INT. J. PROD. RES., 01 APRIL 2004, VOL. 42, NO. 7, 1419–1441

A multi-objective genetic algorithm approach to the design of cellular manufacturing systems

MAGHSUD SOLIMANPUR†, PREM VRAT‡ and RAVI SHANKAR§*

In this paper, a multi-objective integer programming model is constructed for the design of cellular manufacturing systems with independent cells. A genetic algorithm with multiple fitness functions is proposed to solve the formulated problem. The proposed algorithm finds multiple solutions along the Pareto optimal frontier. There are some features that make the proposed algorithm different from other algorithms used in the design of cellular manufacturing systems. These include: (1) a systematic uniform design-based technique, used to determine the search directions, and (2) searching the solution space in multiple directions instead of single direction. Four problems are selected from the literature to evaluate the performance of the proposed approach. The results validate the effectiveness of the proposed method in designing the manufacturing cells.

1. Introduction

Cellular manufacturing (CM) has been recognized as a methodology for organizing the design and operation of a wide range of manufacturing systems so that the advantages of mass production and flexibility of job shop manufacturing can be derived from the production system. A typical CM environment processes a wide variety of parts that have common features. Experience has shown CM to be successful in many diverse environments (Wemmerlöv and Hyer 1989).

Cellular manufacturing has considerable influence on the performance of production systems (Singh and Rajamani 1996). Some of the benefits of CM reported in the literature are: low production cost, low material handling cost, low production time, reduction in work-in-process (WIP) inventories, simple production control, reduction in scrap and waste, decentralization of responsibility, saving manufacturing space, etc. (Wemmerlöv and Hyer 1989, Kusiak 1990, Heragu 1994, Shankar and Vrat 1999).

One of the key issues in the design of manufacturing cells is how to group similar parts in part families (PFs) and their associated machines in machine cells (MCs). This issue is known as the cell formation problem (CFP) in the literature. Different methodologies such as classification and coding systems, similarity coefficient-based

Revision received August 2003.

†Mechanical Engineering Group, Faculty of Engineering, Urmia University, Urmia, West Azerbaijan Province, IRAN.

‡Director, Indian Institute of Technology Roorkee, Roorkee-247 667, Uttaranchal, INDIA.

§Department of Management Studies, Indian Institute of Technology Delhi, New Delhi-110 016, INDIA.

*To whom correspondence should be addressed: e-mail: ravi1@dms.iitd.ernet.in, ravi1shankar@yahoo.com

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540310001638073

methods, array analysis methods, mathematical programming, graph theory, expert systems, neural networks, genetic algorithms, fuzzy set theory, simulation, etc. and also numerous heuristics have been proposed in the literature to solve the cell formation problem. A review of different methods developed for the CFP can be found in Singh (1993) and Selim *et al*. (1998).

Because of the complexity and computational hardness of CFP, a majority of the methods in the literature attempt to optimize only one objective in cell formation (Shankar and Vrat 1998). However, this problem is a multi-objective optimization problem and its solution affects different operational aspects of manufacturing systems such as material handling cost, production cost, throughput time, WIP, machine utilization, etc. (Soleymanpour *et al*. 2002). Ballakur and Steudel (1987) outline nine objectives in CFP. These are: minimization of intercellular material handling costs, minimization of setup times, maximization/minimization of a similarity/dissimilarity measure, minimization of total production cost, minimizing the number of exceptional elements, maximizing the utilization of machines and minimizing machine idle times. Mansouri *et al*. (2000) reviewed the literature with respect to the multi-objective methodologies proposed for cellular manufacturing systems design. Study and analysis of the existing multi-objective methods proposed for CFP reveals that mostly the weighted sum of objectives is used to convert multiple objectives into a single objective. In these approaches, no systematic way is generally reported to set the weights for each objective. The weights are either user-defined or randomly generated. Moreover, the available multi-objective methods provide a single cell configuration, which might be impossible to implement in reality. These facts limit the applicability of existing multi-objective methods proposed for CFP. In this paper, a mathematical model is proposed for CFP and a multi-objective genetic algorithm is developed to solve the problem. In the proposed algorithm, the solution space is searched in multiple directions and the uniform design technique is used to determine the weights of objective functions.

This paper is organized as follows. In section 2, the related research in multi-objective genetic algorithms and the applications of genetic algorithms to the CFP are reviewed. Section 3 provides a brief background about the search vectors and uniform design technique. In section 4, the CFP is formulated as a multi-objective non-linear integer programming model. The elements of the proposed multi-objective genetic algorithm are developed in section 5. In section 6, all the elements proposed in section 5 are synthesized to evolve a unified algorithm. In section 7, the performance of the proposed algorithm is evaluated through solving four problems selected from the literature. Section 8 includes discussions and conclusions.

2. Literature review

Recent developments in multi-objective evolutionary algorithms are quite extensive and rapidly growing. There are many multi-objective evolutionary algorithms such as those due to Schaffer (1985), Hajela and Lin (1992), Horn and Nafpliotis (1993), Srinivas and Deb (1994), etc. in the literature. A review and classification of different multi-objective evolutionary algorithms can be found in Fonseca and Fleming (1995) and Coello (1999). The salient point indicating the differences between these approaches is due to the strategy by which the fitness of each chromosome is assigned. Zitzler *et al*. (2000) classified different multi-objective evolutionary algorithms into three categories including criterion selection, aggregation selection and Pareto selection. The algorithms employing a criterion selection strategy, e.g. the

vector evaluated genetic algorithm (VEGA) proposed by Schaffer (1985), switch between the objectives during the selection phase. In VEGA a certain fraction of the population appearing in the mating pool is selected with regard to each objective. The methods performing aggregation selection use conventional multi-objective optimization techniques where multiple objectives are combined into a scalar objective function. Pareto selection-based algorithms use the systematic definition of Pareto solutions to rank the solutions in the current population. There are different rules for ranking Pareto solutions in the literature (e.g. Goldberg 1989, Srinivas and Deb 1994). Our approach in this paper falls into the first category in which the selection phase is based on fitness functions. The VEGA approach is criticized because it clusters final solutions around the best solution with respect to each objective (Fonseca and Fleming 1995). This is mainly due to the fact that in VEGA each fraction of the next mating pool is selected based on one objective at a time and other objectives are ignored (Srinivas and Deb 1994). In the approach proposed in this paper, selection is based on fitness functions, which are combinations of all objectives. Therefore, it is expected that our approach would overcome the limitation mentioned above. The difference between the proposed and the VEGA approaches is further discussed in section 5.6. The reasons for choosing the fitness functions-based selection approach are as follows:

(1) The proposed selection approach provides the possibility for the designer to define the preferences by giving desired search directions. However, in Pareto selection approaches, all the non-dominated solutions are given equal ranks to appear in the next mating pool.
(2) The number of objective functions in CMS design is large and Pareto ranking-based methods are not expected to work well for optimizing problems with many competing objectives (Horn and Nafpliotis 1993, Fonseca and Fleming 1995).
(3) The Pareto frontier in CFP is discrete and Zitzler *et al*. (2000) have shown that for these kinds of problems VEGA performs better than some other Pareto ranking-based approaches such as niched Pareto genetic algorithm (NPGA) proposed by Horn and Nafpliotis (1993).

