A Tabu Search strategy to solve cell formation problem with ratio level data

R. Kamalakannan*
Department of Mechanical Engineering,
Vickram College of Engineering, Madurai,
Tamil Nadu-630 561, India
Email: kamalbritto@yahoo.co.in
*Corresponding author

R. Sudhakara Pandian
Department of Mechanical Engineering,
SACS MAVMM Engineering College, Madurai
Tamil Nadu-625 301, India
Email: sudhame@gmail.com

Abstract: This paper concentrates on the cell formation problem for the ratio level data to the design of cellular manufacturing system. The aim of this paper is to identify the machine cells and part family and as a result to create production cells in order to reduce the cell load variation. A competent Tabu Search (TS) algorithm is proposed to investigate the search space of all possible solutions with a chain of moves. This method is an iterative process for seeking a global optimum for the discrete combinatorial optimisation problems. The ratio level data is calculated in terms of time in seconds based on the data collected from the processing time of the part, production volume of the part and availability of the machine. A modified grouping efficiency (MGE) is used to compute the cell formation problem. The results clearly indicated that this proposed TS yield good results compared to the chosen benchmark problems.

Keywords: Tabu Search algorithm; cell formation problem; ratio level data; modified grouping efficiency; MGE.

Reference to this paper should be made as follows: Kamalakannan, R. and Pandian, R.S. (2018) ‘A Tabu Search strategy to solve cell formation problem with ratio level data’, Int. J. Business Intelligence and Data Mining, Vol. 13, Nos. 1/2/3, pp.209–220.

Biographical notes: R. Kamalakannan belongs to the Faculty of Mechanical Engineering Department at Vickram College of Engineering. He holds a Bachelor’s degree in Mechanical Engineering, a Master’s degree in Production Engineering and a Management degree in Production. He is currently pursuing his doctoral research in the area of combinatorial optimisation problems. His research interests include multi-objective optimisation and cell formation problems. He has published several papers in national and international conferences and journals.

R. Sudhakara Pandian received his BE degree in Mechanical Engineering and ME degree in Industrial Engineering from Thiagarajar College of Engineering, Madurai. He received his PhD in Operations Management from National
210  R. Kamalakannan and R.S. Pandian

Institute of Technology, Rourkela. He has published more than 20 papers in conferences and journals. He has also published an edited book entitled Operations Management Research and Cellular Manufacturing Systems: Innovative Methods and Researches. His research interests include operations management and optimisation techniques.

This paper is a revised and expanded version of a paper entitled ‘A Tabu Search strategy to solve cell formation problem with ratio level data’, presented at International Conference on Inventive Computation Technologies (ICICT 2016), Coimbatore, 27 August 2016.

1 Introduction

Recent manufacturing systems is under pressure by the unpredictability in demand and the ever diminishing product life cycles and are finding it rigid to cope with these challenges. Growing competency in the manufacturing industry and customer requirement has been insisting product designers to come up with better alternatives. Due to these confront, companies have to give different types of products to meet the customer requirements. As customers have variety of alternatives at different prices and qualities, this situation outcomes in fall of average customer demand per company, which means low utilisation due to low to moderate volumes of demand. Product variety is an expected feature for a company’s endurance. Though, each additional variety of product design usually results in more investment in equipment and machines, more complexity on manufacturing system control and reduction in manufacturing system flexibility. To overwhelm these challenges caused by the product variety and meet varying volumes of demand, there must be a flexible and efficient connection between design and manufacturing. Group Technology is one of the best manufacturing philosophies for this environment. Cellular manufacturing (CM) is an application of the group technology philosophy to manufacturing and provides flexibility in meeting low to moderate demand and dealing with high product variety. This leads to CM as a potential substitute which gives some abrupt benefits of reduction in the costs related to work in process, material handling and setup times. As a result cellular manufacturing system (CMS) is relatively well equipped to face the challenges mentioned above.

In cell formation problem, usually binary machine part incidence matrix, which is built from route sheet information, has been used as input (Kashan et al., 2014). Later, investigators begins to make use of other information or production factors like workload on the machines, operational sequence of the parts, batch size of the parts, machine capacity, etc., that are available in the shop floor. The process of clustering machines into machine cells and parts into part families without using such information may lead to lower manufacturing plans (Mahapatra and Sudhakarapandian, 2008). Hence, the necessary arises to use non-binary data for obtaining groups or clusters of machines and parts. In this research work, the real life production factors like operational time is considered to make cell formation. The workload information is commonly considered as ratio level data in cell formation problem and a modified incidence matrix is formed with this data. The total processing time of a part is computed by multiplying the production quantity of the part with its unit processing time. The workload (or ratio) value replaces
‘1’s in the incidence matrix. The resultant workload values will take any value in the ratio scale, and they represent the ratio level data (George et al., 2003).

