A NOVEL APPROACH FOR THE SOLUTION OF THE MULTIOBJECTIVE CELL-FORMATION PROBLEM

C. Dimopoulos
Department of Computer Science, Cyprus College, 6 Diogenes Str., Nicosia, Cyprus
dimopoulos@cycollege.ac.cy

Abstract
This paper presents a hybrid heuristic methodology for the solution of the multi-objective cell-formation problem. Traditional optimization methodologies employ aggregating schemes in order to transform the problem into a single-objective case. In this way the designer is not presented with a set of non-dominated solutions but with a single compromise solution based on pre-specified weighting priorities. The proposed methodology combines a traditional hierarchical clustering analysis technique with a genetic programming algorithm that is based on the principles of evolutionary computation. The hybrid methodology evolves an approximation of the Pareto set of solutions for multi-objective cell-formation problems. The benefits brought by the proposed approach in comparison to traditional optimization methodologies are illustrated using a typical example taken from the literature.

Keywords:
Cellular manufacturing, evolutionary algorithms, multiobjective optimization.

1 INTRODUCTION

While most practical manufacturing optimization problems require the simultaneous optimization of multiple (usually conflicting) objectives, research in this area has been relatively limited. A typical example is the problem of designing cells in a cellular manufacturing production system. A considerable number of methodologies have been proposed for the solution of the single-objective version of the problem over the last four decades. The multiobjective version of the cell-formation problem has received limited attention. However, a number of researchers indicate ([12], [14]) that multiobjective considerations are very frequent during the design of a manufacturing system.

Evolutionary Algorithms (EAs) are particularly suited for the solution of multiobjective optimization problems since their search mechanism is based on the use of a population of candidate solutions. This feature has been exploited by a number of researchers that have proposed evolutionary systems which search not for a single solution but for the Pareto set of solutions (Multiobjective Evolutionary Algorithms, MOEAs). These MOEAs have been reported to provide efficient solutions to non-trivial multiobjective optimization problems (see [13] for an excellent review of MOEA research).

The methodology presented in this paper combines an EA-based methodology for the solution of single-objective cell-formation problems (GP-SLCA [16]) with NSGA-II [17], a state-of-the-art evolutionary technique for multiobjective optimization. A typical example taken from the literature is used to illustrate the benefits gained from the proposed methodology.

The rest of this paper is organized as follows: A brief review of the multiobjective cell formation problem is provided in section 2. Section 3 provides a description of the proposed solution methodology. Section 4 illustrates its application to an example problem taken from the literature and reviews the experimental findings. The conclusions of this research and possible future developments are discussed in section 5.

2 THE MULTIOBJECTIVE CELL-FORMATION PROBLEM

Cellular Manufacturing (CM) is the application of the organizational approach called Group Technology (GT) [1] at the shop floor production level. It states that there are considerable benefits to be gained by grouping machines into cells that process similar parts. CM has been shown to provide considerable cost benefits to practical manufacturing environments [14]. Despite the advent of new production design techniques such as Just In Time (JIT) systems and Agile manufacturing, CM is still considered to be a useful design principle since its application requires limited capital investment.

The general multiobjective cell-formation problem can be stated as follows: A grouping of machines into cells and parts into associated families needs to be identified that will simultaneously optimize a number of objectives. A single solution that simultaneously optimizes all objectives considered does not generally exist for this problem. Instead, there exists a set of solutions, known as the Pareto set of solutions, which dominate every other solution in the solutions' space with respects to all objectives considered. The solutions belonging to the Pareto set do not dominate each other since no solution is better than the other with respect to all objectives considered. The aim of multiobjective optimization methodologies is to provide the decision maker with the Pareto set of solutions or at least a close approximation of this set. It is the task of the decision maker to choose a solution that fits best his/her preferences by considering all potential trade-offs.

While a considerable number of analytic and heuristic methodologies have been proposed for the solution of the single objective cell-formation problem, a limited number of methodologies have addressed the multiobjective version of the problem. As illustrated in the recent reviews contacted by Mansouri [12] and Dimopoulos [19], the majority of these techniques perform multiobjective optimization by aggregating all objectives considered into a single compromise solution [5], [8], [9], [10]. This mechanism allows the use of typical single-objective optimization techniques for the solution of multiobjective optimization problems; however, it does not provide the set of non-dominated solutions to the decision maker.