It is well known that multi-objective genetic algorithms are among the most useful approaches for multi-objective non-linear discrete optimization problems. This fact makes multi-objective genetic algorithms suitable for solving the CFP as the objective functions are non-linear and the decision variables are integer in the CFP. Recent research on the applications of evolutionary algorithms to solve the CFP have been reviewed in Dimopoulos and Zalzala (2000). Genetic algorithms-based attempts at the cell formation problem will now be reviewed.

Venugopal and Narendran (1992) used a genetic algorithm to solve a bi-objective integer programming cell formation problem. The uniform mutation and simple crossover operators are used to minimize the total number of intercellular moves and the total intra-cell workload variation. Gupta *et al*. (1995) developed a genetic algorithm to minimize the weighted sum of inter-cell and intra-cell moves. An acceptable level of machine utilization is considered to assign parts into manufacturing cells. Gupta *et al*. (1996) considered different cell layouts and used a genetic algorithm to minimize the total number of intra-cell and inter-cell moves and cell load variation. Al-Sultan and Fedjki (1997) applied a genetic algorithm to solve a quadratic assignment model and minimize dissimilarity between the parts. They only considered

machine requirement data in their study. Su and Hsu (1998) used simulated annealing and a genetic algorithm to solve a multi-objective cell formation problem. They considered the following objectives: machine investment cost, inter-cell and intra-cell material handling costs and inter-cell and intra-cell machine imbalances. User-defined weights are used to convert multiple objectives into a single objective. Gravel *et al.* (1998) considered alternate process plans for products and proposed a double-loop genetic algorithm to generate an efficiency frontier of a cell formation problem with two objectives, viz. minimizing inter-cell transfers and intra-cell load balance among machines. Moon and Kim (1999) used a genetic algorithm to maximize the total number of parts flowing between the machines within the same cell. They considered different data such as production volume, cell size, the capacity of the material handling device, etc. Lee-Post (2000) used a genetic algorithm for handling a parts coding and classification scheme, namely DCLASS. The sum of the similarities between the parts is used to evaluate the fitness of strings. Each solution is represented by five digits in which a digit '1' means that the corresponding attribute is to be considered in part family identification. The genetic algorithm is then applied to find the optimum differentiating attributes. The average linkage clustering is used to form part families. Zhao and Wu (2000) used a genetic algorithm to solve a multi-objective cell formation problem. User-defined weights are used to convert multiple objectives into a single objective. The objectives in their research are: total number of exceptional elements, the total within-cell workload variation and the total intra-cell/inter-cell movements. Plaquin and Pierreval (2000) proposed an evolutionary algorithm for the CFP in the case of some specific constraints. The main attempt in their study is to formulate conditions under which particular machines should be or should not be located in the same cell. Uddin and Shanker (2002) used a genetic algorithm to minimize the total number of inter-cellular moves in the presence of multiple process plans for each part. In this approach, two inter-related problems, one pertaining to the assignment of machines to cells and the second relating to the assignment of process routes, are iteratively solved until convergence is achieved.

There are three aspects that make the proposed approach different from other approaches.

(1) In the majority of the existing approaches, user-defined or randomly generated weights are used to convert multiple objectives into a single objective (Mansouri *et al.* 2000). In this paper, we consider a systematic uniform design-based approach to set the weights of objectives.
(2) In many of the existing genetic algorithms for the CFP, the solution space is searched in only one direction. That is, only one weight is considered for each objective and finally only one solution is obtained. In the algorithm proposed in this paper, the solution space is searched in multiple directions, i.e., multiple weights are considered for each objective function and therefore several final solutions are obtained. This feature provides an opportunity for the designer to choose the most appropriate and practical solution.
(3) The mathematical model developed in this paper is relatively comprehensive and the formed cells are independent. The importance of independent cells in the CMS environment is also identified by Gen and Cheng (2000). They have stated the following advantages with the adoption of independent cells: (a) in some cases, especially in labour-intensive environments, machines are inexpensive and usually small so that duplication will not be costly or space

occupation will not be a major concern; (b) independent manufacturing cells provide a simple manufacturing infrastructure that facilitates shop floor automation, production planning and scheduling, maintenance, etc.; and (c) independent cells might be the only cell type allowed in environments such as pharmaceutical, medical and military industries in which products are not allowed to leave the cell and share machines in other cells due to traceability requirements. Moreover, identification of responsibilities is a crucial problem in labour-intensive environments and can be satisfactorily overcome by adoption of independent manufacturing cells. Justification for selecting independent cells also lies in a focused survey of US companies working in the CMS environment in which Wemmerlöv and Hyer (1989) had reported that 80% of the companies with manned cells and 86% of those with unmanned cells had independent cells. This indicates that the approach proposed in this paper is of potential use in a wide range of industries.

3. Theoretical background

3.1. *Non-dominated solution and search vectors*

Definition: Let us represent the objective space of a multi-objective minimization problem as follows:

$$Z = \left\{ \mathbf{z} \in R^q \middle| z_1 = f_1(\mathbf{x}), \quad z_2 = f_2(\mathbf{x}), \ldots, \quad z_q = f_q(\mathbf{x}); \quad \mathbf{x} \in \Omega \right\}$$

where $\mathbf{x}$ is a decision variable and Ω stands for the feasible solution space and q is the number of objectives. A point $\mathbf{z}^0 \in Z$ is called a non-dominated solution if and only if there does not exist a point $\mathbf{z} \in Z$ such that:

$$z_k < z_k^0 \quad \text{for some } k \in \{1, 2, \ldots, q\}$$
$$z_l \leq z_l^0 \quad \text{for all } l \neq k$$

In other words, a point $\mathbf{z^0}$ is called a non-dominated solution if and only if there is no other solution in the feasible solution space, which is better than $\mathbf{z^0}$ with respect to all the objectives. To search the objective space, a weighting vector is mainly used. When there is only one weighting vector the objective space is searched in one direction. Figure 1(a) shows a mono-directional search scheme. In a genetic algorithm, mono-directional search may lead to a few points in a Pareto optimal frontier. These points are shown within the circle in figure 1(a). However, multi-directional search is used to find more points distributed along the Pareto optimal frontier. Figure 1(b) shows a multi-directional search scheme. The next issue in multi-directional search is related to finding search vectors. In the literature, mostly user-defined or randomly generated vectors are used to search the solution space. In this paper, however, a uniform design method is used to construct uniformly directed vectors.

