

generated problems is not as straightforward as in the case of other optimisation problems. In practice, researchers of the cell-formation problem evaluate their methods on test-problems taken from the literature, so, there are many comparative results available. While the second requirement for an appropriate set of test problems is fulfiled, there should be a careful qualitative consideration of the cases that will be chosen in order to be as close as possible to the fulfilment of the first requirement as well.

In this report we have employed 27 problems for the testing of the GP-SLCA methodology. All the problems have been taken from the cellular manufacturing literature and results from alternative cell-formation methods have been reported. The size of the problems ranges from 10×15 to 40×100. The problems have been chosen so that they represent different levels of difficulty as this is indicated from the reported results. All these problems along with their characteristics and their corresponding references are described in table 7. The number in the first column of the table will be used from this point onwards for the identification of these problems. For the reader interested in using the test problems for their own comparisons we should note that problems 1-8 correspond to problems 1-6, 8, 9 in the order presented by Boctor, and problems 16-21 correspond to problems 1-3, 5-7 in the order prresented by Chandrasekharan & Rajagopalan. All test problems are available in text file format and can be provided by the authors of this report.

The GP-SLCA framework was initially applied on all test problems using each of them as an individual fitness case. Two different optimisation objectives were employed, grouping efficiency and grouping efficacy. Twenty runs of GP-SLCA were conducted for each problem and each individual objective. The number and type of experiments that were conducted are summarised in table 6:

Objective	No. of test problems	No. of runs per problem
maximisation of grouping efficiency	27	20
maximisation of grouping efficacy	27	20

Table 6. Number and type of experiments

No.	Reference	Size	e
1	Boctor (1991)	16×30	121
2	"	16×30	106
3	"	16×30	92
4	"	16×30	111
5	"	16×30	107
6	"	16×30	101
7	"	16×30	114
8	"	16×30	118
9	Boe & Cheng (1991)	20×35	153
10	Burbidge (1975)	16×43	126
11	Carrie (1973)	20×35	136
12	Chan & Milner (1982)	10×15	46
13	Chandrasekharan & Rajagopalan (1987)	40×100	420
14	Chandrasekharan & Rajagopalan (1986)	8×20	91
15	Chandrasekharan & Rajagopalan (1986)	8×20	61
16	Chandrasekharan & Rajagopalan (1989)	24×40	131
17	"	24×40	130
18	"	24×40	131
19	"	24×40	131
20	"	24×40	131
21	"	24×40	130
22	Kumar et al. (1986)	23×20	113
23	Kumar & Vannelli (1987)	30×41	128
24	Seifoddini (1989)	11×22	78
25	Stanfel (1985)	14×24	61
26	Stanfel (1985)	30×50	154
27	Stanfel (1985)	30×50	167

Table 7: Test problems

6.2 Results

Before we present the cumulative results of all test problems, it will be interesting to take a closer look at a particular example, so that the operation of the methodology will become clear. The well-known 16×43 m/c component matrix, originally introduced by Burbidge (1975), is presented in fig. 5. Almost every algorithm for the solution of the binary cell-formation problem has been illustrated on this test problem.

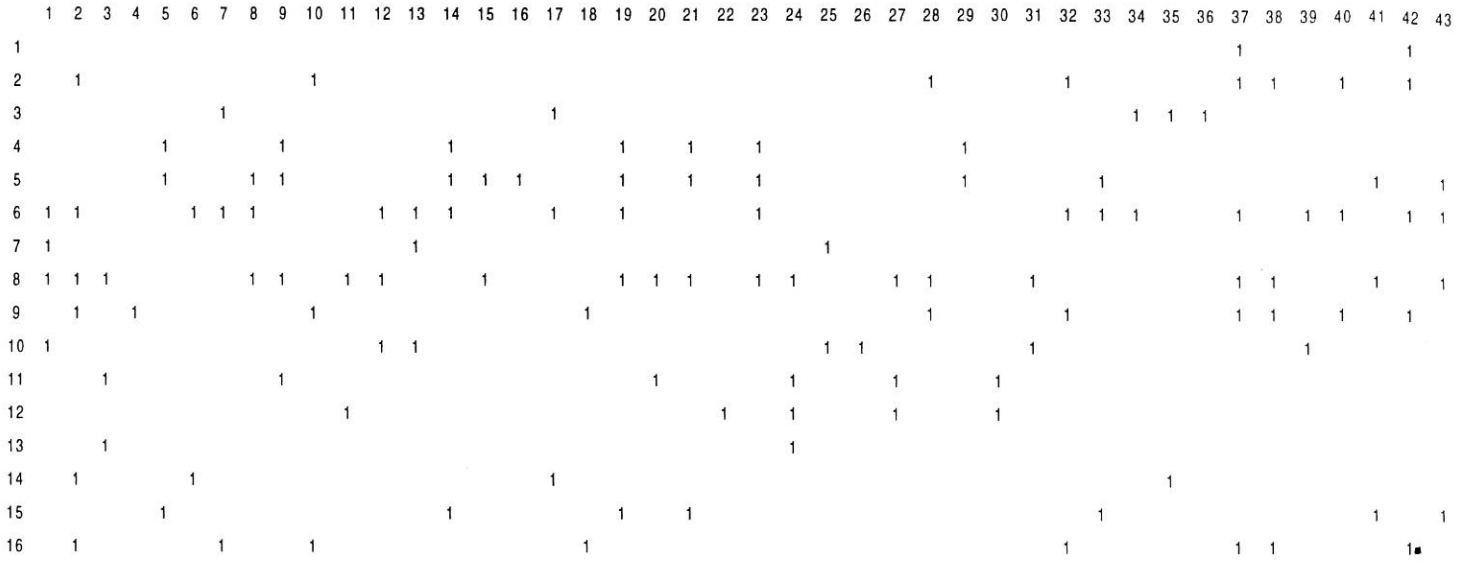


Figure 5: 16×43 test problem introduced by Burbidge

20 runs of the GP-SLCA methodology were performed for this test problem for each of the objectives. The results of the experiments are summarised in the following tables:

Best value of efficiency recorded	**0.935761**
Number of times this value was found	**2**
Mean best value of efficiency per run	**0.932657**
Standard deviation	**0.003324**

Table 8: Grouping efficiency results

Best value of efficacy recorded	**0.567901**
Number of times this value was found	**5**
Mean best value of efficacy per run	**0.566063**
Standard deviation	**0.003658**

Table 9: Grouping efficacy results

The similarity coefficient that produced the best value of grouping efficacy is the following (in LISP symbolic language):

((-(+(/(d)(b))(-(+(d)(a))(/(a)(b))))(-(-(*(+(a)(a))(-
(b)(a)))(/(*(a)(d))(+(b)(c))))(/(*(+(a)(a))(-(b)(a)))(-(+(*(-

(+(a)(a))(*(c)(/(+(/(*(a)(b))(/(d)(b)))(d))(*(a)(a)))))(d))(/(+(/(-(+(d)(a))(/(/(a)(b))(-(d)(b))))(/(+(a)(b))(d)))(a))(a)))(/(+(+(d)(c))(d))(*(a)(a)))))))

The evolved program corresponds to the following formula for the coefficient:

$$a+d+\frac{d}{b}-\frac{a}{b}-2a(b-a)+\left(\frac{ad}{b+c}\right)+\frac{2a(b-a)}{d\left[2a-\frac{c\left(\frac{ab}{d/b}+d\right)}{a^2}\right]-\frac{2d+c}{a^2}+\frac{\left[a+\frac{a+d-\left(\frac{a/b}{d-b}\right)}{(a+b)/d}\right]}{a}}$$

It is not expected that Genetic Programming will evolve a similarity coefficient that will be easy to understand how and why it works. However, in this particular case we can see that the values of *a* and *d*, which indicate a similarity of processing operations between a pair of machines dominate the outcome of the formula. High values of a and d will result in high values for the similarity coefficient. That does not mean that the evolved coefficient is ideal for every cell-formation case. Keep in mind that the fitness of an individual is measured based on the best solution found from the dendrogram. It is always possible that an evolved coefficient which does not utilise the variables in a meaningful way from the cellular manufacturing point of view, will produce a fit partition of machines for the objective considered. The above formula is used for the calculation of similarities for each pair of machines in the plant, resulting in the similarity matrix in fig.6. The normalised similarity matrix is presented in fig.7.

