Multi-criterion Tackling Bottleneck Machines and Exceptional Parts in Cell Formation Using Genetic Algorithms

S. A. Mansouri [1], S. M. Moattar-Husseini [1] and S. H. Zegordi [2]

[1] Department of Industrial Engineering,
Amirkabir University of Technology,
Tehran, Iran

[2] Department of Industrial Engineering,
Tarbiat Modarres University,
Tehran, Iran

Abstract

In this paper a Multi-criterion Decision Making (MCDM) model is introduced to tackle bottleneck machines and exceptional parts in manufacturing cell formation. In a given part-machine grouping scheme, bottleneck machines can be eliminated through machine duplication and exceptional parts may be removed by means of subcontracting. The developed multi-criterion model simultaneously takes into account 4 conflicting criteria regarding: inter-cell part movement, total cost of machine duplication and part subcontracting, overall utilization of the cells, and imbalance of the workloads among the cells. A Multi-objective Genetic Algorithm (MOGA) is then developed to seek for non-dominated or non-inferior solutions to assist the decision maker in his/her final selection. Comparative results in a number of cell formation problems show promising capabilities of the proposed solution approach.

1. Introduction

Cellular Manufacturing (CM) is an important application of Group Technology (GT) in which sets (families) of parts are produced in manufacturing cells or a group of various machines, which are physically close together and can entirely process a family of parts. Identification of part families and machine groups in the design of cellular manufacturing systems is commonly referred to as cell design/formation.

Many solution approaches for the common cell formation problem have been proposed during the last three decades. Readers may refer to Mansouri *et al.* (2000) [1] and Offodile *et al.* (1994) [2] to review these solution approaches.

One of the major problems associated with the design of independent manufacturing cells arises due to the existence of Exceptional Elements (EE's), i.e. *exceptional parts* and *bottleneck machines*. The EE's are the major cause of a

number of difficulties in further implementation of cellular systems as; intercellular part movements and unbalance of the workload across the cells. Machine duplication and part subcontracting, have been suggested to overcome the EE's in a cellular design, say by Shaffer *et al.* (1992) [3] and Seifoddini (1989) [4].

In this paper a systematic approach based on Multi-criterion Decision Making is proposed for dealing with the EE's in a given cellular design. The problem is modelled as a Multi-objective Optimization Problem (MOP). The developed model requires simultaneous optimization of competing criteria or objectives in which usually there is no single optimal solution, but rather a set of alternative solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all objectives are considered. They are known as *non-dominated* or *Pareto-optimal* solutions. A Multi-objective Genetic Algorithm (MOGA) is then developed to find non-dominated solutions to the model. A number of cell design problems are solved to evaluate various measures of effectiveness of the proposed algorithm. Finally, corresponding results are presented and conclusion is made.

2. Problem Definition

An exceptional part is a part that can not be completely processed in a single cell. A bottleneck machine is assigned to a cell while needed by some parts from other cells. Figure (1-a) demonstrates initial machine part incidence matrix in a 4 machines, 6 parts problem. In Figure (1-b), the sorted matrix is shown along with a decomposition scheme which separates all the machines and parts in two interdependent clusters as: Cluster 1: {(M2, M1), (P1, P3, P6)}, and Cluster 2: {(M4, M3), (P2, P4, P5)} where M and P stands for Machine and Part, respectively. There are two exceptional parts (P1 and P4) and two exceptional (bottleneck) machines (M2 and M4) in the decomposition scheme proposed in the Figure (2-b).

Parts

Machines	1	2	3	4	5	6
1	1		1			1
2	1		1	1		1
3		1		1	1	
4	1	1		1	1	

a. Initial matrix

Parts

Machines	1	3	6	2	4	5
2	1	1	1		1	
1	1	1	1			
4	1			1	1	1
3				1	1	1

b. A decomposition scheme on the sorted matrix

Figure 1. Initial and final machine-part incidence matrix

2.1. Assumptions

It is assumed that the cellular system is to be formed based on an existing Job Shop facility in which a grouping scheme is proposed for machine cells and part families. Moreover part subcontracting and machine duplication have been considered as two possible alternatives for the elimination of EE's. Finally it is assumed that partial subcontracting is not allowed, that is if an exceptional part is to be subcontracted, all of its demand should be supplied by subcontractors.