3.2. *Uniform design*

The main attempt in uniform design is to sample a small set of points from a given large set of points. Consider a space with L variables and K possible values for each variable. Then there are K^L points (combinations) in this space. The uniform design selects K points out of K^L points such that the selected points are scattered uniformly over the space of K^L points. The selected K points are denoted by a

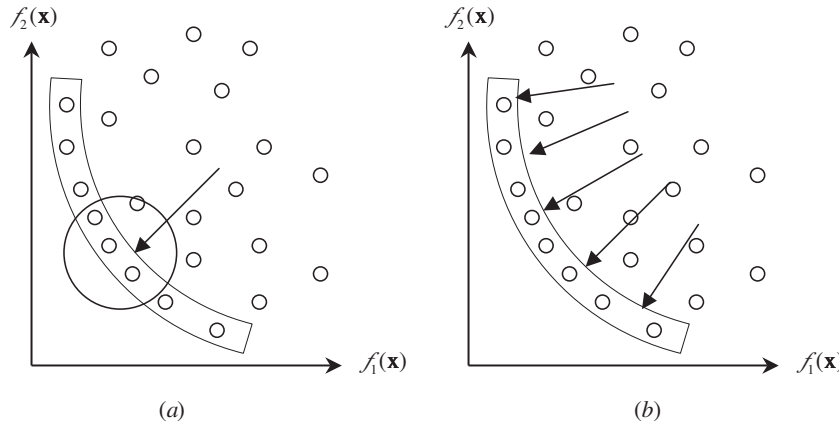

Figure 1. (*a*) Mono-directional search; (*b*) multi-directional search with uniformly distributed search directions.

uniform matrix, $U = [u_{ij}]_{K \times L}$, where u_{ij} is the value of variable j at point i. It can be shown (Leung and Wang 2000) that when K is prime and $K > L$, u_{ij} is given by

$$u_{ij} = \left(i\sigma^{j-1} \bmod K\right) + 1, \tag{1}$$

where σ is a parameter given in table 1. Here L is considered as the number of objective functions and K as the number of search directions (fitness functions).

4. Mathematical formulation of the CFP

The following notation is used throughout this paper to describe the proposed multi-objective integer programming model as well as the multi-objective genetic algorithm.

4.1. *Notation*

i, j = indices for process plans of parts p and q, respectively, $i = 1, 2, \ldots, I_p$; $j = 1, 2, \ldots, I_q$
k = index for fitness functions, $k = 1, 2, \ldots, K$
l = index for objective functions, $l = 1, 2, 3, 4$
m = index for machines, $m = 1, 2, \ldots, M$
n = index for chromosomes in current population, $n = 1, 2, \ldots,$ size_pop
p, q = indices for two different parts, $p, q = 1, 2, \ldots, P$
c = index for cells, $c = 1, 2, \ldots, C$
P = total number of parts
M = total number of machines
C = total number of cells
D_p = demand of part p
I_p = number of process plans of part p.

$$x_{pc}^{i} = \begin{cases} 1 & \text{if part } p \text{ uses process plan } i \text{ in cell } c \\ 0 & \text{otherwise} \end{cases}$$

$$a_{pm}^{i} = \begin{cases} 1 & \text{if part } p \text{ needs processing on machine } m \text{ in process plan } i \\ 0 & \text{otherwise} \end{cases}$$

s_{pq}^{ij} = similarity coefficient between parts p and q when part p uses process plan i and part q uses process plan j ; $\forall p, \forall q, i = 1, 2, \ldots, I_p, j = 1, 2, \ldots, I_q, p \neq q$

No. of search directions	5	7	11	13			17	19		23			29					
No. of objective functions	2–4	2–6	2–10	2	3	4–12	2–16	2–3	4–18	2,13–14, 20–22	8–12	3–7, 15–19	2	3	4–7	8–12, 16–24	13–15	25–28
σ	2	3	7	5	4	6	10	8	14	7	15	17	12	9	16	8	14	18

Table 1. Values of σ for different number of search directions and objective functions (Leung and Wang 2000).

c^i_{pm} = processing cost of part p on machine m with respect to process plan i
t^i_{pm} = processing time of part p on machine m with respect to process plan i
T_m = available time of machine m
A_m = acquisition cost of machine m
U_c = maximum number of parts that can be assigned to cell c
L_c = minimum number of parts to be allocated to cell c.

Let us highlight the difference between the indices k and l. The index l refers to the objective functions and therefore the maximum value of l is identified by the number of objective functions in the mathematical model. Since the proposed model has four objective functions, the index l ranges from 1 to 4. However, the index k refers to the number of fitness functions. Each fitness function is a combination of all objective functions and therefore the maximum value of index k, i.e. K, indicates the number of directions by which the objective space is searched. Thus, K is a control parameter.

4.2. *Mathematical model*

In this section a mathematical model is presented to solve the multi-objective cell formation problem with multiple process plans and independent manufacturing cells. The objectives considered in this model are to: (1) maximize the total similarity between the parts, (2) minimize the total processing cost, (3) minimize the total processing time, and (4) minimize the total investment needed for the acquisition of machines. In order to unify the objectives being optimized, total dissimilarity is minimized instead of maximizing total similarity. To do so, the total similarity function is multiplied by –1. The problem is formulated as:

$$Minimize\ f_1(\mathbf{x}) = -\sum_{p=1}^{P-1}\sum_{q=p+1}^{P}\sum_{i=1}^{I_p}\sum_{j=1}^{I_q}\sum_{c=1}^{C} s^{ij}_{pq} x^i_{pc} x^j_{qc} \tag{2}$$

$$Minimize\ f_2(\mathbf{x}) = \sum_{p=1}^{P}\sum_{c=1}^{C}\sum_{i=1}^{I_p}\sum_{m=1}^{M} D_p c^i_{pm} x^i_{pc} \tag{3}$$

$$Minimize\ f_3(\mathbf{x}) = \sum_{p=1}^{P}\sum_{c=1}^{C}\sum_{i=1}^{I_p}\sum_{m=1}^{M} D_p t^i_{pm} x^i_{pc} \tag{4}$$

$$Minimize\ f_4(\mathbf{x}) = \sum_{c=1}^{C}\sum_{m=1}^{M}\left[\frac{\sum_{p=1}^{P}\sum_{i=1}^{I_p} D_p a^i_{pm} t^i_{pm} x^i_{pc}}{T_m}\right] \times A_m \tag{5}$$

Subject to:

$$\sum_{c=1}^{C}\sum_{i=1}^{I_p} x^i_{pc} = 1 \quad p = 1, 2, \ldots, P \tag{6}$$

$$L_c \le \sum_{p=1}^{P}\sum_{i=1}^{I_p} x^i_{pc} \le U_c \; c = 1, 2, \ldots, C \tag{7}$$

$$x^i_{pc} \in \{0, 1\} \quad p = 1, 2, \ldots, P; \quad c = 1, 2, \ldots, C; \quad i = 1, 2, \ldots, I_p. \tag{8}$$

Objective functions (2), (3), (4) and (5) represent the total dissimilarity, total processing cost, total processing time and total investment in the acquisition of machines, respectively. The symbol $\lceil x \rceil$ indicates the smallest integer value bigger than x. Constraint (6) ensures that each part uses only one process plan and is assigned to only one manufacturing cell. Constraint (7) ensures that the size of cells is not violated.

Choobineh (1988) proposed a similarity coefficient that considers the sequence of operations. This coefficient is used in the proposed model to calculate the similarity between the parts. This similarity is defined as follows:

$$s_{pq}^{ij} = \frac{1}{L'}\left[s_{pq}^{ij}(1) + \sum_{l'=2}^{L'} \frac{C_{pq}^{ij}(l')}{N - l' + 1} \right]; \quad L' < N \tag{9}$$

where s_{pq}^{ij} is the similarity coefficient between the parts p and q when part p uses process plan i and part q uses process plan j, $C_{pq}^{ij}(l')$ is the number of common sequences with length l' between the parts p and q in the related process plans, L' is the length of the longest common sequence between the parts p and q in the related process plans, and N is the number of operations in the shorter process plan, i.e. $N = Min.(N_p^i, N_q^j)$. Here, N_p^i is the number of operations in process plan i of part p and similarly, N_q^j is the number of operations in process plan j of part q. Then $s_{pq}^{ij}(1)$ is computed using the Jaccard similarity coefficient proposed by McAuley (1972).