A small number of researchers in the field of evolutionary computation have proposed solution methodologies that attempt to generate the Pareto set of solutions for specific versions of the multiobjective cell-formation problem. Venugopal and Narendran [6] designed an evolutionary algorithm that evolved two populations of potential solutions in parallel, each one responsible for the optimization of a particular objective. There was no attempt to explicitly generate the Pareto set of solutions; however the designer could consider a number of alternative trade-off solutions by observing the set of solutions already evolved in each population.

Gupta et al. [7] addressed a multiobjective cell-formation problem using a similar evolutionary algorithm, however, the mutliobjective solutions were identified indirectly from the population of solutions that was evolved for the optimization of individual objectives.

Recently, Solinmanpur et al. [20] proposed the first genuine MOEA for the solution of a multiobjective cell-formation problem. They employed an improved version of a primitive MOEA technique called VEGA [3] to drive the simultaneous optimization of all objectives considered. While their approach was able to automatically generate a set of alternative solutions, the use of the VEGA technique is rather outdated, since a number of state-of-the-art EMOA techniques exist that have been reported to provide improved performance [13].

The algorithm presented in this article, follows on the guidelines discussed in [12] and [19] and introduces a robust multiobjective optimization methodology for the cell-formation problem that attempts to provide the decision maker with a good approximation of the Pareto set of solutions. This methodology is described in the following section.

3 THE MULTIOBJECTIVE GP-SLCA METHODOLOGY

3.1 Introduction

The methodology presented in this article is based on the combination of GP-SLCA [16], an established methodology for solving single-objective cell-formation problems, with NSGA-II [17], a state of-the art EMOA technique. While it is not possible to describe in detail the operation of the individual techniques , an outline of the algorithms, as well as a basic introduction on evolutionary computation concepts will be provided in the following paragraphs. A number of references are provided where interested readers can find additional information as well as experimental results.

3.2 Evolutionary Algorithms and Genetic Programming

Evolutionary Algorithms are heuristic optimization techniques that base their operation on a rough analogy to the process of natural selection of species and the principle of the survival of the fittest. Their search is contacted from a population of suitably encoded solution points in parallel. Each solution in the population receives a fitness value based on its performance on the problem considered. This value stochastically determines the probability of a solution 'surviving' to a subsequent 'generation' of solutions. In this way more efficient ('fitter') solutions are continuously promoted, however, since the process of selection is not deterministic, less efficient solutions can survive to subsequent generations. This mechanism ensures that evolutionary algorithms have the ability of escaping local optima in a particular solutions' space. During the evolutionary process new solutions are created either through the exchange of 'genetic material' between existing solutions ('crossover'), or through the random modification of existing solutions ('mutation'). The evolutionary cycle continues for a pre-specified number of generations. An optimal or a near-optimal solution is normally found by the end of this process, however, convergence cannot be mathematically guaranteed for the majority of optimization problems.

Several variants of Evolutionary Algorithms exist. They all share the basic algorithmic structure described in the previous paragraph, but they use different solution representation techniques or selection strategies. The evolutionary algorithm described in this paper is based on Genetic Programming, a technique that evolves solutions in the form of computer programs. These programs are normally evolved as parse trees, structures that many computers use internally to represent computer programs. In this paper Genetic Programming is employed for the evolution of computer programs that correspond to similarity coefficients, i.e. structures that calculate the level of similarity between a pair of entities. An excellent introduction to the basic principles and types of Evolutionary Algorithms can be found in [18].

3.3 GP-SLCA

The operation of GP-SLCA is based on the Single Linkage Cluster Analysis (SLCA) cell-formation technique, originally introduced by McAuley [2].

SLCA uses Jaccard's similarity coefficients to calculate a measure of similarity (similarity coefficient) between all pairs of machines for a given problem. A dendrogram of potential solutions is then generated based on the values of similarity coefficients. A particular configuration of cells is created by choosing a threshold similarity value on the dendrogram of solutions. A simple step-by-step illustration of the SLCA process is provided in the Appendix of this article.

GP-SLCA employs a similar structure; however, instead of using Jaccard's similarity coefficient, a population of similarity coefficients is evolved through a Genetic Programming machine. The first generation of coefficients is created randomly. The similarity inputs used by GP-SLCA for the construction of similarity coefficients are the following (the same inputs are used by the multiobjective GP-SLCA methodology):

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 *j* nor machine *i*

The formulas of the similarity coefficients are created by combining the previous inputs with a typical set of mathematical operators (addition, subtraction, multiplication, division) .