2.2. Problem Formulation

In this section, the multi-criterion model for tackling the EE's is formulated.

2.2.1. Notation

- *Set of indices*
  i: Index for machine types, $i=1,...,m$
  j: Index for part types, $j=1,...,p$
  k: Index for cells, $k=1,...,c$

- *Decision variables*
  Two binary decision variables are defined to formulate the problem as : $X_j = 1$ if part j is subcontracted and $X_j=0$ otherwise; $Y_{ik}=1$ if machine i is duplicated in cell k and $Y_{ik}=0$ otherwise.

- *Set of parameters*

D_j	:	*annual demand for part j;*
S_j	:	*incremental cost of subcontracting a unit of part j;*
t_{ij}	:	*processing time of a unit of part j on machine I;*
PM_{ji}	:	*number of intercellular transfers required by part j as a result of machine type i not being available within the part's manufacturing cell;*
M_i	:	*annual cost of acquiring an additional machine I;*
CM_i	:	*annual machining capacity of each unit of machine i (minutes);*
HF_k	:	*set of parts assigned to cell k;*
MC_k	:	*set of machines assigned to cell k;*
GF_k	:	*set of parts assigned to the cells other than k but require some of the machines in cell k;*
BM_k	:	*set of the bottleneck machines required by the parts in cell k;*
EP_k	:	*set of exceptional parts in cell k;*
EM_j	:	*set of bottleneck machines required by the exceptional part j;*
CS_k	:	*number of machines assigned to cell k;*
MCS	:	*maximum cell size;*
c	:	*number of cells.*

2.2.2. Objective Functions

This sub-section gives a brief explanation of the importance and formulation of the objectives included in the MCDM model:

Objective 1: minimizing intercellular parts movement

Intercellular movement of parts is one of the major problems associated with the EE's in a cellular design, which complicates production and inventory management functions. Minimization of the intercellular parts movement is sought through the following objective function:

$$Min \quad f_1 = \sum_{k=1}^{c} \sum_{j \in EP_k} \left(1 - X_j\right) \times \left(\sum_{i \in EM_k} PM_{ji} \times \left(1 - Y_{ik}\right) \right) \tag{1}$$

Objective 2: minimizing total cost of machine duplication and part subcontracting

Any reduction in intercellular parts movement by machine duplication and / or part subcontracting will result in cost increment. Hence minimization of sum of part subcontracting and machine duplication cost is included in the model as follows:

$$Min \quad f_2 = \sum_{k} \left(\sum_{j \in EP_k} \left(D_j \times S_j \times X_j\right) + \sum_{i \in EM_k} M_i \times Y_{ik} \right) \tag{2}$$

Objective 3: minimizing overall machine under-utilization

Since machine duplication and / or part subcontracting will deteriorate level of utilization in the cellular system, minimization of the overall machines under-utilization, which is equivalent to maximization of their overall utilization, is taken into account employing the following objective function:

$$Min \quad f_3 = 1 - OU = 1 - \frac{\sum_{k=1}^{c} UC_k \cdot \left(CS_k + \sum_{i \in BM_k} Y_{ik} \right)}{\sum_{k=1}^{c} \left(CS_k + \sum_{i \in BM_k} Y_{ik} \right)} \tag{3}$$

where *OU* stands for the *Overall Utilization* and:

$$UC_k = \frac{\sum_{i \in MC_k} \left(\sum_{j \in HF_k} D_j \times t_{ij} - \sum_{j \in EP_k} D_j \times t_{ij} \times X_j + \sum_{j \in GF_k} D_j \times t_{ij} \times \left(1 - X_j\right) \right)}{\sum_{i \in MC_k} CM_i + \sum_{i \in BM_k} Y_{ik} \times CM_i} \tag{4}$$

denotes the *Utilization of Cell k.*

Objective 4: minimizing deviations among utilization of the cells

Significant differences in the level of utilization among the cells may result in major problems in the managerial functions the following objective function is included to minimize deviations among the cells' level of utilization:

$$Min \quad f_4 = \frac{\sum_{k=1}^{c} (UC_k - OU)^2}{c-1} \tag{5}$$

Among the aforementioned objectives, minimizing intercellular parts movement (objective 1) is of special importance due to the fact that intercellular movements are the main cause of cells interdependencies. However any effort to reduce intercellular parts movement by means of machine duplication and part subcontracting, increases cost, deteriorates overall utilization of machinery, and imbalances levels of utilization among the cells. Objectives 2, 3 and 4 have been included in the model to overcome these side effects, respectively.