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14	m15
m2	56.6724														
m3	54.0000	33.7500													
m4	51.0000	31.5000	37.2000												
m5	42.0000	24.7500	30.0000	169.7049											
m6	36.8271	57.9114	44.1750	24.2409	11.1468										
m7	57.0000	36.0000	42.0000	37.7143	29.0769	-31.5418									
m8	45.1000	30.5500	21.6000	41.5745	84.2238	-26.7635	30.9998								
m9	51.2545	300.7515	33.6000	29.7143	21.5385	-61.2140	40.0000	-71.3682							
m10	51.0000	31.5000	37.2000	33.1429	24.7692	-58.3231	84.0851	-58.5406	28.6000						
m11	52.5000	32.6250	38.4000	29.8161	7.4648	18.9474	45.3333	-64.3697	29.7000	34.2857					
m12	54.0000	33.7500	39.6000	35.4286	26.9231	20.0000	46.6667	-56.1498	30.8000	35.4286	69.6667				
m13	58.5000	37.1250	43.2000	38.8571	30.1538	23.1579	50.6667	-34.6101	34.1000	38.8571	58.2819	48.3946			
m14	55.5000	28.6247	59.7232	36.5714	28.0000	483.5966	48.0000	-13.1068	20.7166	36.5714	38.5000	40.8000	55.5000		
m15	51.0000	31.5000	37.2000	76.5809	89.8009	-58.3231	44.0000	-67.7242	28.6000	33.1429	35.0000	37.2000	51.0000	40.0000	
m16	56.6724	149.8667	36.3173	32.0000	23.6923	-54.6544	42.6667	-59.8156	159.2070	32.0000	33.8333	36.0000	49.5000	42.5328	32.0000

Figure 6 Similarity coefficient matrix

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13	m14	m15
m2	0.2307														
m3	0.2259	0.1894													
m4	0.2205	0.1854	0.1956												
m5	0.2043	0.1732	0.1827	0.4344											
m6	0.1950	0.2330	0.2082	0.1723	0.1487										
m7	0.2313	0.1935	0.2043	0.1966	0.1810	0.0718									
m8	0.2099	0.1836	0.1675	0.2035	0.2804	0.0804	0.1845								
m9	0.2210	0.6705	0.1891	0.1821	0.1674	0.0183	0.2007	0.0000							
m10	0.2205	0.1854	0.1956	0.1883	0.1732	0.0235	0.2801	0.0231	0.1801						
m11	0.2232	0.1874	0.1978	0.1823	0.1421	0.1627	0.2103	0.0126	0.1821	0.1904					
m12	0.2259	0.1894	0.2000	0.1924	0.1771	0.1646	0.2127	0.0274	0.1841	0.1924	0.2541				
m13	0.2340	0.1955	0.2064	0.1986	0.1829	0.1703	0.2199	0.0662	0.1900	0.1986	0.2336	0.2158			
m14	0.2286	0.1802	0.2362	0.1945	0.1791	1.0000	0.2151	0.1050	0.1659	0.1945	0.1980	0.2021	0.2286		
m15	0.2205	0.1854	0.1956	0.2666	0.2904	0.0235	0.2079	0.0066	0.1801	0.1883	0.1917	0.1956	0.2205	0.2007	
m16	0.2307	0.3986	0.1940	0.1863	0.1713	0.0301	0.2055	0.0208	0.4155	0.1863	0.1896	0.1935	0.2178	0.2052	0.1863

Figure 7 Normalised similarity coefficient matrix

If either of these matrices is fed as an input to the SLCA algorithm a dendrogram of potential solutions will be produced. In fig.8 you can see the dendrogram for this particular case which has been cut at the similarity level of 0.2340. The cell configuration for this instance of the dendrogram corresponds to the diagonalised matrix of fig.9, which has a grouping efficacy value of 0.5679.

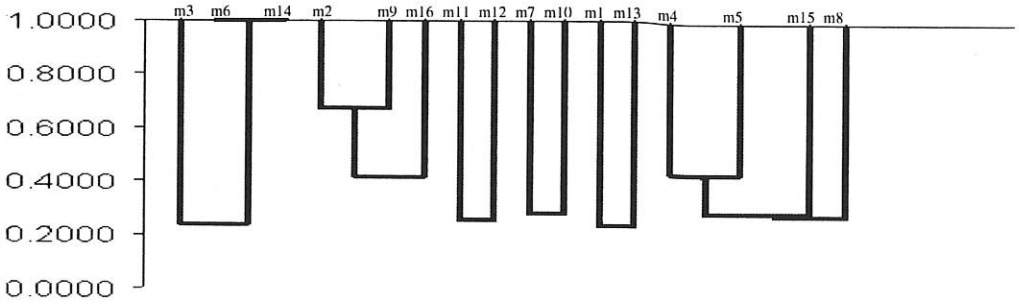


Figure 8: Dendrogram cut at the similarity level 0.2340

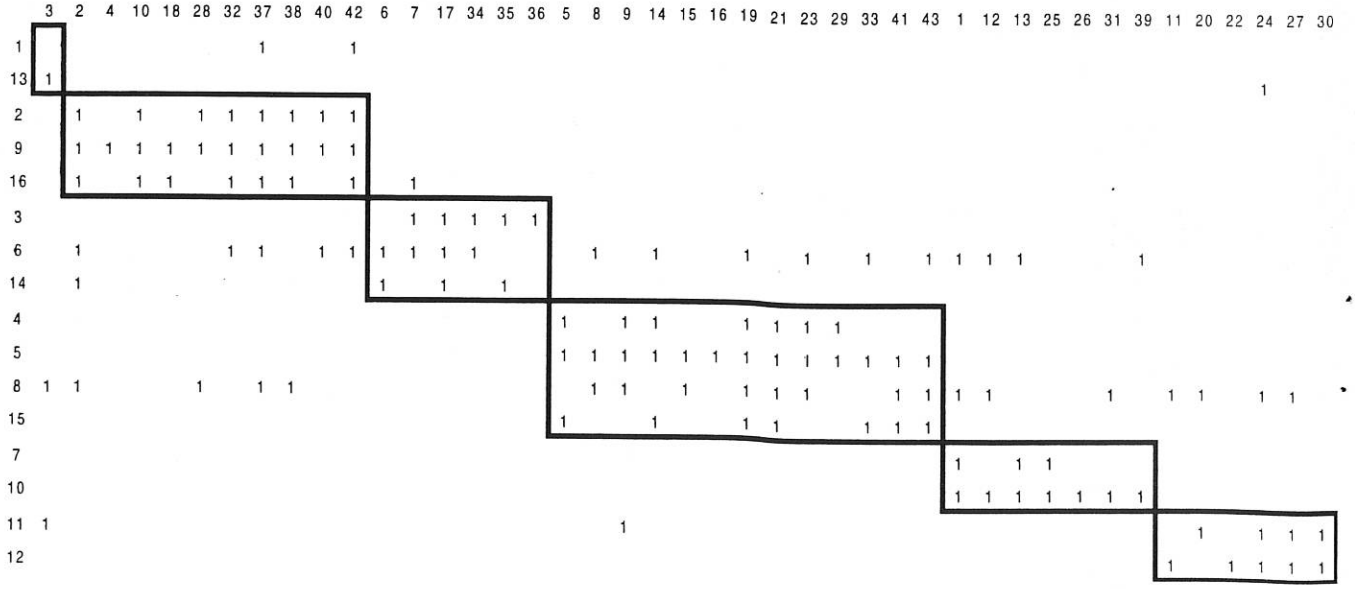


Figure 9: Diagonalised matrix (grouping efficacy)

This value of grouping efficacy is one of the best of those that have been reported in the literature for this particular problem. The performance of the algorithm is equally good in all other problems of the experimental set-up. The cumulative results are presented in two parts. First, in table 10, the detailed results of the GP-SLCA procedure are illustrated. Then, in tables 11 & 12, the best solution evolved by GP-SLCA is compared with a number of solutions that have been produced by alternative cell-formation methods. All diagonalised matrices for both objectives are available on request.