Initially, the SLCA process is applied to each coefficient evolved. During this process all cell groupings in the respective dendrogram of solutions are recorded and their performance is evaluated based on the optimization objective. The best objective value found in the dendrogram of solutions is assigned as the fitness value of the corresponding coefficient.

The evolutionary process continues with the selection and recombination steps, where coefficients are modified using typical recombination operators (crossover and mutation) until a new population of coefficients is formed.

In algorithmic terms, the operation of GP-SLCA is the following:

Procedure GP-SLCA
initialise population of randomly created similarity coefficients
run procedure SLCA for each coefficient
loop
    loop
        select coefficients for crossover or mutation
        apply genetic operators and form new coefficients
        until a new generation of coefficients 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
until a single cell has been formed
assign the best solution found as fitness of the individual

GP-SLCA creates a considerable number of alternative similarity coefficients and corresponding cell groupings which are continuously optimized through the evolutionary process. The efficiency of GP-SLCA has been illustrated on a wide range of simple cell-formation problems that have been published in the literature [16], as well as on advanced versions of the problem [11]. It has also been shown that it is extremely easy to incorporate any cell-size or cell-number constraints in the algorithm, a feature that is not readily available in many alternative solution methodologies [11].

3.4 NSGA-II

Multiobjective GP-SLCA employs the NSGA-II evolutionary multiobjective technique as the driving force of the evolutionary algorithm [17]. Unlike the traditional evolutionary cycle which attempts to find a single optimal or near-optimal solution, NSGA-II promotes the evolution of a set of solutions that is ideally a close approximation of the Pareto-set of solutions for the problem considered.

In short, this is mainly achieved by using a ranking scheme for all solutions evolved. All non-dominated solutions found in a population of solutions are assigned with rank '1' and removed from consideration. The set of non-dominated solutions found in the remaining population are assigned with rank '2'. The process continues until all solutions have been assigned with a rank. The selection of solutions that will form the next generation of coefficients is based on their current ranking value and the set of non-dominated solutions that have already been found during the evolutionary process, which is kept separately.

The algorithm contains special mechanisms that prevent the premature convergence of the algorithm on a particular non-dominated solution by penalizing solutions that are situated very close to each other in the multiobjective space. Note that NSGA-II is an elitist algorithm, thus the final generation of solutions contains all non-dominated solutions that were evolved during the experimental run. A detailed description of the NSGA-II mechanism together with experimental results that illustrate its efficiency can be found in [17].

3.5 Multiobjective GP-SLCA

The proposed methodology combines the GP-SLCA methodology, a mechanism for generating potential solutions for the cell-formation problem, with the NSGA-II MOEA technique, a mechanism that allows the simultaneous evolution of a set of non-dominated solutions based on the conflicting objectives considered. Due to the consideration of multiple objectives, some modifications are necessary to the algorithmic structure of the single-objective GP-SLCA algorithm.

The major modification is related to the fitness assignment process of GP-SLCA: The evolutionary process requires that a fitness value is associated to each evolved coefficient. However, since this is a multiobjective problem, a single 'best' solution does not generally exist in the set of solutions found in the dendrogram of the SLCA process. Instead, there exist multiple equally 'good' non-dominated solutions with respect to all objectives considered. Since a set of objective values for each coefficient is needed by the NSGA-II process for a ranking of solutions to take place, a random similarity threshold value is associated with each coefficient evolved. This threshold value is used as input for the SLCA algorithm. It specifies the cell configuration that corresponds to the similarity coefficient from the respective dendrogram of solutions. In this way, each evolved coefficient generates only one machine-cell configuration unlike the original GP-SLCA algorithm. The objective values that correspond to this configuration constitute the objective values of the coefficient and are subsequently used by the NSGA-II evolutionary technique in order to rank solutions according to their Pareto efficiency. Note that each time a new coefficient is produced through the processes of crossover or mutation, an associated similarity threshold value is randomly generated by the GP machine.

In algorithmic terms, the operation of the multiobjective GP-SLCA is the following:

Procedure multiobjective GP-SLCA
initialize population of randomly created similarity coefficients
run procedure mod-SLCA for each coefficient
rank solutions using the NSGA-II process based on the objective values
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
    rank solutions using the NSGA-II process based on the objective values
until termination criterion is true

Procedure mod-SLCA
compute similarity matrix
create machine cells for the associated random similarity threshold value
assign parts to machine cells
calculate the objective values for the cell configuration

4 EXPERIMENTAL RESULTS

In the case of the single-objective cell-formation problem a large number of test problems exist for which comparative results are available. However, the same cannot be said for the case of the multiobjective cell-formation problem. The review of Mansouri et al. [12] depicts that the majority of published solution methodologies have been tested on

individual cases for which comparative results do not exist. This is mainly due to the fact that a standardised mathematical model of the multiobjective cell-formation problem does not exist. The various models that have been proposed by researchers differ significantly in terms of the input data used, the criteria that need to be optimised and the constraints that must not be violated [12].