**Figure 1** (a) Machine part occurrence matrix for ratio level data (in seconds) (b) Block diagonalised matrix for ratio level data (in seconds)

Tabu Search is assumed to be one of the most successful meta-heuristic techniques for the NP complete applications. Onwubolu and Songore (2000) addressed the cell formation problem with three objective functions such as minimising cell load variation, minimising intercell moves and combining both the previous objectives and designed a Tabu Search method which offers freedom to consider maximum cell size and number of machines within cell and they reported encouraging results. Logendran and Karim (2003) proposed long-term memory based on minimal frequency to solve Cell formation problem, and a Tabu Search approach was developed to improve solutions which was initially developed followed by six different versions of it in order to investigate the impact of long term memory and the use of fixed versus variable tabu list sizes. All approaches outperformed the mixed-integer programming model obtaining solutions which are close to optimal in no significant amount of time.

Foulds et al. (2006) introduced mixed integer programming model combined with assignment of parts to individual machines, the grouping of individual machines into cells, and the modification of individual machines to increase their part processing capability, called sustainable cell formulation problem heuristic and solved this class of problems with Tabu Search with much better result. Lei and Wu (2006) worked with multi-objective cell formation and proposed a Pareto-optimality based on multi-objective Tabu Search with different objectives: minimisation of the weighted sum of intercell and intracell moves and minimisation of the total cell load variation. Chung et al. (2011) proposed an efficient Tabu Search algorithm based on a similarity coefficient to solve the cell formation problem with alternative process routings and machine reliability considerations. In the proposed algorithm, good initial solutions are first generated and later on improved by a Tabu Search algorithm combining the mutation operator and an effective neighbourhood solution searching mechanism. When compared with the
mathematical programming approach which took three hours to solve problems, the proposed algorithm is able to produce optimal solutions in less than 2 sec.

Solimanpur and Elimi (2013) developed a mixed-integer linear programming model for the attempted cell scheduling problem and a nested application of Tabu Search approach is investigated in this paper to solve the problem heuristically. The effectiveness of the proposed nested Tabu Search (NTS) algorithm is evaluated on 16 problems selected from the literature. Comparison of the results of NTS with SVS-algorithm reveals the effectiveness and efficiency of the proposed algorithm.

2 Mathematical model formulation

Let as assume that the

\[ M \] number of machines (for \( i = 1, 2, 3, \ldots, M \))
\[ P \] number of parts (for \( j = 1, 2, 3, \ldots, P \))
\[ K \] number of cells (for \( k = 1, 2, 3, \ldots, K \))
\[ X_{ik} \] machine cell \((m \times k)\) membership matrix
\[ W_{ij} \] machine part \((m \times p)\) matrix in terms of workload on machine \(i\) induced by part \(j\)
\[ L_{kj} \] cell part \((k \times p)\) matrix of average cell load.

In this section a model is shown to solve the cell formation problem with the ratio level data. The objective considered in this model is to reduce the cell load variation (Venugopal and Narendran, 1992). The visit of the components to the machines has been represented in form of their work load as shown in Table 1 for the computation of cell load variation.

Let us assume there are ‘\(k\)’ cells \((k = 1, 2, 3 \ldots K)\).

Each machine \(i\) should belongs to one machine cell \(k\) as given in (3).

\[
X_{ik} = \begin{cases} 
1 & \text{if Machine } i \text{ belongs to cell } k \\
0 & \text{otherwise}
\end{cases}
\] (1)

The cell load in cell \(k\) from part \(j\) is given by

\[
L_{kj} = \frac{\sum_{i=1}^{M} X_{ik} W_{ij}}{\sum_{i=1}^{M} X_{ik}}
\] (2)

where

\[
\sum_{i=1}^{M} X_{ik} W_{ij} \quad \text{is the total cell load of } k \text{ that induced by part } j
\]

\[
\sum_{i=1}^{M} X_{ik} \quad \text{is the total number of machines in cell } k \text{ as given in (2)}.
\]

\[
\text{Min } Z = \sum_{i=1}^{M} \sum_{k=1}^{K} X_{ik} \sum_{j=1}^{P} (W_{ij} - L_{kj})^2
\] (3)
A tabu search strategy to solve cell formation problem with ratio level data

subject to

\[
\sum_{i=1}^{K} X_{ik} = 1 \quad (4)
\]

\[
\sum_{j=1}^{M} X_{ik} > 1 \quad (5)
\]

where \( X_{ik} = 0, 1 \).