Pr. No.	max η	$\overline{\eta}$	σ	e_0	e_v	No. of cells	max Γ	$\overline{\Gamma}$	σ	e_0	e_v	No. of cells
1	**0.917**	**0.909**	0.0051	71	0	10	**0.509**	**0.503**	0.0631	40	38	5
2	**0.935**	**0.903**	0.0146	56	0	10	**0.618**	**0.618**	0	22	30	6
3	**0.952**	**0.949**	0.0024	44	0	10	**0.7**	**0.7**	0	8	28	4
4	**0.926**	**0.924**	0.0007	64	0	10	**0.496**	**0.493**	0.0029	31	40	6
5	**0.930**	**0.926**	0.0032	52	1	9	**0.727**	**0.727**	0	11	25	4
6	**0.938**	**0.933**	0.0023	54	0	11	**0.782**	**0.782**	0	18	18	5
7	**0.930**	**0.926**	0.0020	60	0	10	**0.595**	**0.590**	0.0063	23	39	4
8	**0.927**	**0.913**	0.0058	54	1	9	**0.774**	**0.774**	0	12	19	4
9	**0.930**	**0.915**	0.0107	90	0	13	**0.568**	**0.568**	0	40	46	5
10	**0.940**	**0.933**	0.0033	77	0	13	**0.568**	**0.566**	0.0037	34	36	6
11	**0.944**	**0.916**	0.0133	62	1	11	**0.767**	**0.766**	0.0016	11	27	6
12	**0.96**	**0.96**	0	0	4	3	**0.92**	**0.92**	0	0	4	3
13	**0.964**	**0.956**	0.0029	88	17	13	**0.840**	**0.840**	0	36	37	10
14	**0.788**	**0.788**	0	51	0	4	**0.587**	**0.587**	0	27	18	2
15	**0.958**	**0.958**	0	9	0	3	**0.852**	**0.852**	0	9	0	3
16	**1**	**1**	0	0	0	7	**1**	**1**	0	0	0	7
17	**0.967**	**0.964**	0.0040	31	3	10	**0.851**	**0.851**	0	10	11	7
18	**0.953**	**0.940**	0.0067	51	3	12	**0.735**	**0.735**	0	20	20	7
19	**0.961**	**0.960**	0.0011	70	0	16	**0.533**	**0.531**	0.0029	50	21	11
20	**0.961**	**0.957**	0.0042	70	0	16	**0.479**	**0.476**	0.0020	63	11	13
21	**0.930**	**0.919**	0.0045	78	3	18	**0.437**	**0.435**	0.0016	61	28	11
22	**0.846**	**0.784**	0.0291	66	8	9	**0.490**	**0.453**	0.0179	43	30	5
23	**0.975**	**0.968**	0.0079	59	0	19	**0.607**	**0.607**	0.0011	46	7	16
24	**0.917**	**0.917**	0	28	1	6	**0.731**	**0.731**	0	10	15	3
25	**0.957**	**0.954**	0.0008	26	0	10	**0.718**	**0.718**	0.0010	10	10	7
26	**0.963**	**0.960**	0.0063	70	2	20	**0.594**	**0.583**	0.0063	53	16	14
27	**0.966**	**0.944**	0.0205	96	0	22	**0.5**	**0.488**	0.0044	75	17	15

Table 10. Performance of GP-SLCA on individual test problems

Pr.No.	GP-SLCA	ZODIAC *(Chandr. & Raj., 1987)*	GRAFICS *(Sriniv.& Naren., 1991)*	Assignment *(Srinivasan et al., 1994)*	MST *(Ng, 1993)*	GA-TSP *(Cheng et al., 1998)*
1	0.917	0.643	0.772	0.846	-	-
2	0.935	0.795	0.816	0.810	-	-
3	0.952	0.858	0.869	0.858	-	-
4	0.926	0.586	0.764	0.730	-	-
5	0.930	0.881	0.901	0.881	-	-
6	0.938	0.896	0.908	0.891	-	-
7	0.930	0.636	0.791	0.799	-	-
8	0.927	0.907	0.907	0.907	-	-
9	0.930	0.776	-	-	-	0.796
10	0.940	0.802	0.794	0.776	-	0.794
11	0.944	0.878	0.878	-	0.945	0.878
12	0.96	0.96	0.96	-	0.96	0.96
13	0.964	0.951	0.951	-	0.974	0.951
14	0.788	0.719	0.763	-	-	0.719
15	0.958	0.958	0.958	-	0.958	0.958
16	1	1	1	-	1	1
17	0.967	0.952	0.952	-	0.975	0.952
18	0.953	0.908	0.912	-	-	0.908
19	0.961	0.773	0.789	0.856	-	0.836
20	0.961	0.724	0.791	0.833	-	0.853
21	0.930	0.693	0.791	0.761	-	0.811
22	0.846	0.670	0.762	0.721	-	0.814
23	0.975	0.681	0.823	0.865	-	0.824
24	0.917	0.878	0.878	-	-	-
25	0.957	0.839	0.839	-	-	0.841
26	0.963	0.754	0.852	-	-	0.860
27	0.966	0.629	0.856	-	-	0.822

Table 11. Comparison with alternative cell-formation methods (grouping efficiency)

Pr.No.	GP-SLCA	ZODIAC *(Chandr. & Raj., 1987)*	GRAFICS *(Sriniv.& Naren., 1991)*	Assignment *(Srinivasan et al., 1994)*	MST *(Ng, 1993)*	GA-TSP *(Cheng et al., 1998)*
1	0.509	0.349	0.481	0.447	-	-
2	0.618	0.586	0.534	0.508	-	-
3	0.7	0.686	0.675	0.644	-	-
4	0.496	0.267	0.449	0.407	-	-
5	0.727	0.727	0.691	0.727	-	-
6	0.782	0.764	0.771	0.760	-	-
7	0.595	0.320	0.579	0.530	-	-
8	0.774	0.774	0.774	0.774	-	-
9	0.568	0.511	-	-	-	0.551
10	0.568	0.538	0.544	0.471	-	0.539
11	0.767	0.751	0.751	-	0.767	0.753
12	0.92	0.92	0.92	-	0.92	0.92
13	0.840	0.839	0.839	-	0.831	0.840
14	0.587	0.583	0.581	-	-	0.583
15	0.852	0.852	0.852	-	0.852	0.852
16	1	1	1	-	1	1
17	0.851	0.851	0.851	-	0.851	0.851
18	0.735	0.730	0.735	-	-	0.730
19	0.533	0.204	0.433	0.446	-	0.494
20	0.479	0.182	0.445	0.439	-	0.447
21	0.437	0.176	0.417	0.335	-	0.425
22	0.490	0.387	0.494	0.436	-	0.466
23	0.607	0.337	0.554	0.559	-	0.538
24	0.731	0.731	0.731	-	-	-
25	0.718	0.656	0.656	-	-	0.674
26	0.594	0.461	0.563	-	-	0.566
27	0.5	0.211	0.480	-	-	0.459

Table 12. Comparison with alternative cell-formation methods (grouping efficacy)

6.3 Discussion

Results from tables 10, 11 and 12 indicate that GP-SLCA is a powerful algorithm for the solution of binary cell-formation problems. More specifically, for the grouping efficiency measure GP-SLCA dominates the performance of ZODIAC, GRAFICS, Assignment Algorithm and the GA-TSP heuristic. While the value of grouping efficiency is always higher than 0.9, it is far from obvious that the resulting diagonalised matrices are not ideal for the implementation of a cellular manufacturing system. Table 13 illustrates the diagonalised matrix for problem 9, which has a grouping efficiency value of 0.944.

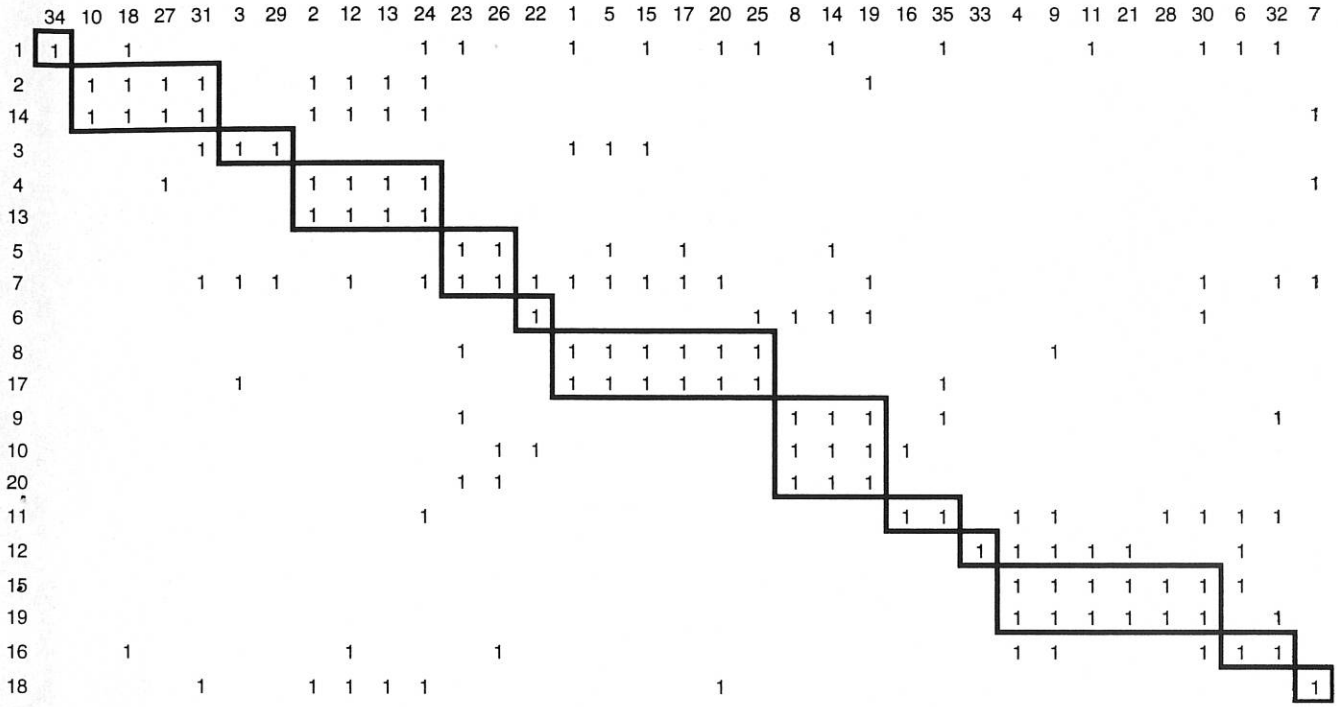


Table 13: Diagonalised matrix for problem 9 (Boe & Cheng, 1991)