The multiobjective GP-SLCA methodology presented in the previous section was applied on a large-sized test problem taken from the literature (Venugopal and Narendran [6]). This problem involves 15 workstations and 30 parts. 20 runs of the algorithm were contacted on the test problem. It is customary for Genetic Programming algorithms to describe the parameters of the experimental runs through the so-called Koza tableau. The respective table for multiobjective GP-SLCA is illustrated in Table 1.

Table 1: Parameter settings for multiobjective GP-SLCA experimental runs

Parameters	***Values***
Objective:	Simultaneous minimisation of total intercell moves and total cell-load variation
Terminal set:	α_{ij}, b_{ij}, c_{ij}, d_{ij} (defined in section 3.3)
Function set:	+, -, ×, % (protected division function)
Population size:	500
Crossover probability:	.5
Mutation probability:	.5
Selection:	Tournament selection
Number of generations:	50
Initialisation method:	Ramped half and half

Multiobjective GP-SLCA attempted to find groupings of machine cells and associated part families that would simultaneously optimize the following objectives (based on the model proposed by Venugopal and Narendran):

F1: Minimization of total intercell moves

This objective is calculated as the total sum of intercell moves for a given cell configuration. Any move made by a part between workstations that belong to different cells is considered to be an intercell move.

F2: Minimization of within cell-load variation

This objective is calculated as the difference between the workload induced by a part on a specific workstation and the average workload induced by the part on the workstation's cell. The minimization of this objective ensures the smooth processing flow of materials within cells.

The objectives *F1* and *F2* are conflicting in nature, thus a single solution that simultaneously minimizes both objectives does not generally exist.

All non-dominated solutions that were found during the experimental runs were recorded. The cumulative results of the experimental runs are presented in Table 2:

Table 2: Experimental results

Multi GP-SLCA solutions		**Venugopal and Narendran solutions**	
F1	*F2*	*F1*	*F2*
0	18647	8.596	918
2.045	18332		
3.772	12759		
5.822	9818		
6.412	6722		
8.596	918		
22.657	455		
38.152	0		

A graphical illustration of all evolved solutions can be found in Figure 1. Multiobjective GP-SLCA automatically evolved a set of non-dominated solutions. Venugopal and Narendran indirectly identified only one of these solutions by manually observing the strings of solutions evolved for the optimization of individual objectives.

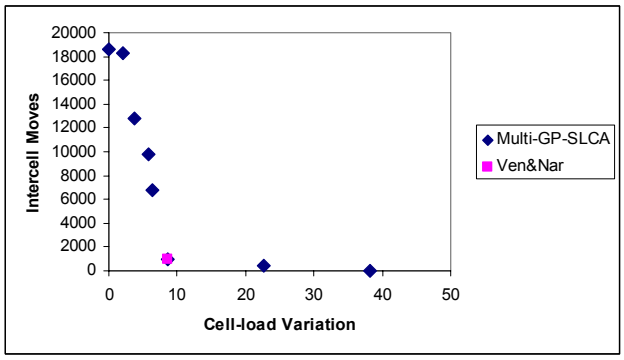

Figure 1: Evolved solutions in the solutions' space.

A closer look at the results of the multiobjective GP-SLCA application reveals that solutions range from the extreme case of each workstation forming an independent cell (*F1*=0), up to the extreme case of all workstations forming a single cell (*F2*=0). The rest of the solutions comprise various cell configurations spread throughout the approximation of the Pareto set. This is the desired condition for multiobjective decision making, since decisions can be made based on the information of the entire trade-off surface.

One of the solutions that was generated through the multiobjective GP-SLCA process is illustrated in Figure 2 (F1=22.657, F2=455). Note that a non-zero entry in the table indicates the workload induced by the part of the corresponding column to the workstation of the corresponding row. The identified cells are enclosed in borders. Due to space limitations it is not possible to present all evolved solutions; however, all cell configurations are available on request.

