MULTIPLE OBJECTIVE BASED MACHINE-PART CELL DESIGN
CONSIDERING ORDINAL AND RATIO DATA THROUGH NSGA II

Iti Dixit¹, Saurabh Jain², Kapil Kumar Goyal³*
¹M M University, Mullana(Ambala), 133207, Email: er.itidixit@gmail.com
²College of Engineering Roorkee, Roorkee, 247667, Email: sjain.me@gmail.com
³* M M University, Mullana(Ambala), 133207, Email: kapilacad@gmail.com

Abstract
Cellular Manufacturing (CM) is an approach to harness the benefits of high production rate of a flow shop while maintaining flexibility, and utilizing facilities, of a job-shop. CM necessitates that parts and machines are allocated into cells to produce the identified part families so that productivity and flexibility of the system can be improved. In this paper, an attempt has been made to propose a clustering methodology based on Non-dominated Sorting Genetic Algorithm II (NSGA II) in which multiple objectives i.e. intercellular movements and within cell load variations are considered to generate the pareto front. The processing data like operation sequence, machine capacity, processing time and batch sizes have been considered to form the realistic generalized cells. The results support the better performance of the proposed algorithm. The novelty of this study lies in the simple and efficient methodology to produce quick solutions with least computational efforts.

Keywords: Cellular Manufacturing, Optimization, NSGA-II, Ratio Level and Ordinal Level Data

1 Introduction

Group technology (GT) is considered as one of the key issues in successful implementation of flexible manufacturing system (FMS). Cellular manufacturing system (CMS) is an application of Group technology (GT). A Cellular manufacturing system can produce medium volume/medium variety parts more economically than other manufacturing systems. The objective of CMS is to group the parts that have similar design or manufacturing characteristic into part families and corresponding dissimilar machines into machine cells, in order to exploit the cost effectiveness of mass production and flexibility of job shop manufacturing. It is the approach to decompose the whole production system into several mutually separable sub-systems as to reduce setup time, reduction in throughput time, work-in-process inventories, improved quality, increment in flexibility, lower scrap rate etc. (Heragu 1994; Wemmerlov and Hyer 1989). These advantages can only be visualized when the machines and parts are grouped considering manufacturing traits.

During last decade, most of the research reflects production flow analysis (PFA) or binary approach. Binary approach ignore many manufacturing factors, it consider that all operation are of equal importance irrespective of their processing time and processing sequence. It also ignores the capacity of machines and production volume of parts. The process of clustering machine into cells and part into part families using BMPM leads to inferior manufacturing plan which could not able to cope up with real manufacturing data (Nair and Narendran 1999). For considering the ratio level data binary matrix values are replaced by processing time of a part. Similarly ordinal data, consider processing sequence of parts in appropriate sequence number to be captured in incidence matrix. It can take any values on an ordinal scale and hence called ordinal-level data (Geroge et al. 2003).

Different methodologies such as classification and coding systems, similarity coefficient-based 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 to solve the cell formation problem. A review of different methods developed for the CFP can be found in Singh (1993), Selim et al. (1998) and Chattopadhyay et al. (2013).

In the recent past few studies have been carried out to develop efficient evolutionary algorithms to address binary part machine grouping problems (Paydar and Saidi-Mehrabad 2013; Kao and Chen 2013). But not giving due consideration to the manufacturing traits like ordinal and ratio level data results in the hindrance in the practical applicability. Venugopal and Narendran (1992) was the first to solve a cell-formation problem using an evolutionary computation algorithm with multi-objective function i.e. cell load variation and intercell movement through composite score. Gravel et al. (1998) proposed EMO algorithm for the solution of
multiple objective cell-formation problem, i.e. minimization of intercell moves and the minimization within-cell load variation trough weighted sum optimization. Zhao and Wu (2000) attempted a cell-formation problem through weighted sum approach that specifically considered real manufacturing data like sequencing of operations, machine workloads and alternative process route. Sarac and Ozcelik (2012) optimized the genetic algorithm parameters for cell formation through ANOVA analysis. Moon and Kim (1999) discussed a cell design considering attributes such as production volume, cell size, the capacity of the material handling device, etc. A genetic algorithm is applied to maximize the total number of parts flowing between the machines within the same cell.

In the present work ordinal and ratio level data have been considered and an approach to locate the true pareto frontiers has been developed. The paper is organized as follows. In section 2, the CFP is formulated as a multi-objective non-linear integer programming model. The elements of the proposed multiple objective NSGA-II algorithm are developed in section 3. In section 4, the performance of the proposed algorithm is evaluated through solving a problem selected from the literature. Section 5 includes discussion and conclusions.