2.2.3. Constraints

Two sets of constraints are included in the model as follows:

$$(CS_k + \sum_{i \in BM_k} Y_{ik}) \leq MCS \quad , \quad k = 1,...,c \tag{6}$$

$$X_j, Y_{ik} \in \{0,1\} \quad , \quad \forall \ i, j, k \tag{7}$$

Constraints (6) prevent cell sizes from exceeding a pre-determined upper bound. Relations (7) restrict the decision variables to get either a value of 0 or 1.

Complexity of the model along with extensive computations required to find satisfactory solutions, justify application of an efficient solution approach especially in the real world problems. For this purpose, a solution approach based on Multi-objective Genetic Algorithms was developed, which will be discussed in the subsequent section.

3. The Proposed Genetic Algorithm Approach

In simple GAs, a candidate solution is represented by a sequence of *genes* and is known as a *chromosome*. A chromosome's potential as a solution is determined by its *fitness function* that evaluates a chromosome with respect to the objective function of the optimisation problem at hand. A judiciously selected set of chromosomes is called a *population* and the population at a given time is a *generation*.

In order to apply genetic algorithms to the developed MCDM model in a problem with *n* decision variables, a chromosomal structure consisting *n* genes is considered. Each gene in the chromosome may take either a value of "0" or "1" that reflects value of its corresponding binary decision variable.

The objective values are normalized so that they lie in the interval of 0 and 1 by means of the following formula:

$$F_i = \frac{C_i}{C_i + f_i} \quad , \quad i = 1,...,4 \tag{8}$$

where: F_i is the fitness value, f_i is the objective value and C_i is the normalizing factor concerning the objective i.

3.1. The XGA Algorithm

This section introduces the main steps of the developed MOGA called XGA.

3.1.1. Notation

Following notation have been used in describing the XGA algorithm:

Pop	:	*Population;*
PS	:	*Population Size;*
ENS_i	:	*Expected Number for the Selection of individual i;*
MP	:	*Mating Pool;*
CNDF	:	*Current Non-Dominated Front;*
DFV	:	*Dummy Fitness Value;*
NC_i	:	*Niche Count of the individual* i *;*
$d_{i,j}$	:	*Distance between the individuals i and j;*
NP	:	*Niching Parameter;*
$Sh[d_{i,j}]$	:	*Sharing value between the individuals i and j;*
SFV_i	:	*Shared Fitness Value of the individual i;*
$PSelect_i$	:	*Probability for the Selection of the individual i;*
ES	:	*Elite Set;*
ITP	:	*Initial Transfer Probability;*
EN	:	*Epsilon Niche of the individuals in the Elite Set (ES);*
DF	:	*Degrading Factor for Transfer Probabilities of the individuals in the EN of a selected individual;*
MS	:	*Mating Set;*
ESS	:	*Elite Set Size;*
CR	:	*Crossover Rate;*
IP	:	*Inversion Probability;*
MR	:	*Mutation Rate;*
SNDF	:	*Successive Non-Dominated Fronts (number of successive generations in which all non-dominated solutions of the current generation have remained non-dominated when compared against the non-dominated frontiers of previous generations);*
Min_SNDF	:	*Minimum number of the SNDF.*

3.1.2. Steps of the XGA Algorithm

Major steps of the XGA algorithm is illustrated in a flow chart presented in Figure 2. Some steps of the algorithm are discussed in more details in the following sub-sections.