5. Application of the genetic algorithm

5.1. *Representation of solutions*

For the cell formation problem formulated in section 4.2, each solution is represented by one chromosome with $2 \times P$ genes, i.e. two genes are considered for each part. The first gene contains the cell number to which the corresponding part is assigned. The second gene contains the selected process plan for the corresponding part. It is seen that by this representation only one process plan and one manufacturing cell is selected for each part and therefore constraints (6) and (8) are satisfied automatically. Figure 2 illustrates the proposed representation scheme.

5.2. *Fitness functions*

Since the objective space is explored in multiple directions, say K directions, then K fitness functions are to be defined. For direction k, the fitness function is formed as follows:

$$fit_k(S) = w_{k1}f_1(S) + w_{k2}f_2(S) + w_{k3}f_3(S) + w_{k4}f_4(S) + \alpha_1 \sum_{c=1}^{C} g_c(S) + \alpha_2 \sum_{c=1}^{C} g'_c(S) \tag{10}$$

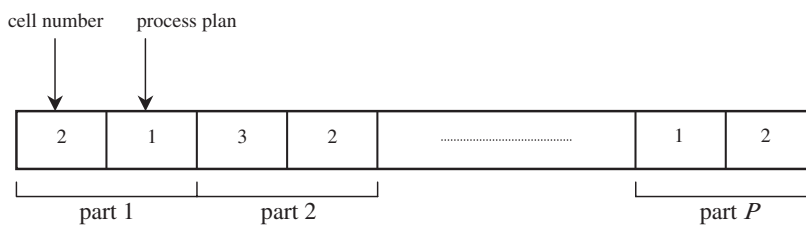

Figure 2. Representation of solutions in the proposed method. In this figure, for example, part 1 is assigned to cell 2 and is processed under process plan 1.

In equation (10), $fit_k(S)$ represents the fitness of chromosome S with respect to the kth search direction, and $f_l(S)$ indicates the value of the lth objective function for chromosome S. The parameters α_1 and α_2 are large positive penalty coefficients for infeasible solutions. The functions $g_c(S)$ and $g'_c(S)$ are defined as follows:

$$g_c(S) = \begin{cases} 0 & \text{if } \sum_{p=1}^{P}\sum_{i=1}^{I_p} x_{pc}^i \le U_c \\ \sum_{p=1}^{P}\sum_{i=1}^{I_p} x_{pc}^i - U_c & \text{otherwise} \end{cases} \quad \forall c = 1, 2, \ldots, C \tag{11}$$

$$g'_c(S) = \begin{cases} 0 & \text{if } \sum_{p=1}^{P}\sum_{i=1}^{I_p} x_{pc}^i \ge L_c \\ L_c - \sum_{p=1}^{P}\sum_{i=1}^{I_p} x_{pc}^i & \text{otherwise} \end{cases} \quad \forall c = 1, 2, \ldots, C \tag{12}$$

As seen in equations (11) and (12), the functions $g_c(S)$ and $g'_c(S)$ are equal to zero for feasible solutions. This implies that the values of coefficients α_1 and α_2 do not affect the fitness value of feasible solutions and only penalize infeasible solutions. In equation (10), since the values of objective functions vary in different ranges, the objective functions with large values may dominate the contribution of other objectives. To alleviate this problem, the fitness function (10) is replaced by equation (13):

$$fit_k(S) = w_{k1}h_1(S) + w_{k2}h_2(S) + w_{k3}h_3(S) + w_{k4}h_4(S) + \alpha_1 \sum_{c=1}^{C} g_c(S) + \alpha_2 \sum_{c=1}^{C} g'_c(S), \tag{13}$$

where

$$h_l(S) = \frac{f_l(S)}{\max\left\{f_l(S') \middle| \forall S' \in \Omega\right\}} \quad l = 1, 2, \ldots, 4. \tag{14}$$

The function $h_l(S)$ denotes the normalized value of the objective function l for chromosome S and Ω stands for the set of all chromosomes under evaluation.

It is preferred to search objective space in uniform directions because uniform directions promise to find uniformly distributed solutions along the Pareto optimal frontier. Thus, the uniform design technique discussed in section 3.2 is applied to form search directions. To do so, objective functions are treated as factors and the number of directions as levels. Hence, search directions are calculated as follows:

$$W = [w_{kl}]_{K\times 4}; \quad w_{kl} = \frac{u_{kl}}{\sum_{l=1}^{4} u_{kl}}; \quad \forall k, l \tag{15}$$

where $U(K, 4) = [u_{kl}]_{K\times 4}$ is the uniform design matrix defined in equation (1). Each row of the matrix W is a search vector and w_{kl} is the weight of the objective function l in fitness function k.

5.3. *Crossover*

The proposed genetic algorithm uses a simple crossover operator in which a random crossover point is determined and the second parts of the chromosomes

are exchanged. The probability of selecting a chromosome for crossover is calculated by

$$p_k(S_r) = \frac{\max\{fit_k(S_n)|n = 1, 2, \ldots, \|\Lambda\|\} - fit_k(S_r)}{\sum_{w=1}^{\|\Lambda\|} \left(\max\{fit_k(S_n)|n = 1, 2, \ldots, \|\Lambda\|\} - fit_k(S_w)\right)} r = 1, 2, \ldots, \|\Lambda\|, \quad (16)$$

where $p_k(S_r)$ is the probability of selecting chromosome S_r for crossover with respect to fitness function k. The crossover operation is done as follows:

Step 1. With respect to fitness function k, $k = 1, 2, \ldots, K$, select size_pop/K chromosomes. Take all the selected chromosomes to set Δ.
Step 2. Randomly mate chromosomes in Δ.
Step 3. For every parent randomly mated in step 2, use the simple crossover operator and replace the created children with relevant parents in Δ.

5.4. *Mutation*

Mutation brings unexpected features to the children that do not exist in parents. Every chromosome in Δ is chosen for mutation with a probability of p_m. In every chromosome selected for mutation, a gene is selected randomly. If the selected gene is odd, the current value of that gene is replaced with a random number selected from $[1, C]$. If the selected gene is even, a process plan of the corresponding part is randomly selected and replaced with the current value of that gene.

5.5. *Evaluation*

In order to select chromosomes for the next generation, all the newly created chromosomes are to be evaluated. Evaluation is done as follows:

Step 1. Set $n = 1$, $k = 1$ and large positive values for α_1 and α_2.
Step 2. Using equations (11)–(15) compute the kth fitness value of the nth chromosome.
Step 3. If $k < K$, set $k = k + 1$ and go to step 2. Otherwise, go to step 4.
Step 4. If $n < \|\Delta\|$, set $n = n + 1$, $k = 1$ and go to step 2. Otherwise, stop.

5.6. *Selection*

In the proposed algorithm, the selection process is done as follows.