3 Tabu Search algorithm

The Tabu Search is one of the local search techniques belongs to a mathematical optimisation method. This Tabu Search increases the performance of a local search method by using the memory structures: once a possible resolution has been determined, it is marked as taboo so that the algorithm does not visit that possibility again and again. Combinatorial optimisation problems are solved using this meta-heuristic algorithm Tabu Search. This algorithm uses a neighbourhood search method to iteratively move from a solution to a new solution in the neighbourhood of a solution until the stopping criterion has been satisfied. To explore regions of the search space that would be left unexplored by the local search procedure, Tabu Search modifies the neighbourhood structure of each solution as the search progresses. The solutions admitted to the new neighbourhood, are determined through the use of memory structures. Tabu list is the important type of memory structure used to determine the new neighbourhood solutions. A short term memory is called as a tabu list which contains the solutions that have been visited in the recent past. Tabu Search excludes solutions in the tabu list from the new neighbourhood. A variation of a tabu list prohibits solutions that have certain attributes. Selected attributes in solutions recently visited are labelled as ‘tabu-active’ Solutions that contain tabu-active elements are ‘tabu’. This type of short-term memory is also called ‘recency-based’ memory.

Tabu lists containing attributes can be more effective for some domains, although they raise a new problem. When a single attribute is marked as tabu, this typically results in more than one solution being tabu. Some of these solutions that must now be avoided could be of excellent quality and might not have been visited. To mitigate this problem, ‘aspiration criteria’ are introduced. These override a solution’s tabu state, thereby including the otherwise-excluded solution in the allowed set. A commonly used aspiration criterion is to allow solutions which are better than the currently-known best solution. Tabu Search is based on the premise that problem solving, in order to qualify as intelligent, must incorporate adaptive memory and responsive exploration. The adaptive memory feature of Tabu Search allows the implementation of procedures that are capable of searching the solution space economically and effectively.

3.1 Tabu Search – pseudo code

Giving a solution \( x^* \) is a initial machine sequence with objective function value \( z^* \) that have the cell load variation

Let \( x = x^* \) with \( z(x) = z^* \).
Iteration:
While stopping criteria is not reached do
Begin.
1 Select best admissible move that transforms $x$ in to $x'$ with objective value $z(x')$
2 Perform tabu list management: compute moves to be set tabu, i.e., updated the tabu list
3 Perform exchanges:
   $x = x'$ and $z(x) = z(x')$
   If $z(x) < z^*$ then
   $Z^* = z(x), x^* = x$
End if
Finally, $x^*$ is the best machine sequence of all determined solutions, with objective value $z^*$ where it shows the least cell load variation for the particular problem.

3.2 Tabu Search scheme
Start with an initial configuration, label this configuration as the current one, and evaluate the objective function for that configuration. Then using the current configuration and a procedure for obtaining trial solutions, generate a certain set of candidate moves and evaluate their corresponding objective function values. Repeat the above procedure for a certain number of iterations. On termination the best solution obtained so far is the solution obtained by the algorithm. Note that the move is chosen at certain iteration is put in the tabu list so that it cannot be reversed in the next few iterations. The basic role of the tabu list is to prevent cycling. The tabu list has a certain size, and when its length reaches that size and a new move enters that list, then the first move in the tabu list becomes non-tabu and the process continues. The tabu list size controls the Tabu Search to either emphasise exploration or intensification. If the tabu list is small, then intensification is emphasised, i.e., a local search around the current point is intensified. If the tabu list is large, then exploration is emphasised, i.e., points that are far from the current point are examined.

3.3 Algorithm for Tabu Search
1 For the given problem choose an initial sequence randomly.
2 Calculate the cell load variation for that initial solution using the objective function which will allocate the machines in different cells and assign the part assignment rule for the different machine cells to evaluate the exceptional element
3 Generate neighbourhoods and calculate the exceptional element for the each neighbourhood as said above.
4 Choose the best of the neighbourhoods with respective to the minimum exceptional element.
5 Store the best of the swapped variables in tabu list.
6 Repeat the steps 3 to 5 until the stopping criterion is met.
If the stopping criterion is met then display the minimised exceptional element for the particular problem.

3.4 Proposed Tabu Search algorithm

The different sections of the proposed Tabu Search to solve the cell formation problem associated with ratio level data is outlined as a flowchart as given in Figure 2.

**Figure 2** Flow chart of the proposed Tabu Search algorithm

The different sections are described below:

- **Input module**: The data related to the problem number of machines (M); number of parts (P) in the shop; number of cells (K); work load data for all operations are given as input.