The proposed grouping of machines and parts is not a realistic solution and its implementation would rather disrupt the manufacturing process. However, this is not due to the inefficiency of the GP-SLCA algorithm, which always produces solutions that maximise the desired objective. However, the maximisation of grouping efficiency does not correspond to suitable solutions for the implementation of a cellular manufacturing system. In section 5 we indicated the deficiencies of this grouping measure. Table 12 illustrates the result of assigning excessive importance to the minimisation of voids in comparison to the minimisation of exceptional elements. The algorithm minimises the number of voids by creating small compact matrices of positive elements along the main diagonal of the m/c matrix. Table 10 illustrates that all evolved GP-SLCA solutions for the maximisation of grouping efficiency focus on the elimination of voids.

This unusual configuration of the diagonalised matrices is the reason for the slightly worse performance of GP-SLCA over the MST algorithm. When a large number of small-sized cells has been formed, the optimal assignment of parts to machines becomes almost a random search procedure since there are many candidate cells that satisfy both allocation criteria. As we described earlier GP-SLCA breaks double ties randomly. In contrast MST uses a special procedure for maximising the grouping

measure by reassigning parts and machines after the initial machine cell – part family configuration has been created. This procedure is particularly useful in this situation where the assignment of parts to cells is not straightforward. However, the resulted partitions are even more impractical than the ones already described. Ng reports a final solution containing 15 cells in a 20-machine problem (pr.9). An increase in the population size of GP-SLCA will almost certainly produce similar solutions, simply because more random assignments of parts to machines will be generated. In any case, it is obvious that the grouping efficiency measure is inadequate for judging the quality of cell-formation solutions.

GP-SLCA performs equally well when the maximisation of the grouping efficacy measure is used as the objective of optimisation. Evolved solutions are always equal or better than those reported for all the comparing cell-formation methodologies. The quality of results indicate the suitability of the grouping efficacy measure for the practical implementation of cellular manufacturing. Solutions are balanced in terms of the number of voids and the number of exceptional elements. In addition, the value of grouping efficacy always gives a good indication of the quality of diagonalisation. In that way, an assessment of the difficulty of the problem can be made.

It is obvious from tables 11 and 12 that the only competitive algorithm to GP-SLCA is MST. In order to further compare the performance of these two procedures, the maximisation of the weighted grouping efficacy was employed as the optimisation objective. This measure has been introduced by Ng himself. The weight value of 0.2 was used for the calculation of the objective function. Results are presented in table 14.

Problem number	GP-SLCA	MST
11	0.732	0.732
12	0.92	0.92
13	0.680	0.680
15	0.591	0.591
16	1	1
17	0.702	0.702

Table 14: Comparison with the MST method (weighted grouping efficacy, q=0.2)

As it can be seen, both algorithms produce identical results. It is interesting that two methodologies from quite diverse backgrounds are similar in their performance. However, a more thorough investigation is needed since only a limited number of results were available for comparison. In any case, GP-SLCA has two significant advantages over MST:

- GP-SLCA can be further enhanced by increasing the population size and the number of generations per run. Fine tuning of the parameters could also result in better solutions. MST cannot be further improved in that way since there are not any variable parameters that determine the outcome of the run.
- GP-SLCA is quite flexible since it can be used for the solution of a variety of cell-formation problems by modifying the terminals and the objective function of the evolutionary procedure.

7 Evolution of general-purpose similarity coefficients

7.1 Experimental set-up

In the previous section GP-SLCA was employed for the evolution of similarity coefficients that were specific for the problem considered, i.e. the test problem was used as the only fitness case of the evolutionary procedure. The method was not focused in finding meaningful similarity coefficients but rather in diagonalising the m/c matrix using information that is considered to be relevant to the solution of the problem.

In this section we examine the possibility of evolving a similarity coefficient that has general applicability on binary cell formation problems. Evolved coefficients are evaluated on a prespecified number of test problems that are used as fitness cases. The fitness of a solution is calculated by adding the value of the objective function from each individual test problem. In that way similarity coefficients that produce good overall performance will prevail during the evolutionary procedure. Apart from that change, the basic operation of the GP-SLCA algorithm remains the same. The maximisation of grouping efficacy was used as the objective of the evolutionary procedure.

The selection of problems that will form the set of fitness cases is not straightforward since the difficulty of cell-formation problems cannot be described in terms of parameters. At the same time the evaluation function of the algorithm is computationally quite expensive, especially for large-sized problems, thus the number of test cases must be kept within certain limits depending on the available computational power.

Our experimental set-up is comprised of ten different combinations of test problems, all taken from the batch of 27 test problems described in table 7. All sets are comprised of indifferent problems in terms of their characteristics (size, grouping difficulty as it has been reported in published results, etc.) in an attempt to fulfil the requirements described in the previous paragraph. Table 15 illustrates the configuration of all sets.

Name	No. of fitness cases	Problems
SET1	8	1 – 8
SET2	6	16 - 21
SET3	6	22 – 27
SET4	6	9 – 12, 14, 15
SET5	6	1 – 4, 16 – 18
SET6	7	5 – 8, 19 – 21
SET7	8	11, 12, 14, 15, 22 – 25
SET8	8	9 – 12, 24 – 27
SET9	14	1 – 12, 14, 15
SET10	14	14 – 27

Table 15: Experimental sets for the evolution of similarity coefficients

The reasons for using a small number of fitness cases are the following:

- Since only 27 test problems are available, the set of validation problems should be large enough for the assessment of generalisation of the evolved coefficient
- Computational constraints restrict the number of fitness cases that can be evaluated in reasonable time

Both these problems are of technical nature, i.e. they can be accommodated by simply having more test cases available and by buying more computational power. Thus, the results presented in the following sub-section are just an indication of the performance of the proposed methodology rather than the best possible solutions that can be found.

7.2 Results

Twenty runs of the GP-SLCA algorithm were conducted for each experimental set-up. The cumulative results of the best coefficients evolved for each set-up are presented in table 16. For comparison reasons, the performance of Jaccard's similarity coefficient on the same set of problems is also given. Note that the outlined problems in each column denote the fitness cases that were used for the evolution of the corresponding similarity coefficient. The rest of the problems form the validation set.