Step 1. Add newly created chromosomes to current population, i.e. $\Omega = \Lambda \cup \Delta$. Then set $\Lambda = \Phi$ and $k = 1$.
Step 2. With respect to fitness function k, calculate the following selection probabilities for each chromosome:

$$p_k(S_n) = \frac{\max\{fit_k(S_n)|n = 1, 2, \ldots, \|\Omega\|\} - fit_k(S_n)}{\sum_{n=1}^{\|\Omega\|} \left(\max\{fit_k(S_n)|n = 1, 2, \ldots, \|\Omega\|\} - fit_k(S_n)\right)} n = 1, 2, \ldots, \|\Omega\| \quad (17)$$

where $p_k(S_n)$ is interpreted as the probability of selecting chromosome S_n with respect to fitness function k.

Step 3. Compute cumulative probabilities from step 2. Let $CDF_k(S_n)$ represent the cumulative probability of chromosome S_n with respect to fitness function k. Set $\upsilon = 1$.

Step 4. Generate a random number r from a uniform continuous distribution in $[0, 1]$. Use the following relation to select the nth chromosome for the next generation:

$$CDF_k(S_{n-1}) < r \leq CDF_k(S_n) \qquad n = 1, 2, \ldots, \|\Omega\|. \tag{18}$$

Take chromosome S_n to set Λ.

Step 5. If $\upsilon < \lceil size\_pop/K \rceil$, set $\upsilon = \upsilon + 1$ and go to step 4. Otherwise, go to step 6.

Step 6. If $k < K$, set $k = k + 1$ and go to step 2. Otherwise, stop.

The proposed selection approach selects size_pop/K chromosomes with respect to each fitness function at a time (steps 2–5). Our approach is different from the VEGA approach developed by Schaffer (1985). In VEGA, selection is first done on the basis of one objective followed by selection based on the next objective, and so on. In the proposed approach, however, selection is based on fitness functions rather than objective functions and every time size_pop/K chromosomes are selected where K is the number of fitness functions. In fact, the VEGA approach is a special case of our general selection approach as *identity vectors* are used in VEGA to explore the objective function space. Our selection approach would function like VEGA if we set identity vectors as search directions. One of the major limitations of the VEGA approach is its tendency to cluster solutions around the best solution with respect to each objective (Goldberg 1989). This is due to the fact that the VEGA approach explores the Pareto frontier in identity directions meaning that objectives are considered independently (Srinivas and Deb 1994). In other words, in every search direction only one objective is considered and all other objectives are ignored. Our proposed approach does not use identity vectors and therefore provides the possibility to cluster solutions throughout the Pareto optimal frontier. Application of the uniform design technique increases the possibility of obtaining well-distributed solutions along the Pareto optimal frontier. Therefore, the major limitation of VEGA can be alleviated by the proposed approach. However, comparison of the performance of our selection approach with VEGA as well as other approaches requires independent research as it is beyond the scope of this paper, which proposes a comprehensive multi-objective integer programming model to the design of cellular manufacturing systems.

6. Genetic algorithm for the multi-objective cell formation problem

In this section, all the elements discussed in section 5 are synthesized to evolve a multi-objective genetic algorithm with multiple fitness functions.

Step 1. Set values of parameters K, σ , size_pop, max_gen, and p_m. Set gen = 1.

Step 2. Generate an initial population of size size_pop and call it Λ. Use equations (11)–(15) to calculate the fitness values of all the chromosomes in Λ.

Step 3. Use the crossover algorithm explained in section 5.3 to do the crossover operation.

Step 4. Use the mutation algorithm explained in section 5.4 to do the mutation operation.

Step 5. Use the evaluation algorithm presented in section 5.5 to evaluate the chromosomes created by crossover and mutation operations.

Step 6. Use the selection algorithm explained in section 5.6 to select size_pop chromosomes for the next generation. Keep the fitness values of selected chromosomes for the next generation. The new population is represented by Λ.

Step 7. Add the newly created population to the set of Pareto optimal solutions, i.e. $\Gamma = \Gamma \cup \Lambda$. Remove all the dominated solutions from Γ.

Step 8. If gen $<$ max_gen, set gen $=$ gen $+ 1$ and go to step 3. Otherwise, stop.

Some guidelines are given to set the values of parameters in step 1. According to equation (1), the number of fitness functions, K, is a prime number greater than the number of objective functions. A large number of fitness functions would result in relatively exhaustive exploration of the decision space and more non-dominated solutions, though this would increase the computation time. The size of the population represented by the parameter size_pop can be set with respect to the number of decision variables in the problem. As the number of decision variables increases, a large value for parameter size_pop is required to better explore the solution space. Moreover, the parameter size_pop should be a multiple of K. The parameter max_gen is a simple stopping condition indicating the maximum number of generations to be run. The mutation probability, p_m, is a relatively small value and based on our experience a probability of 0.1 performs well. The value of parameter σ is determined with respect to the number of objective functions and fitness functions as given in table 1. As seen in this table, for four objectives and five fitness functions the parameter σ is set to 2.

It is notable that at the end of each iteration all non-dominated solutions are recorded in set Γ. This guarantees that a good solution obtained in the earlier generations will never be lost during the execution of this algorithm.

7. Illustrative examples

To evaluate the performance of the proposed algorithm four examples are adopted from the literature. Since a majority of cellular manufacturing systems operate with few cells and machines, the selected problems can provide a general perspective of the applicability of the proposed algorithm. A survey of 32 cell users in the USA revealed that these firms had six cells each with six machines, on average (Wemmerlöv and Hyer 1989).

Based on our computational experience the following values are considered for the parameters: $K = 5$, $p_m = 0.1$, size_pop $= 50$, max_gen $= 700$ and $\sigma = 2$.

Example 1

The first example is adopted from Aktürk and Balkose (1996) and includes 20 parts and 10 machines. The proposed algorithm is tested in two different schemes. In the first scheme, only the original data given in Aktürk and Balkose (1996) are used to show the effectiveness of the proposed method. In addition, relevant processing costs have been added to the original data. In this scheme, there is only one process plan for each part. In the second scheme, however, without manipulation of the original data, additional process plans along with related production costs and processing times are superimposed on the original data. Table 2 contains all the original as well as superimposed data. The original data are given by the first process plan of each part. The acquisition costs of machines are 106, 136, 65, 140, 103, 61, 126, 93, 94 and 70 for machines A, B, F, G, H, K, L, M, R and T, respectively.