2. Mathematical Formulation

In this section a mathematical model is presented to solve the multiple objective cell formation problem with ordinal and ratio data. The objectives considered in this model are to: (1) minimize the cell load variation and, (2) minimize the intercellular movement for the acquisition of machines. The problem is formulated as:

2.1 Assumptions

- The operating times for all part type operations on different machine types are known.
- The product mix and demand for each part type in each period is known.
- The capabilities and capacity of each machine type are known and constant over time.
- Each machine type can perform one operation. Likewise, each operation can be done on one machine type only.
- No inventories are considered.
- Set up times are not considered.
- Machine breakdowns are not considered.
- Machines are available at the start of the period (zero installation time).

2.2 Indices

\( g = \) cell index \( 1, 2, 3, \ldots, G \)
\( i = \) part index \( 1, 2, 3, \ldots, I \)
\( k = \) machine index \( 1, 2, 3, \ldots, K \)
\( r = \) operation sequence of parts \( 1, 2, 3, \ldots, R_i \)

2.3 Parameters

\( w_{ki} \rightarrow \) workload on machine \( k \) induced by part \( i \).
\( t_{ki} \rightarrow \) processing time (seconds /part) of part \( i \) on machine \( k \).
\( T_k \rightarrow \) available time on machine \( k \) in a given period of time.
\( N_i \rightarrow \) production requirement of part \( i \) in a given period of time.
\( e_{ik} \rightarrow \) transpose of \( W_{ki} \) matrix where time replaced by 1.
\( x_{kg} \rightarrow \) shows membership of \( k^{th} \) machine in \( g \) cell.
\( y_{ig} \rightarrow \) shows membership of \( i^{th} \) part in \( g \) cell.
\( M \rightarrow \) average cell load matrix.
\( T_k \rightarrow \) capacity of each machine of type \( k \).

\[ w_{ki} = \frac{t_{ki} \times N_i}{T_k} \]

\[ m_{ig} = \frac{\text{Total load of cell } g \text{ induced by part } i}{\text{Number of machines in cell } g} = \frac{\sum_{k=1}^{K} x_{kg} \times w_{ki}}{\sum_{k=1}^{K} x_{kg}} \]

2.4 Decision variable

\[ x_{kg} = \begin{cases} 1 & \text{if machine } k \text{ is in cell } g \\ 0 & \text{otherwise} \end{cases} \]

\[ y_{ig} = \begin{cases} 1 & \text{if } \sum_{k=1}^{K} e_{ik} \times x_{kg} > 0 \\ 0 & \text{otherwise} \end{cases} \]

\[ e_{ik} = \begin{cases} 1 & \text{if } t_{ki} > 0 \\ 0 & \text{otherwise} \end{cases} \]

\[ x_{ir} = \begin{cases} 1 & \text{if operations } r, r + 1 \text{ are performed in the same cell} \\ 0 & \text{otherwise} \end{cases} \]

2.5 Objective function

Minimize \( O_1 = \sum_{k=1}^{K} \sum_{g=1}^{G} \sum_{i=1}^{I} (w_{ki} - m_{ig})^2 \)

Minimize \( O_2 = \sum_{i=1}^{I} N_i \left[ \sum_{r=1}^{R_i} \sum_{i=1}^{I} (x_{ir}) \right] \)

Subject to:
The objective functions (1) and (2) represent the cell load variation and intercellular movements. Constraints (3) and (4) ensure that each machine and part can only be assigned to only one cell. Constraints (5) and (6) assure that each cell should be accompanied by at least two machines and parts and the (7) constraint ensures that no cell is empty.

3 Cell Formation Through Non-dominated Sorting Genetic Algorithm-II

The Non-dominated Sorting Genetic Algorithm (NSGA-II) was proposed by Deb et al., (2002) and is based on several layers of classifications for the individuals. In the present study, there are two objectives i.e. cell load variation and intercellular movement. Therefore, to locate the true pareto frontiers efficiently, NSGA-II has been implemented. Figure 1 presents the flow chart of NSGA-II. In the first step of algorithm, an initial population is generated, randomly. For each generation, the operators like selection, crossover and mutation are performed to create the offspring chromosomes, which are further evaluated along with the old population in various levels of fronts. Further in each front all the solutions are further ranked through the crowding distance. Afterwards, the best solutions, in the first front according to crowding distance fitness are considered for the next population and then next front and so on. When at the end the whole solutions from a front cannot be accommodated in the population then best solutions according to crowding distance are selected for the next population to complete. The algorithm is stopped if the stopping criterion is satisfied. Since a stochastic remainder proportionate selection was used in this approach, individuals in the first front have the maximum fitness value and they always get more copies then the rest of the population. This allows searching for non-dominated regions and results in quick convergence of the population towards such regions.

\[ \sum_{g=1}^{G} x_{kg} = 1 \text{ for } k = 1, 2, \ldots, K \]  
(3)

\[ \sum_{g=1}^{G} y_{ig} = 1 \text{ for } i = 1, 2, \ldots, I \]  
(4)

\[ \sum_{k=1}^{K} x_{kg} \geq 2 \text{ for } g = 1, 2, \ldots, G \]  
(5)

\[ \sum_{k=1}^{K} y_{ig} \geq 2 \text{ for } g = 1, 2, \ldots, G \]  
(6)

\[ \sum_{k=1}^{K} x_{kg} = 1 \]  
(7)

The main components of NSGA-II for implementation are as follows.