Pr.no.	SET1	SET2	SET3	SET4	SET5	SET6	SET7	SET8	SET9	SET10	Jaccard
1	0.5	0.471	0.451	0.471	0.5	0.467	0.438	0.438	0.490	0.467	0.471
2	0.615	0583	0.586	0.586	0.618	0.571	0.588	0.586	0.611	0.601	0.571
3	0.7	0.698	0.698	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
4	0.474	0.459	0.240	0.455	0.489	0.470	0.409	0.231	0.467	0.475	0.474
5	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.727
6	0.752	0.752	0.752	0.752	0.752	0.752	0.752	0.752	0.752	0.742	0.752
7	0.579	0.579	0.579	0.579	0.570	0.579	0.579	0.238	0.579	0.568	0.579
8	0.773	0.773	0.773	0.773	0.773	0.773	0.748	0.748	0.774	0.774	0.774
9	0.568	0.412	0.554	0.568	0.520	0.562	0.568	0.568	0.568	0.568	0.568
10	0.544	0.556	0.367	0.568	0.545	0.383	0.543	0.552	0.568	0.545	0.544
11	0.760	0.757	0.760	0.760	0.757	0.757	0.760	0.760	0.767	0.757	0.757
12	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
13	0.840	0.840	0.840	0.840	0.840	0.840	0.840	0.840	0.840	0.84	0.840
14	0.569	0.569	0.569	0.587	0.569	0.587	0.587	0.569	0.587	0.587	0.569
15	0.852	0.852	0.852	0.852	0.639	0.852	0.852	0.852	0.852	0.852	0.852
16	1	1	1	1	1	1	1	1	1	1	1
17	0.851	0.851	0.851	0.851	0.851	0.851	0.851	0.581	0.851	0.851	0.851
18	0.735	0.735	0.735	0.735	0.735	0.735	0.735	0.735	0.735	0.735	0.735
19	0.443	0.532	0.149	0.465	0.503	0.523	0.522	0.136	0.513	0.507	0.517
20	0.454	0.472	0.194	0.466	0.458	0.477	0.309	0.136	0.295	0.453	0.199
21	0.410	0.429	0.330	0.429	0.41	0.431	0.203	0.41	0.203	0.429	0.232
22	0.283	0.246	0.430	0.246	0.246	0.246	0.479	0.246	0.385	0.337	0.246
23	0.520	0.525	0.6	0.528	0.558	0.530	0.6	0.543	0.516	0.585	0.552
24	0.709	0.677	0.731	0.682	0.682	0.650	0.731	0.731	0.720	0.682	0.682
25	0.671	0.7	0.718	0.671	0.699	0.710	0.710	0.706	0.666	0.696	0.671
26	0.558	0.570	0.571	0.567	0.558	0.482	0.468	0.573	0.521	0.561	0.565
27	0.176	0.298	0.484	0.244	0.161	0.224	0.165	0.479	0.147	0.479	0.389

Table 16: Cumulative results on test problems

7.3 Discussion

Results from table 16 indicate that the GP-SLCA framework was able to evolve coefficients that generalised over the entire set of problems. As a measure of the quality of the generalisation, the mean value of grouping efficacy was calculated and compared with the same value achieved by Jaccard's coefficient (table 17).

	SET1	SET2	SET3	SET4	SET5	SET6	SET7	SET8	SET9	SET10	Jaccard
$\bar{\Gamma}$	0.629	0.629	0.610	0.630	0.621	0.622	0.622	0.584	0.621	0.646	0.620

Table 17: Mean value of grouping efficacy

The mean value of grouping efficacy produced by all evolved coefficients is similar to the one produced by Jaccard's coefficient. Coefficients SET4 and SET10 performed particularly well on the entire set of problems, producing an increase of 1% and 2.6% respectively on average grouping efficiency in comparison to Jaccard's coefficient. Since the difference in performance is relatively small, further research is needed in order to see if SET4 and SET10 can be distinguished from Jaccard's coefficient. A winner-takes-all comparison of their relative performance on the test problems is presented in table 18.

	Jaccard's coefficient better	**Jaccard's coefficient worse**	**Jaccard's coefficient equal**
SET4	5	7	15
SET10	5	10	12

Table 18. Jaccard's coefficient vs. SET4 and SET10 in terms of non-dominated solutions

It is obvious that there is a large number of problems where the same level of grouping efficacy has been achieved by all coefficients, thus we cannot safely reject the null hypothesis. The Analysis Of Variance (ANOVA) between the three sets of values confirms this statement (table 19).

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
SET4	27	17.022	0.630444	0.035404
SET10	27	17.438	0.645852	0.027259
JACCARD	27	16.737	0.619889	0.041745

ANOVA

F	*P-value*	*F crit*
0.132259	0.87631	3.113797

Table 19: ANOVA for SET4, SET10 and Jaccard's coefficient (α=0.05)

From the *F*-value it is obvious that we cannot reject the null hypothesis. If coefficient SET4 is taken out of consideration, the resulting ANOVA is presented in table 20:

SUMMARY

Groups	Count	Sum	Average	Variance
SET10	27	17.438	0.645852	0.027259
JACCARD	27	16.737	0.619889	0.041745

ANOVA

F	P-value	F crit
0.263753	0.609729	4.026631

Table 20: ANOVA for SET10 and Jaccard's coefficient (α=0.05)

While the value of *F* has increased, it is still much smaller than the critical value, thus the null hypothesis cannot be rejected.

In any case from the above results we can safely conclude that the GP-SLCA algorithm was able to evolve similarity coefficients that perform at least as good as the similarity coefficient that has been devised by human intuition. It will be interesting to take a closer look at the structure of the evolved similarity coefficients. Coefficient SET4 is calculated using the following formula:

$$d\left(d+c+\frac{b}{c}+a^2\right)-\left(\frac{d}{\frac{b}{c}-a}\right)$$

Notice that GP evolves structures that do not follow the elegant form of Jaccard's coefficient, but are just as effective. From the above formula it is clear that the value of the coefficient is proportional to the values of *a* and *d*. This is expected since these values are indicative of the similarity of parts processed between a pair of machines. The structure of coefficient SET10 is much more complicated, as the following formula depicts:

$$4d+3b-b^2-ab+ad+\left[\left(\frac{FACTOR-d-b-(d+b)cb+bc+ab}{d(a+b)}\right)\right]-\left(\frac{c-c}{d+b}\right)-\left(\frac{ad+d+b}{b^2c^2-a-b^2}\right)$$

where *FACTOR* =

$$\frac{a}{-a+(bc-db-cd-2a+b)\left[d-\left(\frac{(2a-3b)(a-b)+ab+db-dc-a+c+\left(\frac{ad}{a}\right)}{a}\right)+\left(\frac{ad}{\left(\frac{a-b}{d-a}\right)\Big/ab^2c}\right)\right]}$$

It is not easy to explain why this particular coefficient seems to perform slightly better than Jaccard's coefficient, because of its size and complexity. This is a common situation in GP, since the application of genetic operators leads to quick growth of programs up to the prespecified maximum depth constraint. It is still evident that the value of the coefficient is proportional to the values of *a* and *d*, however, a number of control terms are also present which seem to fine tune its value in particular fitness

cases. Note that there are two terms that according to common algebra should have been simplified:

$$\frac{c-c}{d+b} \quad \text{and} \quad \left(\frac{ad}{a}\right)$$

However, due to the operation of the protected division function, these expressions will evaluate to '1' if the denominator is equal to '0', which is not an unlikely case. Thus, they should be considered in this form during the calculation of the coefficient value.

The generalisation of this coefficient is quite good. While other coefficients fail to generalise in specific test problems, SET10 seem to have captured information that is relevant to the solution of the problem. In problems 19-21, where the m/c matrices have been customly designed to be difficult for grouping, Jaccard's coefficient fails to find fit partitions. On the same problems SET10 creates cell configurations with much higher levels of grouping efficacy. On the notorious problem 27, where evolved coefficients either completely fail to generalise, or their good performance is not mirrored in the set of validation problems, SET10 produces an excellent level of grouping efficacy. While the difference in performance between SET10 and Jaccard's coefficient could not be mathematically confirmed, results on specific test problems indicate that SET10 is able to handle ill-structured matrices in a more efficient way.

8 Conclusions

In this report we investigated the use of Genetic Programming for the solution of binary cell-formation problems. McAuley' Single Linkage Cluster Analysis (SLCA) algorithm was used as basis for the development of our methodology. SLCA employs Jaccard's similarity coefficient for the creation of a pictorial representation of solutions in the form of a 'dendrogram'. A variety of cell configurations can be created by choosing a particular similarity level in the dendrogram.

Genetic Programming was utilised in two different ways. First, a similarity coefficient was evolved for independent cell formation problems. The coefficient was fed into an SLCA procedure which returned the best solution found in the dendrogram for the desired objective. The methodology was tested on a number of published test problems, performing at least as good as other leading cell-formation algorithms.

GP was also employed for the evolution of general-purpose similarity coefficients. In this case, a number of indifferent test problems were used as fitness cases during the evolutionary procedure. GP-SLCA was able to evolve coefficients which performed at least as good as Jaccard's coefficient on the experimental set-up. One coefficient in particular outperformed Jaccard's coefficient on a number of cases and especially on ill-structured problems.

The GP-SLCA methodology is quite flexible since it can be used with a variety of grouping objectives without altering its main operation. On the other hand, as the size of the problem increases the evaluation function becomes computationally expensive since the value of the coefficient is calculated for each pair of machines in the plant and every possible solution from the dendrogram is evaluated.

GP-SLCA was developed for the solution of binary cell-formation problems, thus it can be criticised as all other similar methods that it does not consider manufacturing data that are important for the design of production cells, like product demands, processing

times, operation sequences etc. However GP-SLCA can be modified to include some of these data as terminals for the evolution of a similarity coefficient. It is in the intention of the authors to continue their research towards this direction.

REFERENCES

Adil, G.K., Rajamani, D., and Strong, D. (1997), "Assignment allocation and simulated annealing algorithms for cell formation", *IIE Transactions*, vol.29, pp.53-67.