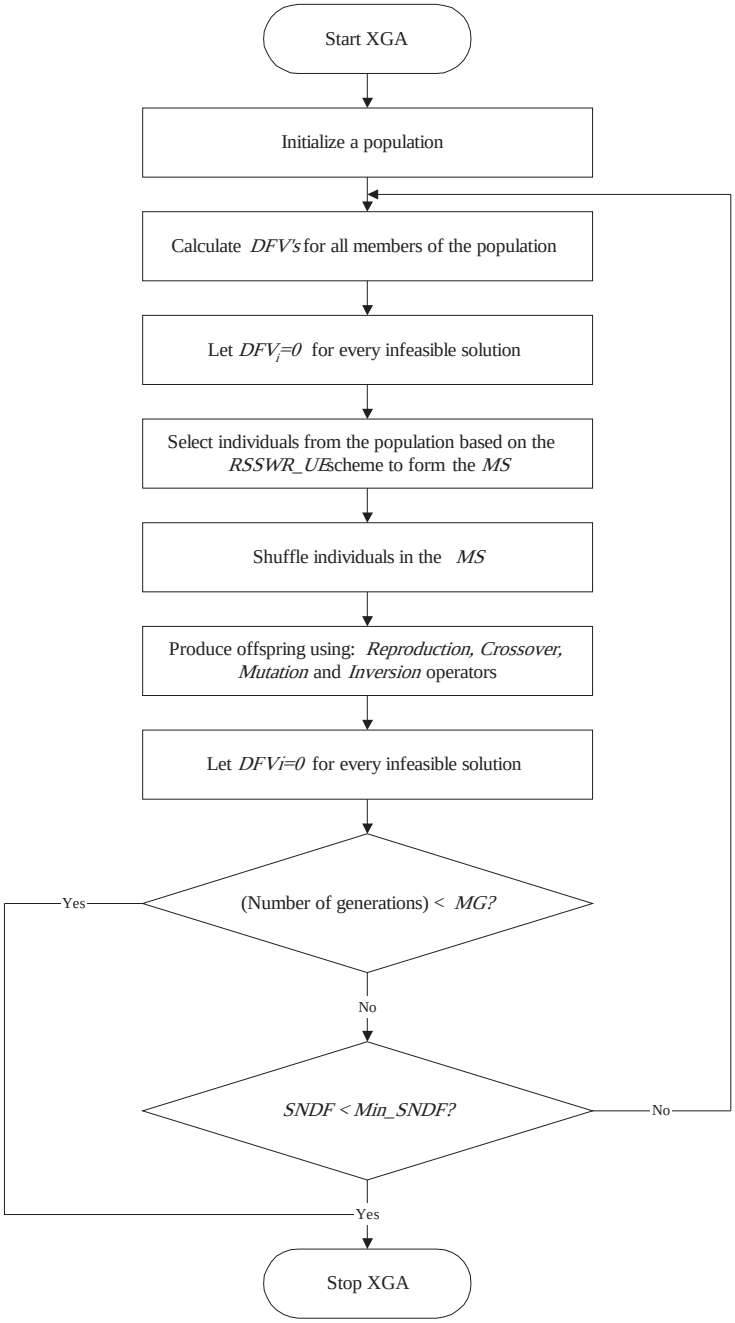

Figure 2. Flow chart of the XGA algorithm

3.1.3. Calculation of Shared Fitness Values

The *Shared Fitness Values (SFVs)* are calculated based on the non-dominated sorting method of Srinivas and Deb (1994) [5].

3.1.4. Selection

For selection, a novel scheme is employed that makes use of the *Reminder Stochastic Sampling Without Replacement* in conjunction with a new *Elitism* operator, called *RSSWR_UE*. Major steps of the selection scheme are depicted in Figure 3.

3.1.5. Recombination

All selected individuals are then shuffled and mutually recombined, with probability *CR*, through single-point crossover. In single-point crossover, the two selected parents are cut from a random point along their length into two sections. Section 1 of parent 1(2) attaching section 2 of parent 2(1) form offspring 1(2). A small portion of genes in the population are then mutated according to the probability *MR* from "1" into "0" and *vice versa* through the mutation operator.

3.1.6. Updating the Elite Set

In many multi-objective problems, size of the elite set and how to update the set so that its size does not exceed a predetermined limit have significant effect on the elitism. The updating mechanism of the *Elite Set* in the XGA algorithm employs the notion of niching to improve diversity of the individuals in the set.