1. Chromosomes representation
2. Initial population
3. Fitness function
4. Selection
5. Crossover
6. Mutation

3.1 Chromosome Representation

A chromosome is an encoding schema of a feasible solution of the problem at hand. Specifically, for the cell formation problem, a chromosome represents a possible partition of the set of machines into a number of groups called cells. A coding scheme used in the present work is a string of integer numbers. In this string the position occupied by the number defines the machine number and the value of that number corresponds to the cell number. For example, the string \((1 2 2 3 1 1 3 1 3)\), containing 9 machines, may represent a possible solution with three cells. Machines 1, 5, 6 and 8 compose Cell 1, identified by the digit 1. Machines 2 and 3 form Cell 2 and, finally, Cell 3 has the remaining machines, 4, 7 and 9.

3.2 Selection

Roulette wheel selection procedure, proposed by Goldberg (1989). Under this scheme, individuals are selected for reproduction to generate new strings according to their objective function values. Therefore a string with a high fitness value has more chances to be
selected as one of parents than a string with a low fitness value. It is a scheme that is often used as a selection operator is implemented by finding the probability of each string to be selected.

If \( N \) be the number of strings in each population (i.e. population size), strings are denoted by \( \{x_1, x_2, x_3, \ldots, x_n\} \) and \( f(x_i) \) is the fitness of the \( x_i \) string.

In the roulette wheel selection scheme, the selection probability, \( p(x_i) \) of solution \( x_i \) is given as:

\[
p(x_i) = \frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}
\]

### 3.3 Crossover

Crossover enables the algorithm to extract the best genes from different individual and recombine into potentially superior child. Two chromosomes are selected randomly from mating pool and a portion of the individual is exchanged between the two chromosomes. The resulting chromosomes are the offspring replace the parents for next generation. For example, the crossover operator applied to the following two strings assuming the crossover site is 4.

| String1  | 3 1 2 1 3 2 2 1 |
| String2  | 1 1 2 3 | 2 3 1 1 |

After crossover the above two strings are changed in the following manner:

| String1  | 3 1 2 1 2 3 1 1 |
| String2  | 1 1 2 3 3 2 2 1 |

### 3.4 Mutation

To maintain diversity in the population, mutation operators are applied on the strings generated by crossover. The mutation operator changes a 1 to a 0 and vice versa in case of a binary string, with some mutation probability. It can be viewed as a transition from a current solution to its neighborhood solution in local search algorithms. In applications, mutation operators are not as complicated as a crossover. However, mutation operators are different for binary and permutation coded strings. For example, when the string is mutat-ed between the sites 3 and 6, we have

| String1  | 2 1 2 3 3 2 2 1 |
| String1  | 2 1 1 3 3 3 2 1 |

### 4 Case Study

To investigate and test the proposed methodology, the data set were obtained from Nair and Narendran (1999). The illustration includes 10 machine 12 parts data set with ordinal level and ratio level data. Table 1, presents the ratio level data i.e. the workload of various parts on the respective machines. While the Table 2 represents the ordinal data i.e. the operation sequences of parts i.e. precedence constraints to be followed along with the production volume or batch size of parts.

#### Table 1: Ratio level data in machine part matrix

<table>
<thead>
<tr>
<th>Parts</th>
<th>1 2 3 4 5 6 7 8 9 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.96 0.95 0.63</td>
</tr>
<tr>
<td>2</td>
<td>0.86 0.54 0.04 0.67</td>
</tr>
<tr>
<td>3</td>
<td>0.63 0.07 0.89 0.61 0.72</td>
</tr>
<tr>
<td>4</td>
<td>0.07 0.83 0.72</td>
</tr>
<tr>
<td>5</td>
<td>0.73 0.83 0.49 0.72</td>
</tr>
<tr>
<td>6</td>
<td>0.61 0.83 0.49 0.72</td>
</tr>
<tr>
<td>7</td>
<td>1.20 1.00 0.81 0.77</td>
</tr>
<tr>
<td>8</td>
<td>0.54 0.92 0.72 0.72</td>
</tr>
<tr>
<td>9</td>
<td>0.39 0.61 0.72 0.72</td>
</tr>
<tr>
<td>10</td>
<td>0.70 0.72 0.71 0.02</td>
</tr>
</tbody>
</table>