Akturk, M.S. and Balkose, H.O. (1996), "Part-machine grouping using a multi-objective cluster analysis", *Int.J. of Production Research*, vol.34, no.8, pp.2299-2315.

Aljaber, N., Baek, W., and Chen, C.-L., (1997), "A tabu search approach to the cell formation problem", *Computers & Industrial Engineering*, vol.32, no.1, pp.169-185.

Al-Qattan, I. (1990), "Designing flexible manufacturing cells using a branch and bound method", *Int.J. of Production Research*, vol.28, no.2, pp.325-336.

Amirahmadi, F. and Choobineh, F. (1996), "Identifying the composition of a cellular manufacturing system", *Int.J. of Production Research*, vol.34, no.9, pp.2471-2488.

Askin, R.G., and Chiu, K.S. (1990), "A graph partitioning procedure for machine assignment and cell formation in group technology", *Int.J. of Production Research*, vol.28, no.8, pp.1555-1572.

Askin, R.G., and Subramanian, S.P. (1987), "A cost-based heuristic for group technology configuration", *Int.J. of Production Research*, vol.25, no.1, pp.101-113.

Askin, R.G., Cresswell, S.H., Goldberg, J.B., and Vakharia, A.J., (1991), "A hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing", *Int.J. of Production Research*, vol.29, no.6, pp.1081-1100.

Askin, R.G., Selim, H.M., and Vakharia, A.J. (1997), "A methodology for designing flexible cellular manufacturing systems", IIE Transactions, vol.29, pp.599-610.

Atmani, A., Lashkari, R.S., and Caron, R.J. (1995), "A mathematical programming approach to joint cell formation and operation allocation in cellular manufacturing", *Int.J. of Production Research*, vol.33, no.1, pp.1-15.

Balakrishnan, J. (1996), "Manufacturing cell formation using similarity coefficients and pair-wise interchange: formulation and comparison" *Production Planning & Control*, vol.7, no.1, pp.11-21.

Banzhaf W., Nordin P., Keller R.E. and Francone F.D. (1998), *Genetic Programming: An Introduction*, Morgan Kaufman, San Francisco, CA.

Beaulieu, A., Ghabi, A. and Ait-Kadi (1997), "An algorithm for the cell formation and the machine selection problem in the design of a cellular manufacturing system", *Int.J. of Production Research*, vol.35, no.7, pp.1857-1874.

Bertsekas, D., and Tseng, P. (1988), "Relaxation methods for minimum cost ordinary and generalised network flow problems", *Operations Research*, vol.36, pp.93-114.

Billo, R.E., Bidanda, B., and Tate, D. (1996), "A genetic cluster algorithm for the machine-component grouping problem", *Journal of Intelligent Manufacturing*, vol.7, no.3, pp.229-243.

Boctor, F.F. (1991), "A linear formulation of the machine-part cell formation problem", *Int.J. of Production Research*, vol.29, no.2, pp.343-356.

Boe, W.J., and Cheng, C.H. (1991), "A close neighbour algorithm for designing cellular manufacturing systems", *Int.J. of Production Research*, vol.29, no.10, pp.2097-2116.

Burbidge, J.L. (1963), "Production Flow Analysis", *Production Engineer*, vol.42, no.12, pp.742-752.

Burbidge, J.L. (1971), "Production Flow Analysis", *Production Engineer*, vol.50, pp. 139-152.

Burbidge, J.L. (1975), *The Introduction of Group Technology*, Halste Press, John Wiley, New York, U.S.A.

Burbidge, J.L. (1977), "A manual method for Production Flow Analysis", *Production Engineer*, vol.56, pp.34-38.

Carrie, A.S., (1973), "Numerical taxonomy applied to group technology and plant layout", *Int.J. of Production Research*, vol.11, no.4, pp.399-416.

Chan, H.M., and Milner, D.A. (1982), "Direct clustering algorithm for group formation in cellular manufacture", *Journal of Manufacturing Systems*, vol.1, no.1, pp.65-75.

Chandrasekharan, M.P., and Rajagopalan, R. (1986), "MODROC: An extension of rank order clustering for group technology", *Int.J. of Production Research*, vol.24, no.5, pp.1221-1233.

Chandrasekharan, M.P., and Rajagopalan, R. (1986b), "An ideal-seed non-hierarchical clustering algorithm for celllar manufacturing", *Int.J. of Production Research*, vol.24, no.2, pp.451-464.

Chandrasekharan, M.P., and Rajagopalan, R. (1987), "ZODIAC - an algorithm for concurrent formation of part families and machine-cells", *Int.J. of Production Research*, vol.25, no.6, pp.835-850.

Chandrasekharan, M.P., and Rajagopalan, R. (1989), "GROUPABILITY: an analysis of the properties of binary data matrices for group technology", *Int.J. of Production Research*, vol.27, no.6, pp.1035-1052.

Cheng, C.H., Gupta, Y.P., Lee, W.H., and Wong, K.F. (1998), "A TSP-based heuristic for forming machine groups and part families", *Int.J. of Production Research*, vol.36, no.5, pp.1325-1337.

Cheng, C.-H., Madan, M.S., and Motwani, J. (1996), "Designing cellular manufacturing systems by a truncated tree search", *Int.J. of Production Research*, vol.34, no.2, pp.349-361.

Choobineh, F.C. (1988), "A framework for the design of cellular manufacturing systems", *Int.J. of Production Research*, vol.26, no.7, pp.1161-1172.

Chu, C.-H. (1989), "Cluster analysis in manufacturing cellular formation", *OMEGA, Int.J. of Management Science*, vol.17, no.3, pp.289-295.

Chu, C.-H., and Hayya, J.C., (1991), "A fuzzy-clustering approach to manufacturing cell formation", *Int.J.of Production Research*, vol.29, no.7, pp.1475-1487.

Co, H.C., and Araar, A. (1988), "Configuring cellular manufacturing systems", *Int.J. of Production Research*, vol.26, no.9, pp.1511-1522.

Da Silveira, G. (1999), "A methodology of implementation of cellular manufacturing", *Int.J. of Production Research*, vol.37, no.2, pp.467-479.

Dahel, N.-E., and Smith, J.B. (1993), "Designing flexibility into cellular manufacturing systems", *Int.J. of Production Research*, vol.31, no.4, pp.933-945.

De Witte, J. (1980), "The use of similarity coefficient in production flow analysis", *Int.J. of Production Research*, vol.18, no.4, pp.503-514.

Deutsch, M.B., Freeman, S.F., and Helander, M. (1998), "Manufacturing cell formation using an improved p-median model", *Computers & Industrial Engineering*, vol.34, no.1, pp.135-146.

Diday, E., Lemaire, J., Pouget, J., and Testud, F. (1982), *Eléments d'Analyse des Données*, Dunod, 1982.

Dimopoulos, C., and Zalzala AMS. (1999a), "Recent developments in evolutionary computation for manufacturing optimisation: problems, solutions and comparisons", *IEEE Transactions in Evolutionary Computation*, in print.

Dimopoulos, C., and Zalzala AMS. (1999b), "A genetic programming heuristic for the one-machine total tardiness problem", in *Congress on Evolutionary Computation, (CEC'99).*

Dimopoulos, C., and Zalzala AMS. (1999c), "Evolving scheduling policies through a genetic programming framework", in *Genetic and Evolutionary Computation Conference (GECCO'99)*, Banzhaf *et al.* (eds), vol.2, pp.1231, Morgan Kaufman, San Francisco, CA, U.S.A.

El-Essawy, L.G.K., and Torrance, J. (1972), "Component Flow Analysis-an effective approach to production systems design", *Production Engineer*, vol.51, pp.165-170.

Garcia, H., and Proth, J.M. (1986), "A new cross-decomposition algorithm: the GPM. Comparison with thebond-energy method", *Control & Cybernetics*, vol.15, no.2, pp.155-165.

Gomory, R.E., and Hu, T.C. (1971), "Multi-terminal network flows", *SIAM Journal of Applied Mathematics*, vol.9, pp.551-571.

Gravel, M., Nsakanda, A.L., and Price, W. (1998), "Efficient solutions to the cell formation problem with multiple routings via a double-loop genetic algorithm", *European Journal of Operational Research*, vol.109, pp.286-298.

Gu, P. (1991), "Process-based machine grouping for cellular manufacturing systems", Computers in Industry, vol.17, pp.9-17.

Gunasingh, R.K., and Lashkari, R.S. (1991), "Simultaneous grouping of parts and machines in cellular manufacturing systems", *Computers & Industrial Engineering*, vol.20, no.1, pp.111-117.