- **Initialisation module**: In this step the initial sequence will be generated based on the number of machines and it is represented by their location of cells for each machine. In this if the sequence generated is 11222 then it represents the machines 1, 2, 3 are located in cell 1 and machines 4, 5, 6 are located in cell 2. The optimum allocation for the particular sequence can be find out the using the objective function $Z$. The
least Z value for the initial sequence will be taken among the machine allocations such as 12-3456, 123-456, 1234-56. After the machine assignment the parts will be allocated based on the part assignment rule.

3.4.1 Neighbourhood generation module

Pair wise exchange module is used to categorise the moves that guide from one solution to the next. With this module, we can get \( N(N-1)/2 \) neighbourhood solutions where the \( N \) is size of the string.

- **Tabu list:** The tabu list is updated based on the current solutions obtained. The tabu list operates like as short term memory to prevent the search from endlessly cycling between the same solutions. The goal of the tabu list is to permit good moves in each iteration without revisiting solutions already encountered. The tabu list consists of an array of elements where the number of elements equals the total number of the machines. Each element in the tabu list specifies a number that decides how long a move should remain unchanged.

- **Aspiration criteria:** The main condition of aspiration criteria is to overrule the tabu status when the solution looks attractive. Then repeat the steps until the stopping condition is reached. The termination criteria considered for this problem is 300 iterations. Finally, print the global best solution of the particular problem.

3.4.2 Part assignment procedure

The part assignment procedure is used to allocate the parts in to the machine cell. A machine cell which processes the part for a larger number of operations than any other machine cells and the respective part is assigned into that cell. Ties are broken by choosing the machine cell which indicates the highest percentage of machines visited by the part. In the case of tie again the machine cell with the least identification number is selected. Thus all the parts are assigned to all the cells which form part families using membership index given in (6).

\[
P_{kj} = \frac{f_{kj}}{f_{k}} \ast \frac{f_{k}}{f_{j}} \ast \frac{T_{kj}}{T_{j}}
\]

\( P_{kj} \) membership index of the part \( j \) belongs to the cell \( k \)

\( f_{kj} \) number of machines in the cell \( k \) required by the part \( j \)

\( f_{k} \) total number of machines in the cell \( k \)

\( f_{j} \) total number of machines required by the part \( j \)

\( r_{kj} \) processing time of the part \( j \) in the cell \( k \)

\( T_{j} \) total processing time required by the part \( j \).
3.4.3 Modified grouping efficiency

To evaluate the grouping in the cell formation problem the grouping efficiency and grouping efficacy are the performance measures usually considered in the literature. Since these measures are found not suitable for the problem considered in this work which is of real valued data a new performance measure known as modified grouping efficiency (MGE) proposed by Pandian and Mahaptra (2009) to assess the grouping efficiency of the proposed algorithm and to compare the efficiency with other algorithms found in the literature. The proposed MGE is calculated. To evaluate the grouping in the cell formation problem the performance measures usually considered in the literature are grouping efficiency and modified grouping efficiency. As this measures are not suitable for the problem considered in this work which is of real valued data a new performance measure known as MGE is formulated to assess the grouping efficiency of the proposed algorithm and to evaluate the efficiency with other algorithms found in the literature. The proposed MGE is calculated using (7).

\[
MGE = \frac{T_{pi}}{T_{pto} + \sum_{k=1}^{K} T_{ptk} + \sum_{k=1}^{K} T_{ptk} \cdot \frac{N_{vk}}{N_{ek}}}
\]

- \( T_{pi} \) total processing time inside the cells
- \( T_{pto} \) total processing time outside the cells
- \( T_{ptk} \) total processing time of the cell \( k \)
- \( N_{vk} \) number of voids in the cell \( k \)
- \( N_{ek} \) total number of elements in the cell \( k \).