5 CONCLUSIONS

In this article a novel methodology for the solution of multiobjective cell-formation problems was presented. Multiobjective GP-SLCA combines a modified version of GP-SLCA, a standard methodology for the solution of single-objective cell-formation problems, with NSGA-II, an evolutionary technique for multiobjective optimization. The proposed methodology was applied to a typical test problem taken from the literature. Although a much larger experimental base is needed for a proper evaluation of the methodology, these preliminary results indicate the benefits gained from its application:

- Unlike existing techniques, multiobjective GP-SLCA automatically generates a set of alternative trade-off solutions that can be used by the designer of the cellular manufacturing system.
- The decision maker does not need to pre-specify the total number of cells in the plant. Cell configurations for all possible number of cells are generated during a single experimental run. All existing methodologies require the specification of the total number of cells in the plant before the application of the algorithm on specific problems.
- If cell-size constraints are required by the decision maker, then these can be easily incorporated in multiobjective GP-SLCA by simply penalizing all solutions that violate the constraints. A similar approach

was followed successfully for the single-objective GP-SLCA algorithm [11].

- Multiobjective GP-SLCA provides the decision maker with significant support on the design and redesign of a cellular manufacturing system. Since solutions are not generated based on weighting assumptions, any cost model can be recursively applied to them.

These preliminary results provide promising indications for the efficiency of the multiobjective GP-SLCA methodology; however, as stated earlier, a wider experimental basis is needed for a more detailed evaluation of the proposed methodology. It is in the author's intentions to present the results of this experimentation in the future.

	Parts																												
Workstation	P1	P3	P4	P6	P7	P8	P9	P10	P11	P13	P15	P17	P19	P21	P23	P24	P25	P28	P29	P30	P2	P5	P12	P14	P16	P18	P20	P22	P26	P27
W1		0.3		0.6	0.6	0.2	0.2	0.5	0.7			0.4	0.6																	
W2	0.4	0.5		0.7	0.3	0.4	0.3	0.6	0.8			0.9	0.2																	
W3	0.6	0.7		0.2	0.4	0.9	0.6	0.2	0.2			0.3	0.5																	
W7	0.8	0.9		0.3	0.5	0.5	0.7	0.3	0.5			0.6	0.9																	
W10	0.6	0.2		0.3	0.9	0.2	0.3	0.4	0.5			0.6	0.8																	
W11			0.2							0.3	0.4			0.5	0.9	0.2	0.5	0.6	0.7	0.8										
W12			0.6							0.7	0.8			0.9	0.9	0.3	0.5	0.5	0.6	0.7										
W13			0.7							0.5	0.6			0.8	0.5	0.3	0.4	0.5	0.7	0.8										
W14			0.2							0.6	0.8			1	0.5	0.4	0.6	0.8	0.2	0.8										
W15			0.5							0.7	0.9				0.3		0.7	0.9	0.3	0.4										
W4																					0.2	0.3		0.7	0.6	0.2	0.4	0.4	0.5	0.6
W5																					0.2	0.3	0.4	0.5	0.7	0.8	0.9	0.6	0.8	0.2
W6																					0.8	0.9	1	0.7	0.2	0.3	0.4	0.5	0.6	0.8
W8																					1.1	1.2	0.3	0.8	0.3	0.9	0.2	0.3	0.4	0.5
W9																					0.4	0.5	0.6	0.9	0.5	0.6	0.7	0.8	0.9	1

Figure 2: Cell configuration for a typical solution evolved by multiobjective GP-SLCA (*F1*=22.657, *F2*=455).

6 REFERENCES

[1] Burbidge, J.L., Production Flow Analysis, Production Engineer, 1971, 50, 139-152.

[2] McAuley, J., Machine Grouping for Efficient Production, Production Engineer, 1972, 51, 53-57.

[3] Schaffer, J.D., Multiple objective optimization with vector evaluated genetic algorithms, PhD thesis, 1984, Vanderbilt University.

[4] Kusiak, A., Intelligent Manufacturing Systems, 1990, Prentice Hall, Englewood Cliffs.

[5] Wei, J.C., and Gaither, N., An Optimal Model for Cell-Formation Decisions, Decision Sciences, 1990, 21, 416-433

[6] Venugopal, V. and Narendran, T.T, A genetic algorithm approach to the machine-component grouping problem with multiple objectives, Computers and Industrial Engineering, 1992, 22, 469-480.