Part no.	PP	Sequence of operations	Processing costs in terms of sequence number					Processing times in terms of sequence number					Demand
			1	2	3	4	5	1	2	3	4	5	
1	1	B F G	2	5	6			2	3	2			150
	2	F G R	5	3	3			1	3	2			
2	1	A B F L	1	3	2	4		2	3	4	3		226
3	1	F K M R	6	1	4	4		2	3	4	2		335
4	1	F G H R	3	3	2	4		3	2	4	2		446
	2	F G R T	3	3	3	3		3	3	2	2		
5	1	A M T	1	3	4			2	3	4			274
	2	B M T	1	3	4			3	3	4			
6	1	A B L M T	5	1	3	3	2	2	3	2	1	5	171
	2	A B M T	5	1	4	2		2	3	2	5		
7	1	F G R L T	2	2	4	1	2	3	2	3	1	2	218
	2	F G H R T	2	2	1	4	2	3	2	2	3	2	
	3	F H R L T	2	1	4	1	2	3	2	2	1	2	
8	1	K A R T	4	2	1	1		2	3	1	2		273
	2	A K R T	2	4	1	1		3	2	1	2		
9	1	A K M R T	3	2	2	2	5	3	1	4	2	3	307
	2	A K R T	3	2	3	5		3	2	3	3		
10	1	A K R T	3	1	2	4		3	2	2	1		414
	2	K M R T	4	1	2	4		3	2	2	2		
11	1	A B R T	3	3	3	2		2	3	4	1		223
	2	A B M T	3	3	3	2		2	3	5	1		
12	1	B F R L	2	2	1	1		3	2	1	1		378
	2	B F G L	2	2	1	1		3	2	1	1		
13	1	M R T	3	1	3			3	2	4			328
	2	K R T	3	1	3			2	3	4			
14	1	A R T	1	5	2			1	3	2			280
	2	B R T	2	5	2			2	3	2			
15	1	F H R L	2	2	1	1		1	2	1	1		270
	2	F H R T	2	2	1	2		1	2	1	2		
16	1	A B M T	2	2	4	1		2	3	2	3		182
17	1	A B F T	3	2	1	2		3	2	3	4		244
18	1	A B F G	3	1	1	2		5	4	2	2		152
19	1	F G R T	4	1	3	2		4	1	1	1		366
	2	F G R L	4	1	3	1		3	1	1	1		
20	1	G R T	3	2	2			1	1	2			226
	2	F G R	1	3	2			1	1	1			

PP: process plan.

Table 2. Data for example 1.

As mentioned in the original data, the capacity of machines and the maximum number of parts in each cell are assumed to be 3000 and 11, respectively.

Table 3 shows the solutions obtained by the proposed genetic algorithm and the solution reported in Aktürk and Balkose (1996). As shown in table 3, our first and third solutions dominate the solution obtained by Aktürk and Balkose. The solutions obtained by the proposed method provide higher similarity between the parts and less investment in the acquisition of machines.

Table 4 shows the solutions obtained by the proposed genetic algorithm for the second scheme. In table 4, the selected process plan for each part and the required number of each machine are indicated within parentheses. Due to multiple process plans in the second scheme, the proposed genetic algorithm has obtained more

Solution	Cell	Parts	Machines[a]	Total similarity	Total cost ($)	Total time (sec)	Investment ($)
1	I	2, 5, 6, 8, 10, 11, 13, 16, 17	A(2), B(1), F(1), K(1), L(1), M(1), R(1), T(2)	32.09	55439	49975	2080
	II	1, 3, 4, 7, 9, 12, 14, 15, 18, 19, 20	A(1), B(1), F(2), G(1), H(1), K(1), L(1), M(1), R(2), T(1)				
2	I	1, 2, 4, 7, 12, 15, 17, 18, 19, 20	A(1), B(2), F(3), G(1), H(1), L(1), R(1), T(1)	37.01	55439	49975	2290
	II	3, 5, 6, 8, 9, 10, 11, 13, 14, 16	A(2), B(1), F(1), K(1), L(1), M(2), R(2), T(3)				
3	I	1, 4, 7, 12, 15, 17, 18, 19, 20	A(1), B(1), F(2), G(1), H(1), L(1), R(1), T(1)	35.98	55439	49975	2089
	II	2, 3, 5, 6, 8, 9, 10, 11, 13, 14, 16	A(2), B(1), F(1), K(1), L(1), M(2), R(2), T(3)				
Solution of Aktürk and Balkose (1996)	I	1, 4, 7, 9, 10, 12, 13, 15, 18, 19, 20	A(1), B(1), F(2), G(1), H(1), K(1), L(1), M(1), R(2), T(2)	31.57	55439	49975	2150
	II	2, 3, 5, 6, 8, 11, 14, 16, 17	A(2), B(1), F(1), K(1), L(1), M(1), R(1), T(2)				

[a]The values within parentheses in this column indicate the number of associated machine types needed in the corresponding cell

Table 3. Solutions obtained by Aktürk and Balkose (1996) and the proposed algorithm for scheme 1 of example 1.

Solution	Cell	Parts[a]	Machines[b]	Total similarity	Total cost ($)	Total time (sec)	Investment ($)
1	I	4(2), 5(2), 6(2), 8(2), 9(2), 10(1), 11(1), 13(2), 14(1), 16(1), 17(1)	A(2), B(2), F(1), G(1), K(1), M(1), R(2), T(3)	39.108	51477	48328	2300
	II	1(2), 2(1), 3(1), 7(3), 12(2), 15(2), 18(1), 19(2), 20(2)	A(1), B(1), F(2), G(1), H(1), K(1), L(1), M(1), R(1), T(1)				
2	I	5(2), 6(2), 8(2), 9(1), 10(1), 11(1), 13(2), 14(1), 16(1)	A(2), B(1), K(1), M(1), R(2)T(3)	39.812	51784	48942	2258
	II	1(2), 2(1), 3(1), 4(2), 7(3), 12(2), 15(2), 17(1), 18(1), 19(2), 20(2)	A(1), B(1), F(3), G(2), H(1), K(1), L(1), M(1), R(2), T(1)				
3	I	3(1), 4(2), 5(2), 6(2), 8(2), 9(2), 10(1), 11(1), 13(2), 14(1), 16(1)	A(2), B(1), F(1), G(1), K(2), M(1), R(3), T(3)	38.329	51477	48328	2165
	II	1(2), 2(1), 7(3), 12(2), 15(2)17(1), 18(1), 19(2), 20(2)	A(1), B(1), F(2), G(1), H(1), L(1), R(1), T(1)				
4	I	1(2), 2(1), 4(2), 7(3), 11(1), 12(1), 14(1), 15(1), 19(2), 20(2)	A(1), B(1), F(2), G(1), H(1), L(1), R(2), T(1)	35	51207	48058	2197
	II	3(1), 5(2), 6(2), 8(1), 9(2), 10(1), 13(1), 16(1), 17(1), 18(1)	A(2), B(1), F(1), G(1), K(1), M(2), R(2), T(3)				
5	I	1(2), 2(1), 3(1), 4(2), 7(3), 11(1), 12(1), 14(1), 15(1), 19(2), 20(2)	A(1), B(1), F(3), G(1), H(1), K(1), L(1), M(1), R(2), T(1)	35.686	51207	48058	2229
	II	5(2), 6(2), 8(1), 9(2), 10(1), 13(1), 16(1), 17(1), 18(1)	A(2), B(1), F(1), G(1), K(1), M(1), R(1), T(3)				

[a]The numbers within parentheses in this column indicate the process plan selected for the associated part
[b]The values within parentheses in this column indicate the number of associated machine types needed in the corresponding cell.

Table 4. Solutions obtained by the proposed algorithm for the second scheme of example 1.

favourable solutions. In the solutions of the second scheme, the first three objectives including the total similarity, total processing cost, and total processing time have considerably improved, though the total investment is slightly poor.