3.1.7. Stopping Criteria

The algorithm terminates either as soon as it has been converged to a robust non-dominated front or a predetermined number of generations *(MG)* have been completed. A robust non-dominated front has been reached if the *SNDF* reaches the *Min_SNDF*. The *SNDF* is updated in each generation.

3.2. Parameter Setting

In order to find a good set of values for the parameters of the XGA, two medium-sized cell design problems were selected. Their true non-dominated fronts were found through total enumeration and employed as the references for the evaluation of the algorithm. Three measures for judgement on the effectiveness of the set of parameters were used as follows:

MP_1 : *Quality of non-dominated solutions: ratio of true non-dominated solutions in the final non-dominated front of the algorithm*

MP_2 : *Diversity of solutions in the final non-dominated front, measured by the number of solutions in the front*

MP_3 : *CPU time*

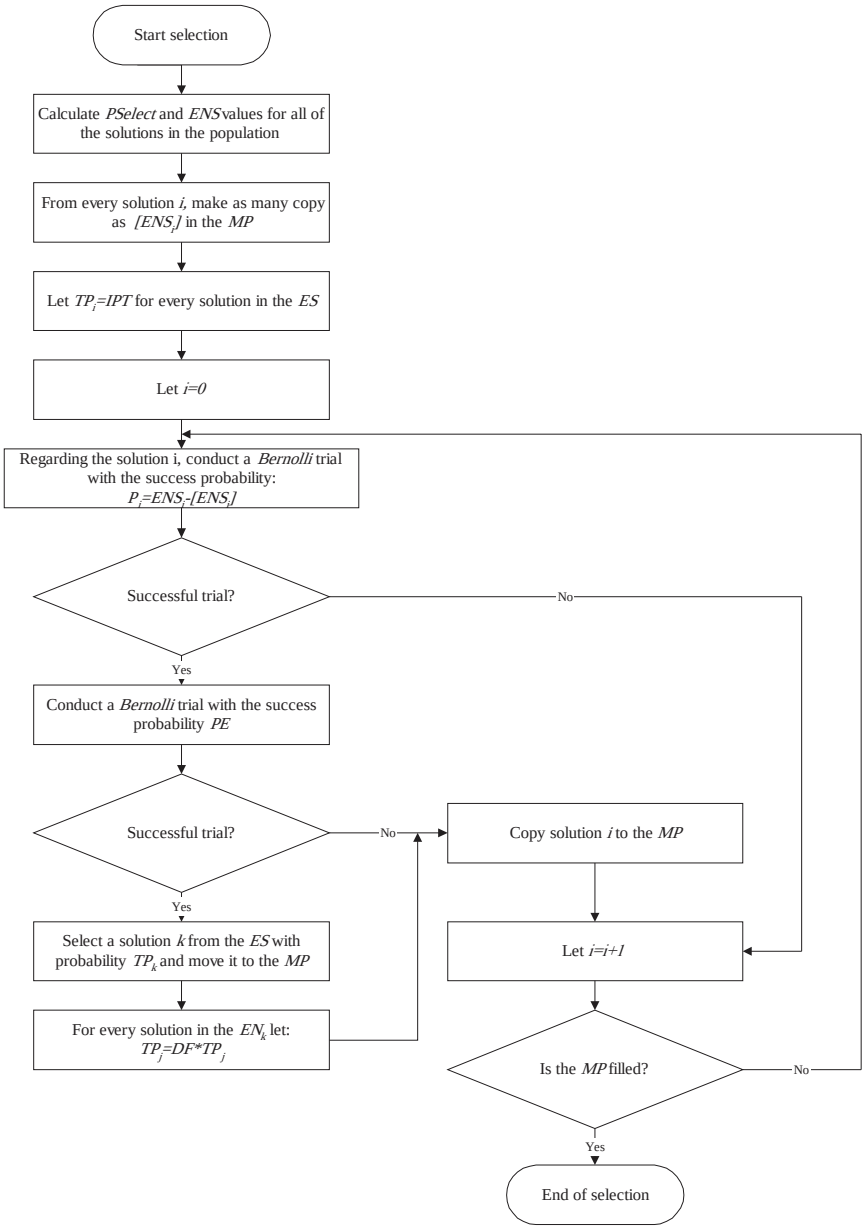

Figure 3. Flow chart of the selection scheme

Considering these measures, extensive experiments were conducted with various combinations of the parameters and the set of parameters was selected as: *PS=150, CR=0.50, MR=0.03, NP=0.60, IP=0.00, Min_SNDF=15, EP=1.00, ITP=0.10, EN=0.30, ESS=50* and *DF=0.80.*