Gupta, T. (1993), "Design of manufacturing cells for flexible environment considering alternative routeing", *Int.J. of Production Research*, vol.31, no.6, pp.1259-1273.

Gupta, T., and Seifoddini, H. (1990), "Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system", *Int.J. of Production Research*, vol.28, no.7, pp.1247-1269.

Gupta, Y., Gupta, M., Kumar, A., and Sundaram, C., (1996), "A genetic algorithm-based approach to cell-composition and layout design problems", *Int.J.of Production Research*, vol.34, no.2 pp.447-482.

Harhalakis, G.R., Nagi, R., and Proth, J.-M. (1990), "An efficient heuristic in manufacturing cell formation for group technology applications", *Int.J. of Production Research*, vol.28, no.1, pp.185-198.

Held, M., and Karp, R. (1962), "A dynamic programming approach for sequencing problems", *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, vol.10, no.1, pp.196-210.

Heragu, S.S., and Gupta, Y.P. (1994), "A heuristic for designingcellular manufacturing facilities", *Int.J. of Production Research*, vol.32, no.1, pp.125-140.

Ho, Y.-C., and Moodie, C.L., (1996), "Solving cell formation problems in a manufacturing environment with flexible processing and routeing capabilities", *Int.J. of Production Research*, vol.34, no.10, pp.2901-2923.

Hsu, C.-M., and Su, C.-T. (1998) " Multi-objective machine-component grouping in cellular manufacturing: a genetic algorithm", *Production Planning & Control*, vol.9, no.2, pp.155-166.

Hwang, H., and Ree, P. (1996), "Routes selection for the cell formation problem with alternative part process plans", *Computers & Industrial Engineering*, vol.30, no.3, pp.423-431.

Hwang, H., and Sun, J.-U. (1996), "A genetic algorithm-based heuristic for the GT cell formation problem", *Computers & Industrial Engineerin*g, vol.30, no.4, pp.941-955.

Hyer, N.L., AND Wemmerlov, U. (1985), "Group Technology oriented coding systems: structures, application and implementation", *Production and Inventory Management*, vol.26, no.2, pp.125-147.

Iri, M. (1968), "On the sunthesi of loop and cutset matrices and related problems", *RAAG Memoirs*, vol.4, A-XII, pp.376-410.

Jaccard, P. (1908), "Nouvelles recherches sur la distribution florale", *Bull.Soc.Vaud.Sci.Nat.*, vol.44, pp.223-270.

Jain, A.K., Kasilingam, R.J., and Bhole, S.D. (1991), "Joint consideration of cell formation and tool provisioning problems in flexible mabufacturing systems", *Computers & Industrial Engineering*, vol.20, no.2, pp.271-277.

Joines, J.A., Culbreth, C.T., and King, R.E. (1996), "Manufacturing cell design: an integer programming model employing genetic algorithms", *IIE Transactions*, vol.28,no.1, pp.69-85.

Kandiller, L. (1998), "A cell formation algorithm: Hypergraph approximation – Cut tree", *European Journal of Operational research*, vol.109, pp.686-702.

Kao, Y., and Moon, Y.B. (1998), "Feature-based memory association for group technology", *Int.J. of Production Research*, vol.36, no.6, pp.1653-1677.

Kernighan, B.W., and Lin, S. (1970), "An efficient heuristic procedure for partitioning graphs", *Bell Systems Technology Journal*, vol.49, pp.291-307.

Khator, S.K., and Irani, S.A. (1987), "Cell formation in group technology: a new approach", *Computers & Industrial Engineering*, vol.12, no.2, pp.131-142.

King, J.R. & Nakornchai, V. (1982), "Machine-component group formation in group technology: review and extension", *Int.J. of Production Research*, vol.20, no.2, pp.117-133.

King, J.R. (1980), "Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm", *Int.J. of Production Research*, vol.18, no.2, pp.213-232.

Koza, J.R. (1992), *Genetic Programming: On the programming of computers by means of natural selection*, MIT Press, Cambridge.

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", *Int.J. of Production Research*, vol.28, no.2, pp.603-612.

Kumar, K.R., and Vannelli, A. (1987), "Strategic subcontrcting for efficient disaggregated manufacturing", *Int.J. of Production Research*, vol.25, no.12, pp.1715-1728.

Kumar, K.R., Kusiak, A., and Vannelli, A. (1986), "Grouping of parts and components in flexible manufacturing systems", *European Journal of Operational research*, vol.24, pp.387-397.

Kusiak, A. (1987), "The generalised group technology concept", *Int.J. of Production Research*, vol.25, no.4, pp.561-569

Kusiak, A. (1991), "Branching algorithms for solving the group technology problem", *Journal of Manufacturing Systems*, vol.10, no.4, pp.332-343.

Kusiak, A., and Cho, M. (1992), "Similarity coefficient algorithms for solving the group technology problem", *Int.J. of Production Research*, vol.30, no.11, pp.2633-2646.

Kusiak, A., and Chow, W.S. (1987), "Efficient solving of the group technology problem", *Journal of Manufacturing Systems*, vol.6, no.2, pp.117-124.

Kusiak, A., and Heragu, S.S. (1987), "Group Technology", *Computers in Industry*, vol.9, pp.83-91.

Lee, H., and Garcia-Diaz, A. (1993), "A network flow approach to solve clustering problems" *Int.J. of Production Research*, vol.31, no.3, pp.603-612.

Lee, J.G., Vogt, W.G., and Mickle, M.H. (1979), "Optimal decomposition of large-scale networks", *IEEE Transactions on Supply and Maintenance Command*, SMC-9, pp.369.

Lee, M.K., Luong, H.S., and Abhary, K. (1997), "A genetic algorithm based cell design considering alternative routing", *Computer Integrated Manufacturing Systems*, vol.10, no.2, pp.93-107.

Lee, S.-D., and Chen, Y.-L. (1997), "A weighted approach for cellular manufacturing design: minimising intercell movement and balancing workload among duplicated machines", *Int.J. of Production Research*, vol.35, no.4, pp.1125-1146.

Lin, L.T., Dessouky, M.M., Kumar, K.R. and Ng, M.S. (1996), "A heuristic-based procedure for the weighted production – cell formation problem", *IIE Transactions*, vol.28, pp.579-589.

Logendran, R. (1992), "A model for duplicating bottleneck machines in the presence of budgetary limitations in cellular manufacturing", *Int.J. of Production Research*, vol.30, no.3, pp.683-694.

Luong, L.H.S. (1993), "A cellular similarity coefficient algorithm for the design of anufacturing cells", *Int.J. of Production Research*, vol.31, no.8, pp.1757-1766.

MacQueen, J.B. (1967), "Some methods for classification and analysis of multivariate observations", in *Proc.of the 5th Symposium on Mathematical Statistics and Probability*", vol.1, pp.281, Univ.of California, Berkeley.

McAuley, J. (1972), "Machine grouping for efficient production", *Production Engineer*, vol.51, no.2, pp.53-57.

McCormick, W.T., Schweitzer, R.J., and White, T.W. (1972), "Problem decomposition and data reorganisation by clustering techniques", *Operations Research*, vol.20, no.5, pp.993-1009.

McKay, B.M., Willis, M.J., Hiden, H.G., Montague, G.A., and Barton, G.W. (1996), "Identification of industrial processes using genetic programming", in *Proc.of the Conf .on Identification in Engineering Systems*, Friswell & Mottershead (Eds.), pp.510-519, Univ.of Wales, Swansea, UK. 1996.

Mitrovanov, S.P. (1966), *The Scientific Principles of Group Technology*, National Lending Library Translation, Boston Spa, Yorkshire, U.K.

Mosier, C. and Taube, L. (1985), "Weighted similarity measure heuristics for the group technology machine clustering problem", OMEGA: *Int.J. of Management Science*, vol.13, n0.6, pp.577-583.

Mukhopadhyay, S.K., and Bhandari, C. (1997), "Conjoint measurement approach for machine-part grouping in manufacturing cell formation", *Int.J. of Production Research*, vol.35, no.8, pp.2237-2251.

Mukhopadhyay, S.K., and Gopalakrishnan, A. (1995a), "A vector analysis (VECAN) method for solving the machine-part grouping problem in GT", *Int.J. of Production Research*, vol.33, no.3, pp.795-818.

Mukhopadhyay, S.K., Gopalakrishnan, A., and Kripalani, M.R. (1995b), "Moments-based clustering techniques for manufacturing cell formation", *Int.J. of Production Research*, vol.33, no.4, pp.1091-1115.

Mukhopadhyay, S.K., Sarkar, P., and Panda, R.P. (1995a), "Machine-component grouping in cellular manufacturing by multidimensional scaling", *Int.J. of Production Research*, vol.32, no.2, pp.457-477.