Example 2

The second example is adopted from Gen and Cheng (2000) and contains seven parts, 13 process plans, and 10 machines. The objective function of this problem is to minimize the sum of the total processing cost and total investment. Since the sequence of operations and processing times are not available in the original problem, these data are superimposed on the original data. Table 5 shows all the data needed for this problem. In this table, each row stands for one process plan and each column represents one machine. The entry $a(c, t)$ in this table indicates that the corresponding part visits the related machine in the ath order and the processing cost and processing time are c and t, respectively. The last column shows the production volume of each part and the last row shows the unit price of each machine.

Our proposed method provided 13 non-dominated solutions along the Pareto optimal frontier. Out of 13 solutions, one solution is the one reported in Gen and Cheng (2000). This solution is the optimum solution considering the sum of total processing cost and total investment in the acquisition of machines as a single objective. The referred solution is given in table 6.

		Machines										
Parts	PP*	1	2	3	4	5	6	7	8	9	10	D_p
1	1	2(2, 3)		1(3, 4)							3(2, 2)	80
2	1					2(2, 5)				1(3, 2)	3(1, 2)	
	2	3(4, 2)		2(5, 1)		4(4, 3)				1(7, 3)		80
	3	2(6, 2)		3(5, 2)		1(7, 2)				4(5, 1)		
3	1	2(6, 2)				3(5, 1)		1(6, 3)				80
	2					2(3, 4)		1(3, 3)				
4	1			2(4, 2)		3(5, 3)					1(7, 2)	
	2		1(4, 3)			2(7, 2)					3(6, 3)	80
	3					1(3, 4)				2(4, 3)	3(3, 2)	
5	1			1(4, 3)	2(2, 2)					3(5, 1)	4(2, 2)	80
6	1	1(5, 1)	2(6, 3)				4(4, 3)		3(8, 2)			80
	2	1(2, 3)	3(4, 2)				2(2, 2)			4(4, 3)		
7	1			1(5, 4)	2(7, 3)					3(3, 2)		80
A_m (\$)		2500	2300	2000	2200	2000	2500	2500	2000	2000	2000	

*PP: process plan.

Table 5. Data for example 2.

Cell	Part no. (process plan)	Machine no. (required no.)
Cell 1	6(2)	1(1), 2(1), 6(1), 9(1)
Cell 2	1(1), 5(1), 7(1)	1(1), 3(1), 4(1), 9(1), 10(1)
Cell 3	2(1), 3(2), 4(3)	5(1), 7(1), 9(1), 10(1)

Table 6. One of the solutions obtained by the proposed method which is the optimum solution considering the sum of total processing and total investment costs as a single objective.

Example 3

The third problem used to evaluate the performance of the proposed method is problem 4 in Gupta *et al.* (1996) with 30 parts and 15 machines. In this problem, each part is processed through one process plan. Since Gupta *et al.* have not considered the sequence of operations, it is assumed here that parts visit machines in increasing order of machine indices. For example, part 2 in this problem needs processing on machines 1, 3, 6, 8, 11 and 14. Hence, we assume the sequence of operations of part 2 as 1-3-6-8-11-14. Processing costs are randomly generated from a uniform distribution in [0, 10]. The production volume of each part and the unit price of each machine have been generated from a discrete uniform distribution in [0, 100]. The processing times of operations have been calculated using the generated production volumes and the workload data given in Gupta *et al.* (1996). As considered by Gupta *et al.*, the capacity of machines, the number of cells, and the minimum number of parts in each cell are assumed to be eight units of time, four cells, and two parts, respectively. The solutions obtained through the proposed algorithm, and also the solution reported in Gupta *et al.* (1996) are shown in table 7. The difference between the solutions obtained by the proposed algorithm is due to the allocation of part 15 as it belongs to cell 2 in the first solution and to cell 1 in the second solution. As seen in table 7, both the solutions obtained by the proposed method dominate the solution obtained by Gupta *et al.* (1996). The solutions obtained by the proposed method provide better similarity between the parts with less investment in the acquisition of machines. Total processing cost and total processing time are the same for both methods as each part has only one process plan.

Example 4

The fourth problem adopted from Aktürk and Turkcan (2000) has 20 parts, 34 process plans and six machines. Aktürk and Turkcan have proposed a heuristic with three stages to form manufacturing cells and determine the layout of machines within the cells. We consider the solution obtained by Aktürk and Turkcan at the end of the second stage in which manufacturing cells are independent and there is no inter-cell movement of parts. Similar to the assumption in Aktürk and Turkcan (2000), the number of manufacturing cells is three and the minimum number of parts in each cell is three. We obtained 27 non-dominated solutions along the Pareto optimal frontier of which one solution is reported here to compare with the solution obtained by Aktürk and Turkcan (2000). Table 8 shows the solution obtained by the proposed method as well as the solution reported in Aktürk and Turkcan (2000). As seen in table 8, the proposed algorithm results in better solution in terms of maximum similarity, minimum processing cost and minimum investment. The total processing time is slightly in favour of the solution obtained by Aktürk and Turkcan (2000).

8. Discussion and conclusion

Since all the complexity in real life cannot be formulated through mathematical models, it is preferred to provide reasonable solutions for the decision-maker to select the most suited one. In this paper, a multi-objective mathematical model is developed to the design of cellular manufacturing systems. However, the formulated mathematical model is still open for adding more objectives like intra-cell load variation, inter-cell load variation, etc. To solve the mathematical model, a multi-objective genetic algorithm with multiple uniform search directions is proposed. The proposed method provides a new approach for multi-objective optimization using

Solution	Cell	Parts	Total similarity	Total cost ($)	Total time (sec)	Investment ($)
Our solution 1	I	1, 4 to 14, 16, 19 to 22, 26, 28, 29, 30	37.0362	28533	967.2	8545
	II	2, 15, 24				
	III	3, 23, 25, 27				
	IV	17, 18				
Our solution 2	I	1, 4 to 16, 19 to 22, 26, 28, 29, 30	38.0053	28533	967.2	8627
	II	2, 24				
	III	3, 23, 25, 27				
	IV	17, 18				
Gupta *et al.* 1996	I	9, 10, 12	36.77	28533	967.2	9063
	II	8, 14, 27				
	III	1 to 7, 11, 13, 15 to 21, 23, 25, 26, 28, 29, 30				
	IV	22, 24				

Table 7. Solutions obtained by Gupta *et al.* (1996) and the proposed algorithm for example 3.

Solution	Cell	Parts	Total similarity	Total cost ($)	Total time (sec)	Investment ($)
Our solution	I	2(1), 8(1), 13(2), 15(1), 16(1), 17(1), 20(1)	24.1167	21778	17592	21316
	II	3(1), 9(1), 10(1)				
	III	1(2), 4(1), 5(1), 6(1), 7(1), 11(2), 12(1), 14(1), 18(2), 19(2)				
Solution of Aktürk and Turkcan-2000	I	1(2), 3(1), 4(1), 10(1), 14(1), 16(1), 17(1), 20(1)	21.8	22045	17434	22802
	II	5(1), 7(2), 9(1), 12(1), 15(1)				
	III	2(1), 6(1), 8(1), 11(2), 13(2), 18(2), 19(2)				

Table 8. Solutions obtained by Aktürk and Turkcan (2000) and the proposed algorithm for example 4.

genetic algorithms, which is used in the design of cellular manufacturing systems in this paper. The novelty of the proposed method is claimed as follows: (1) using a systematic method based on a uniform design technique to set the weights of objective functions, (2) using multiple fitness functions instead of a single fitness function, and (3) proposing a comprehensive model to the design of independent manufacturing cells. The first feature brings a facility to define search vectors in equal distances with each other. In previous attempts, search directions are established either by the perception of decision-makers or generated randomly. The second feature generates several solutions along the Pareto optimal frontier thus making the proposed algorithm suitable for the development of a decision support system for the cell formation problem. The third feature relates to the comprehensiveness and independence of cells in the proposed approach. The proposed mathematical model is relatively comprehensive as it considers machine requirements, processing costs, processing times, sequence of operations, investment in the acquisition of machines, multiple process plans, production volumes, capacity of machines, etc. On the other hand, the independence of manufacturing cells facilitates production planning, scheduling and material handling within the manufacturing system, which are the main goals in adopting cellular manufacturing. In addition, this independence significantly simplifies implementation of modern manufacturing technologies such as CAD/CAM, CIM, JIT, etc. Since a large number of companies operating in the CMS environment operate with independent cells (Wemmerlöv and Hyer 1989), the proposed algorithm can provide effective design of the manufacturing system for these industries.