[7] Gupta, Y., Gupta, M., Kumar, A. and Sundaram, C., A Genetic Algorithm-based Approach to Cell-Composition and Layout Design Problems, International Journal of Production Research, 1996, 34, 447-482.

[8] Lin, L.T., Dessouky, M.M., Kumar, K.R. and Ng, M.S., A Heuristic-based Procedure for the Weighted Production – Cell-Formation problem, IIE Transactions, 1996, 28, 579-589.

[9] Su, C.-T. and Hsu, C.-M., A Two-phased Genetic Algorithm for the Cell Formation Problem, International Journal of Industrial Engineering, 1996, 3, 114-125

[10] Lee, H. and Chen, Y.-L., A Weighted Approach for Cellular Manufacturing Design Minimizing Intercell Movement and Balancing Workload Among Duplicated Machines, International Journal of Production Research, 1997, 35, 1125-1146

[11] Dimopoulos, C. and Mort, N., 2000, Solving Cell-Formation Problems Under Alternative Quality Criteria and Constraints with a Genetic Programming-based Hierarchical Clustering Algorithm. Proceedings of the Sixth International Conference on Control, Automation, Robotics and Vision, 3445-3446.

[12] Mansouri, S.A., Moattar Husseini, S.M. and Newman, S.T., A Review of the Modern Approaches to Multi-criteria Cell Design, International Journal of Production Research, 2000, 38, 1201-1218.

[13] Van Veldhuizen, D.A. and Lamont, G., Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art, Evolutionary Computation, 2000, 8 125-147.

[14] Wemmerlov, U., and Johnson, D.J., Empirical Findings in Manufacturing Cell Design, International Journal of Production Research, 2000, 38, 481-507.

[15] Zhao, C. and Wu, Z., A Genetic Algorithm for Manufacturing Cell-Formation with Multiple Routes and Multiple Objectives, International Journal of Production Research, 2000, 38, 385-395.

[16] Dimopoulos, C., and Mort, N., A Hierarchical Clustering Methodology Based on Genetic Programming for the Solution of Simple Cell-Formation Problems, International Journal of Production Research, 2001, 39, 1-19.

[17] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., A Fast and Elitist Multiobjective Genetic Algorithm: NSGA—II, IEEE Transactions on Evolutionary Computation, 2002, 6, 182-197.

[18] Eiben, A.E., Smith, J.E., Eiben, A.E. and Smith J.D., Introduction to Evolutionary Computing, 2003, Springer-Verlag

[19] Dimopoulos C., 2004, A Review of Evolutionary Multiobjective Optimization Applications in the Area of Production Research, Proceedings of the Congress on Evolutionary Computation 2004, 1487-1494.

[20] Solimanpur, M., Vrat, P., and Shankar, R., A multi-objective genetic algorithm approach to the design of cellular manufacturing systems, International Journal of Production Research, 2004, 42, 1419-1441.

7 APPENDIX

McAuleys Single Linkage Clustering Analysis (SLCA) Algorithm

	p1	p2	p3	p4	p5
m1	1	0	1	0	0
m2	0	1	0	1	1
m3	1	0	1	0	0
m4	1	1	0	1	0

Figure 3: Example matrix for the illustration of SLCA.

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

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*

Figure 4: Jaccard's similarity coefficient.

$$S_{1,3} = \frac{2}{2+0+0} = 1 \quad S_{1,2} = \frac{0}{0+2+3} = 0$$

$$S_{1,4} = \frac{1}{1+1+2} = 0.25 \quad S_{3,4} = \frac{1}{1+1+2} = 0.25$$

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

Figure 5: Calculation of similarities using Jaccard's similarity coefficient.

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

Figure 6: Similarity matrix for the example problem.

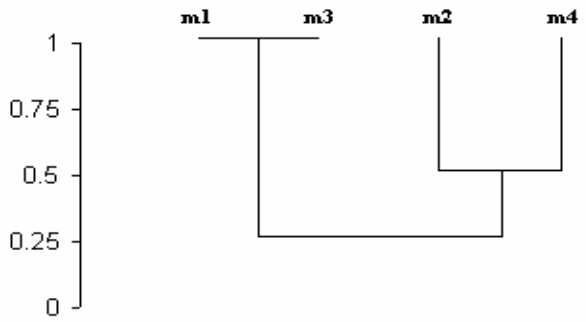

Figure 7: Dendrogram of solutions for the example problem based on the similarity matrix.

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=0.25)	*cell 1*: m_1, m_2, m_3, m_4

Figure 8: Potential solutions for the example problem.