Nagi, R., Harhalakis, G., and Proth, J.-M. (1990), "Multiple routings and capacity considerations in group technology applications", *Int.J. of Production Research*, vol.28, no.12, pp.2243-2257.

Nair, J.G., nad Narendran, T.T. (1998), "CASE: a clustering algorithm for cell formation with sequence data", *Int.J. of Production Research*, vol.36, no.1, pp.157-179.

Nair, J.G., nad Narendran, T.T. (1999), "ACCORD: a bicriteria algorithm for cell formation using ordinal and ratio level data", *Int.J. of Production Research*, vol.37, no.3, pp.539-556.

Ng, S.M. (1993), "Worst-case analysis of an algorithm for cellular manufacturing", *European Journal of Operational Research*, vol.69, pp.384-398.

Offodile, O.F., Mehrez, A., and Grznar, J. (1994), "Cellular manufacturing: a taxonomic review framework", *Journal of Manufacturing Systems*, vol.13, no.3, pp.196-220.

Okogbaa, O.G., Chen, M.T., Chaweng, C., and Shell, R.L. (1992), "Manufacturing system cell formation and evaluation using a new inter-flow reduction heuristic", *Int.J. of Production Research*, vol.30, no.5, pp.1101-1118.

Polheim, H., and Marenback, P. (1996) "Generation of structured process models using genetic programming", in *Evolutionary Computing. AISB Workshop. Selected Papers*, Lecture Notes in Computer Science (1143), T.C.Fogarty (Ed.), pp.102-109, Springer-Verlag, Berlin, Germany.

Rajagopalan, R., and Batra, J.L. (1975), "Design of cellular production systems: a graph-theoretic approach", *Int.J. of Production Research*, vol.13, no.6, pp.567-579.

Rajamani, D., Singh, N., and Aneja, Y.P. (1990), "Integrated design of cellular manufacturing systems in the presence of alternative process plans", *Int.J. of Production Research*, vol.28, no.8, pp.1541-1554.

Rath, S.K., Das, S., and Sahu, S. (1995), "Graph colouring approach for cell- formation problems", *Production Planning & Control*, vol.6, no.5, pp.421-428.

Seifoddini, H (1989), "Singke linkage vs. avaerage linkage clustering in machine cells formation application", *Computers & Industrial Engineering*, vol.16, pp.419-426.

Seifoddini, H. and Wolfe, P.M. (1986), "Selection of a threshold value based on material handling cost in machine-component grouping", *AIIE Transactions*, vol.19, no.3, pp.266-270.

Seifoddini, H., and Djassemi, M. (1995), "Merits of the production volume based similarity coefficient in machine cell formation", *Journal of Manufacturing Systems*, vol.14, no.1, pp.35-44.

Selim, M.H., Askin, R.G., and Vakharia, A.J. (1998), "Cell formation in group technology: review, evaluation and directions for future research", *Computers & Industrial Engineering*, vol.34, no.1, pp.3-20.

Shafer, S.M., and Rogers, D.F. (1991), "A goal programming approach to the cell formation problem", Journal of Operations Management, vol.10, no.1, pp.28-43.

Shafer, S.M., Kern, G.M., and Wei, J.C. (1992), "A mathematical programming approach for dealing with exceptional elements in cellular manufacturing", *Int.J. of Production Research*, vol.30, no.5, pp.1029-1036.

Sharker, B.R. (1986), "The resemblance coeficients in group technology: a survey and comparative study of relational metrics", *Computers & Industrial Engineering*, vol.30, no.1, pp.103-116.

Sharker, B.R., and Mondal, S. (1999), "Grouping efficiency measures in cellular manufacturing: a survey and critical review", *Int.J. of Production Research*, vol.37, no.2, pp.385-314.

Shtub, A. (1989), "Modelling group technology cell formation as a generalised assignment problem", *Int.J. of Production Research*, vol.27, no.5, pp.775-782.

Singh, N. (1993), "Cellular manufacturing systems: an invited review", *European Journal of Operational Research*, vol.69, pp.284-291.

Sneath, P.H.A, (1957), "The application of computers to taxonomy", *J.of Gen.Microb.*, vol.17, pp.201-206.

Sofianopoulou, S. (1997), "Application of simulated annealing to a linear model for the formulation of machine cells in group technology", *Int.J. of Production Research*, vol.35, no.2, pp.501-511.

Sofianopoulou, S. (1999), "Manufacturing cells design with alternative process plans and/or replicate machines", *Int.J. of Production Research*, vol.37, no.3, pp.707-720.

Srinivasan, G, and Narendran, T.T. (1991), "GRAFICS – a nonhierarchical clustering algorithm for group technology", *Int.J. of Production Research*, vol.29, no.3, pp.463-478.

Srinivasan, G, and Narendran, T.T., and Mahaderan, B. (1990), "An assignment model for the part-families problem in group technology", *Int.J. of Production Research*, vol.28, no.1, pp.145-152.

Srinivasan, G. (1994), "A clustering algorithm for machine cell formation in group technology using minimum spanning trees", *Int.J. of Production Research*, vol.32, no.9, pp.2149-2158.

Stanfel, L.E. (1985), "Machine clustering for economic production", *Engineering Costs & Production Economics*, vol.9, pp.73-81.

Steudel, H.J., and Ballakur, A., (1987), "A dynamic programming – based heuristic for machine grouping in manufacturing cell formation", *Computers & Industrial Engineering*, vol.12, no.3, pp.215-222.

Sun, D., Lin, L., and Batta, R. (1995), "Cell formation using tabu search", *Computers & Industrial Enegineering*, vol.28, no.3, pp.485-494.

Suresh, N.C., Slomp, J., and Kaparthi, S. (1995), "The capacitated cell formation problem: a new hierarchicla methodology", *Int.J. of Production Research*, vol.33, no.6, pp.1761-1784.

Taboun, S.M., Merchawi, N.S., and Ulger, T. (1998), "Part family and machine cell formation in multi-period planning horizons of cellular manufacturing systems", *Production Planning & Control*, vol.9, no.6, pp.561-571.

Tam, K.Y. (1990), "An operation sequence based similarity coefficient for part families formations", *Journal of Manufacturing Systems*, vol.9, no.1, pp.55-68.

Teitz, M.B., and Bart, P. (1968), "Heuristic methods for estimating the generalised vertex median of a weighted graph", *Operations Research*, vol.16, pp.955-961.

Vakharia, A.J., and Chang, Y.-L. (1997), "Cell formation in group technology: a combinatorial search approach", *Int.J. of Production Research*, vol.35, no.7, pp.2025-2043.

Vakharia, A.J., and Wemmerlov, U. (1987), "A new design method for cellular manufacturing systems", in *Proceedings of the IXth ICPR*, pp.2357-2363.

Vakharia, A.J., and Wemmerlov, U. (1990), "Designing a cellular manufacturing system: a material flow approach based on operation sequences", *IIE Transactions*, vol.22, no.1, pp.84-97.

Vannelli, A., and Kumar, K.R. (1986), "A method for finding minimal bottleneck cells for grouping part-machine families", *Int.J. of Production Research*, vol.24, no.2, pp.387-400.

Venugopal, V., and Narendran, T.T. (1992), "A genetic algorithm approach to the machine-component grouping problem with multiple objectives", *Computers & Industrial Engineering*, vol.22, no.4, pp.269-480.

Vohra, T., Chen, D.-S., Chang, J.C., and Chen, H.-C. (1990), "A network approach to cell formation in cellular manufacturing", *Int.J. of Production Research*, vol.28, no.11, pp.2075-2084.

Waghodekar, P.H., and Sahu, S. (1984), "Machine-component cell formation in group technology: MACE", *Int.J. of Production Research*, vol.22, no.6, pp.937-948.

Wei, J.C., and Gaither, N. (1990), "An optimal model for cell formation decisions", Decision Sciences, vol.21, no.2, pp.416-433.

Wei, J.C., and Kern, G.M. (1989), "Commonality analysis: alinear cell clustering algorithm for group technology", *Int.J. of Production Research*, vol.27, no.12, pp.2053-2062.

Wu, N. (1998), "A concurrent approach to cell formation and assignment of identical machines in group technology", *Int.J. of Production Research*, vol.36, no.8, pp.2099-2114.

Wu, N., and Salvendy, G. (1993), "A modified network approach for the design of cellular manufacturing systems", *Int.J. of Production Research*, vol.31, no.6, pp.1409-1421.

Zhu, Z., Heady, R.B., nad Reiners, S. (1995), "An efficient zero-one formulation of the cell formation problem", *Computers & Industrial Engineering*, vol.28, no.4, pp.911-916.