#### Table 2: Ordinal data (processing sequence) of parts

<table>
<thead>
<tr>
<th>Parts</th>
<th>Processing sequence</th>
<th>Production Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3-4-6</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>2-10-5-8</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>8-2-10-5</td>
<td>120</td>
</tr>
<tr>
<td>4</td>
<td>4-9-7</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>6-8-1-4-3</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>4-9-7</td>
<td>90</td>
</tr>
<tr>
<td>7</td>
<td>2-10-5</td>
<td>150</td>
</tr>
<tr>
<td>8</td>
<td>2-10</td>
<td>75</td>
</tr>
<tr>
<td>9</td>
<td>3-6-8-1</td>
<td>45</td>
</tr>
<tr>
<td>10</td>
<td>3-6-1</td>
<td>80</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>25</td>
</tr>
<tr>
<td>12</td>
<td>8-5-2</td>
<td>110</td>
</tr>
</tbody>
</table>

#### Table 3: NSGA-II Parameter for case study

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Crossover Probability</td>
<td>0.8</td>
</tr>
<tr>
<td>Mutation Probability</td>
<td>0.03</td>
</tr>
<tr>
<td>Number of generations</td>
<td>50</td>
</tr>
</tbody>
</table>
5 Result and Discussion

In this study, an efficient algorithm is proposed based on NSGA-II for multi-objective cell formation problem. The algorithm is coded in Matlab 7.0 and run on a Dell Pentium V PC with 2.4 GHz Processor. The algorithm also takes care of avoiding cells with singleton part family that may be encountered at times. The number of possible solutions for this problem can be found by calculating the number of ways we can assign 10 machines to 3 cells so that no cell can have less than two machines. The set of parameters adopted in the case study are tabulated in Table 3. After running NSGA-II with objectives (cell load variation and intercell movement) only three non-dominated solutions, were found. Figure 2 shows the following Pareto front of the solution.

![Figure 2: Pareto Front for the case study](image)

The detailed result for the above three alternative solution on the pareto front are given in Table 4.

<table>
<thead>
<tr>
<th>Sol. No.</th>
<th>Cell-1</th>
<th>Cell-2</th>
<th>Cell-3</th>
<th>Cell Load Variation ($O_1$)</th>
<th>Intercell Movement ($O_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,3,4,6,8</td>
<td>2,5,10</td>
<td>7,9</td>
<td>3.50</td>
<td>455</td>
</tr>
<tr>
<td></td>
<td>(1, 5, 9, 10)</td>
<td>(2,3,7,8,12)</td>
<td>(4,6,11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2,5,10</td>
<td>1,3,6,8</td>
<td>4,7,9</td>
<td>4.55</td>
<td>409</td>
</tr>
<tr>
<td></td>
<td>(2,3,7,8,12)</td>
<td>(1,5,9,10)</td>
<td>(4,6,11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2,5,8,10</td>
<td>1,3,6</td>
<td>4,7,9</td>
<td>6.98</td>
<td>274</td>
</tr>
<tr>
<td></td>
<td>(2,3,7,8,12)</td>
<td>(1,5,9,10)</td>
<td>(4,6,11)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Bold values represent the machine clusters and values in braces represent part families

Table 4 illustrates the objective values of the non dominated solutions that were evolved using NSGA-II. The total number of non-dominated solutions evolved during all experimental runs was three. Table 4 also presents the design of cells. Each solution proposes three manufacturing cells having an optimal group of machines and parts. This is worth to mention here that even in this illustrative problem authors were able to locate three pareto frontiers giving the policy maker a flexibility to trade off between uniform flow and uniformly loaded cells.

6 Conclusions and Future Scope

In the present study, application of NSGA-II, to a cell formation problem considering the ordinal and ratio level data has been investigated. The cellular design multiple objective problem is formulated as a bi-objective problem. The objectives considered are minimization of cell load variation and total intercellular movements. The proposed clustering algorithm has been explained with numerical illustrations relating to workload (ratio level) data and sequence data (sequence data) with real manufacturing attributes like production volume, machine capacity, processing sequence, processing time of each part and machine capacity. Implementation of NSGA-II provide the decision maker with a considerable number of alternative non-dominated solutions (Pareto-front), now it depends upon the decision maker to make a trade-off between two conflicting objectives. The NSGA-II based algorithm can be suitably modified and employed to solve the cell formation problem with other non binary real value data like machine duplication, layout considerations and material handling systems enhancing it to a more generalized manufacturing environment.

Reference