4. Evaluation of the XGA Algorithm

In order to evaluate the XGA algorithm, 6 cell formation problems in different sizes were selected from the literature. Major characteristics of the test problems are presented in Table 1. The size of solution space associated with the test problems ranges from a space having $2^{10} = 1024$ solutions to a space with $2^{43} = 8.796\times 10^{12}$ solutions. Moreover maximum number of mutual comparisons required for total enumeration of these problems, ranges from $(2^{10}!)/[(2!)(2^{10}-2)!] = 523{,}776$ to $(2^{43}!)/[(2!)(2^{43}-2)!] = 3.869\times 10^{25}$ where complete enumeration is impossible.

Table 1. Main characteristics of test problems

Problem	Reference	Number of:			
		Decision Variables	Machines	Parts	Cells
VN92	[7]	10	15	30	3
Bur69	[8]	15	20	35	5
KLA97	[9]	19	24	40	7
ACGV91	[10]	30	12	19	3
Sei89	[4]	35	16	43	5
BC91	[11]	43	20	35	4

Performance of the XGA was compared against three multi-objective genetic algorithms, namely: VEGA (Schaffer 1985 [12]), NPGA (Horn and Nafpliotis 1993 [13]) and NSGA (Srinivas and Deb 1994 [5]). All the algorithms were coded in C++ and implemented on a Pentium-II (Celeron) CPU at 333 MHZ with 64 MB of RAM under Windows 2000. For the reference algorithms the parameter set employed by Zitzler *et al.* (2000) [6] was adopted. Zitzler *et al.* (2000) [6] employed these set to compare a number of MOGAs, including those selected for our comparisons, on a number of multi-objective test problems. It should be noted that no further tuning were applied to the reference algorithms for current research.

To evaluate quality (MP_1) of the non-dominated front found by the algorithm, a reference set for every problem was formed. In small to medium sized problems, i.e. the problems with less than or equal to 15 decision variables, the reference sets werecreated through total enumeration. Concerning the large problems, i.e. the problems with more than 16 variables, where total enumeration was practically impossible, a refining scheme was devised to establish a set of near non-dominated frontiers. In the refining scheme, successive runs of the XGA were conducted, each run using a randomly selected set of parameters. Non-dominated solutions of the first run were adopted as a preliminary reference set. Adding non-dominated solutions of the next run into the reference set, a mutual dominance check was performed between the old members of the set and the new entrants. Dominated solutions were removed and the remaining solutions formed the new reference set. This procedure was repeated 50 times for each problem and the final reference set was adopted for later comparisons of the algorithm.

Quality of each run (MP_1) of an algorithm is then calculated by comparing the final results against the corresponding reference sets. The diversity was simply measured by the number of non-dominated solutions found and represented by MP_2.

Table 2. Comparative results

Problem	Performance Measures of the Algorithms											
	NPGA			VEGA			NSGA			XGA		
	MP_1	MP_2	MP_3	MP_1	MP_2	MP_3	MP_1	MP_2	MP_3	MP_1	MP_2	MP_3
VN92	0.532	18.3	171.7	0.467	8.8	31.2	0.766	22.8	32.3	0.983	30.6	18.9
Bur69	0.087	28.3	247.2	0.071	8.5	37.5	0.700	54.6	38.2	0.789	75.3	22.8
KLA97	0.032	33.5	303.8	0.051	13.4	44.8	0.240	50.9	47.9	0.396	74.8	17.7
ACGV91	0.046	56.1	514.1	0.441	29.7	33.3	0.220	67.9	40.1	0.599	101.4	22.5
Sei89	0.073	47.4	563.0	0.445	35.5	48.6	0.359	67.0	58.2	0.695	106.5	30.7
BC91	0.029	57.5	701.1	0.169	34.1	53.6	0.216	69.2	67.2	0.596	97.1	24.8

CPU time *(MP_3)* was also measured and used as the third measure. Each problem was then solved 20 times by each algorithm. Table 2 presents the average results of these runs.