The proposed method is tested on different problems selected from the literature. The solutions obtained by the proposed method provide improvements compared to the approaches proposed by others in the adopted problems. The comprehensiveness of the mathematical model and the effectiveness of the proposed multi-objective genetic algorithm suit the attempted research for the design of cellular manufacturing systems.

Acknowledgements

The authors would like to put on record their appreciation to the anonymous referee for valuable suggestions, which have enhanced the quality of the paper over its earlier version.

References

Aktürk, M. S. and Balkose, H. O., 1996, Part-machine grouping using a multi-objective cluster analysis. *International Journal of Production Research*, **34**(8), 2299–2315.

Aktürk, M. S. and Turkcan, A., 2000, Cellular manufacturing system design using a holonistic approach. *International Journal of Production Research*, **38**(10), 2327–2347.

Al-Sultan, K. S. and Fedjki, C. A., 1997, A genetic algorithm for the part family formation problem. *Production Planning and Control*, **8**(8), 788–796.

Ballakur, A. and Steudel, H. J., 1987, A within-cell utilization based heuristic for designing cellular manufacturing systems. *International Journal of Production Research*, **25**(5), 639–665.

Choobineh, F., 1988, A framework for the design of cellular manufacturing systems. *International Journal of Production Research*, **26**(7), 1161–1172.

Coello, C. A. C., 1999, A comprehensive survey of evolutionary-based multi-objective optimization. *Knowledge and Information Systems*, **1**(3), 269–308.

Dimopoulos, C. and Zalzala A. M. S., 2000, Recent developments in evolutionary computation for manufacturing optimization: problems, solutions and comparisons. *IEEE Transactions on Evolutionary Computation*, **4**(2), 93–113.

Fonseca, C. M. and Fleming, P. J., 1995, An overview of evolutionary algorithms in multiobjective optimization. *Journal of Evolutionary Computatio*n, **3**(1), 1–16.

Gen, M. and Cheng, R., 2000, Manufacturing cell design. In *Genetic Algorithms and Engineering Optimization* (New York: John Wiley), pp. 390–450.

Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learnin*g (Reading, MA: Addison-Wesley).

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*, **109**, 286–298.

Gupta, Y., Gupta, M., Kumar, A. and Sundaram, C., 1995, Minimizing total intercell and intracell moves in cellular manufacturing: a genetic algorithm approach. *International Journal of Computer Integrated Manufacturing*, **8**(2), 92–101.

Gupta, Y., Gupta, M., Kumar, A. and Sundaram, C., 1996, A genetic algorithm-based approach to cell composition and layout design problems. *International Journal of Production Research*, **34**(2), 447–482.

Hajela, P. and Lin, C.-Y., 1992, Genetic search strategies in multi-criterion optimal design. *Structural Optimizatio*n, **4**, 99–107.

Heragu, S. S., 1994, Group technology and cellular manufacturing, *IEEE Transactions on Systems, Man and Cybernetics*, **24**(2), 203–214.

Horn, J. and Nafpliotis, N., 1993, Multi-objective optimization using the niched Pareto genetic algorithm. IlliGAL technical report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Illinois.

Kusiak, A., 1990, *Intelligent Manufacturing Systems* (Englewood Cliffs, NJ: Prentice Hall).

Lee-Post, A., 2000, Part family identification using a simple genetic algorithm. *International Journal of Production Research*, **38**(4), 793–810.

Leung, Y. W. and Wang, Y., 2000, Multi-objective programming using uniform design and genetic algorithm. *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews*, **30**(3), 293–304.

Mansouri, S. A., Moattar Husseini, S. M. and Newman, S. T., 2000, A review of the modern approaches to multi-criteria cell design. *International Journal of Production Research*, **38**(5), 1201–1218.

McAuley, J., 1972, Machine grouping for efficient production. *Production Engineer*, **51**(2), 53–57.

Moon, C. and Kim, J., 1999, Genetic algorithm for maximizing the parts flow within cells in manufacturing cell design. *Computers & Industrial Engineering*, **36**, 379–389.

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

Schaffer, J. D., 1985, Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the International Conference on Genetic Algorithms and their Applications*, Pittsburgh, PA, pp. 93–100.

Selim, H. M., Askin, R. G. and Vakharia A. J., 1998, Cell formation in group technology: review, evaluation and directions for future research. *Computers & Industrial Engineering*, **34**(1), 3–20.

Shankar, R. and Vrat, P., 1998, Post design modeling for cellular manufacturing system with cost uncertainty. *International Journal of Production Economics*, **55**(1), 97–109.

Shankar, R. and Vrat, P., 1999, Some design issues in cellular manufacturing using the fuzzy programming approach. *International Journal of Production Research*, **37**(11), 2545–2563.

Singh, N., 1993, Design of cellular manufacturing systems: an invited review. *European Journal of Operational Research*, **69**(3), 284–291.

Singh, N. and Rajamani, D., 1996, *Cellular Manufacturing Systems Design, Planning and Control* (London: Chapman & Hall).

Soleymanpour, M., Vrat, P. and Shankar, R., 2002, A transiently chaotic neural network approach to the design of cellular manufacturing. *International Journal of Production Research*, **40**(10), 2225–2244.

Srinivas, N. and Deb, K., 1994, Multi-objective optimization using non-dominated sorting in genetic algorithms. *Journal of Evolutionary Computatio*n, **2**(3), 221–248.

Su, C.-T. and Hsu, C.-M., 1998, Multi-objective machine-part cell formation through parallel simulated annealing. *International Journal of Production Research*, **36**, 2185–2207.

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

Venugopal, V. and Narendran, T. T., 1992, A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers & Industrial Engineering*, **22**(4), 469–480.

Wemmerlöv, U. and Hyer, N. L., 1989, Cellular manufacturing in the U.S. industry: A survey of users. *International Journal of Production Research*, **27**(9), 1511–1530.

Zhao, C. and Wu, Z., 2000, A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives. *International Journal of Production Research*, **38**(2), 385–395.

Zitzler, E., Deb, K. and Thiele, L., 2000, Comparison of multi-objective evolutionary algorithms: Empirical results. *Journal of Evolutionary Computation*, **8**(2), 173–195.