In order examine the mean difference between the measures of XGA and those of the reference algorithms (using the parameter set of Zitzler *et al.* (2000) [6]), pair-wise comparisons were made at significance level $\alpha = 0.05$ employing one-tail t-test. Table 3 shows summary of these comparisons.

As it can be derived from Table 3, XGA with confidence level of 95% dominates the reference algorithms when parameter set of Zitzler *et al.* (2000) [6] is used. It is superior to the reference algorithms regarding MP1 measure from 22.2% to 65.1%. In this relation, NSGA comes next followed by VEGA and NPGA respectively. Concerning MP2, XGA shows better performance over the others from 19.8% to 53.1%. NSGA once again takes the second position, followed by NPGA and VEGA. With respect to the MP3 measure, XGA outperforms the reference algorithms from 28.9% up to 54.2%. Considering this measure, VEGA takes the next place followed by NSGA and NPGA respectively.

Table 3. Summary of paired comparisons between XGA and reference algorithms

Reference algorithms	Superiority of XGA over reference algorithms at $\alpha = 0.05$ based on:		
	MP1	MP2	MP3
NSGA	22.2%	19.8%	31.7%
VEGA	34.0%	53.1%	28.9%
NPGA	65.1%	33.4%	54.2%

5. Conclusion

In this paper, the problem of dealing with exceptional elements in the design of manufacturing cells was addressed as a Multi-criterion Decision Making (MODM) problem. Due to the conflicts among the objectives and complexity of the developed model, a Multi-objective Genetic Algorithm (MOGA) was developed to seek for non-dominated or non-inferior solutions. The developed MOGA was evaluated in a

number of cell design problems and its results compared with those of three other MOGAs. The obtained results were promising in three aspects of performance, i.e. quality, diversity and CPU time.

References

1. Mansouri SA, Moattar-Husseini SM, Newman ST, (2000) A review of the modern approaches to multi-criteria cell design. International Journal of Production Research, 38(5): 1201-1218.
2. Offodile OF, Mehrez A, Grznar J, (1994) Cellular manufacturing: a taxonomic review framework. Journal of Manufacturing Systems. 13: 196-220.
3. Shaffer SM, Kern GM, Wei JC, (1992) A mathematical programming approach for dealing with exceptional elements in cellular manufacturing. International Journal of Production Research , 30: 1029-1036.
4. Seifoddini H, (1989) Duplication process in machine cells formation in group technology. IIE Transactions, 21 (4): 382-388.
5. Srinivas N, Deb K, (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 2(3): 221–248.
6. Zitzler E, Deb K, Thiele L, (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evolutionary Computation, 8(2): 173-195.
7. Venugopal V, Narendran TT, (1992) A genetic algorithm approach to the machine-component grouping problem with multiple objectives. Computers and Industrial Engineering, 22(4): 469-480.
8. Burbidge JL, (1969) An introduction of group technology. Proc. Seminar on GT, Turin, Italy.
9. Kazerooni M, Loung HS, Abhary K, (1997) A genetic algorithm based cell design considering alternative routing. Computer Integrated Manufacturing Systems, 10(2): 93-107.
10. Askin RG, Cresswell SH, Goldberg JB, Vakharia AJ, (1991) A Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing. International Journal of Production Research, 29 (6): 1081-1100.
11. Boe WJ, Cheng CH, (1991) A close neighbor algorithm for designing cellular manufacturing systems. International Journal of Production Research, 29 (10): 2097-2116.
12. Schaffer JD, (1985) Multiple objective optimization with vector evaluated genetic algorithms. Proc. 1st Int. Conf. Genetic Algorithms and their Applications, Carnegie-Mellon University, Pittsburgh, USA: 93-100.
13. Horn J, Nafpliotis N, (1993) Multiobjective optimization using the niched pareto genetic algorithm. IlliGAL Report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana Champaign.