4 Results and discussions

In this work, Tabu Search is considered for the cell formation problem considering operational time of the parts instead of conventional zero-one incidence matrix with the objective of reducing total cell load variation. The algorithm is coded in MATLAB and tested on the Pentium IV machine. Depending on the size of problem these values can be varied to tune the algorithm. The number of iterations is varied from problem to problem in the range of 100 to 1,000 iterations. MGE, is used to measure the performance of the grouping with real data values. The results are then compared with the results obtained from modified ART1 algorithm, K-means clustering algorithm and genetic algorithm as presented in Table 1. In Figure 3, the graph is showed for the proposed Tabu Search and other algorithm for the comparison of grouping efficacy for the benchmark problems. The size of machine-part incidence matrices considered in this paper ranges from \( 5 \times 7 \) to \( 20 \times 20 \). The number of cells is varied from 2 to 4.
<table>
<thead>
<tr>
<th>S. no.</th>
<th>Problem</th>
<th>Size of the problem</th>
<th>No. of cells</th>
<th>K-means</th>
<th>Modified ART1</th>
<th>GA</th>
<th>Proposed tabu search algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Exceptional elements</td>
<td>MGE</td>
<td>Exceptional elements</td>
<td>MGE</td>
</tr>
<tr>
<td>1</td>
<td>King and Nakornchai (1982)</td>
<td>5 × 7</td>
<td>2</td>
<td>2</td>
<td>77.25</td>
<td>2</td>
<td>77.25</td>
</tr>
<tr>
<td>2</td>
<td>Waghodekar and Sahu (1984)</td>
<td>5 × 7</td>
<td>2</td>
<td>2</td>
<td>78.34</td>
<td>2</td>
<td>78.34</td>
</tr>
<tr>
<td>3</td>
<td>Seifoddini (1989)</td>
<td>5 × 18</td>
<td>2</td>
<td>7</td>
<td>81.87</td>
<td>7</td>
<td>81.87</td>
</tr>
<tr>
<td>4</td>
<td>Kusiak and Chung (1992)</td>
<td>6 × 8</td>
<td>2</td>
<td>2</td>
<td>79.85</td>
<td>2</td>
<td>79.85</td>
</tr>
<tr>
<td>5</td>
<td>Kusiak (1987)</td>
<td>7 × 11</td>
<td>2</td>
<td>3</td>
<td>61.77</td>
<td>3</td>
<td>61.77</td>
</tr>
<tr>
<td>6</td>
<td>Boctor (1991)</td>
<td>7 × 11</td>
<td>2</td>
<td>1</td>
<td>65.48</td>
<td>1</td>
<td>65.48</td>
</tr>
<tr>
<td>7</td>
<td>Seifoddini and Wolfe (1986)</td>
<td>8 × 12</td>
<td>2</td>
<td>6</td>
<td>57.00</td>
<td>4</td>
<td>69.70</td>
</tr>
<tr>
<td>8</td>
<td>Chandrasekharan and Rajagopalan (1986a)</td>
<td>8 × 20</td>
<td>2</td>
<td>28</td>
<td>60.00</td>
<td>25</td>
<td>61.30</td>
</tr>
<tr>
<td>9</td>
<td>Chandrasekharan and Rajagopalan (1986b)</td>
<td>8 × 20</td>
<td>3</td>
<td>9</td>
<td>83.40</td>
<td>9</td>
<td>83.40</td>
</tr>
<tr>
<td>10</td>
<td>Mosier and Taube (1985)</td>
<td>10 × 10</td>
<td>3</td>
<td>0</td>
<td>77.14</td>
<td>0</td>
<td>77.14</td>
</tr>
<tr>
<td>11</td>
<td>Chan and Milner (1982)</td>
<td>10 × 15</td>
<td>3</td>
<td>0</td>
<td>93.28</td>
<td>0</td>
<td>93.28</td>
</tr>
<tr>
<td>12</td>
<td>Askin and Subramanian (1987)</td>
<td>14 × 23</td>
<td>2</td>
<td>2</td>
<td>59.43</td>
<td>2</td>
<td>60.59</td>
</tr>
<tr>
<td>13</td>
<td>Stanfill (1985)</td>
<td>14 × 24</td>
<td>4</td>
<td>7</td>
<td>68.13</td>
<td>7</td>
<td>68.13</td>
</tr>
<tr>
<td>14</td>
<td>McCormick et al. (1972)</td>
<td>24 × 16</td>
<td>4</td>
<td>34</td>
<td>46.70</td>
<td>30</td>
<td>51.39</td>
</tr>
<tr>
<td>15</td>
<td>Boctor (1991)-2</td>
<td>16 × 30</td>
<td>3</td>
<td>6</td>
<td>68.55</td>
<td>6</td>
<td>68.55</td>
</tr>
<tr>
<td>16</td>
<td>Boctor (1991)-3</td>
<td>16 × 30</td>
<td>3</td>
<td>4</td>
<td>67.89</td>
<td>4</td>
<td>67.89</td>
</tr>
<tr>
<td>17</td>
<td>Boctor (1991)-5</td>
<td>16 × 30</td>
<td>3</td>
<td>8</td>
<td>70.05</td>
<td>9</td>
<td>68.99</td>
</tr>
<tr>
<td>18</td>
<td>Boctor (1991)-6</td>
<td>16 × 30</td>
<td>3</td>
<td>4</td>
<td>69.73</td>
<td>4</td>
<td>69.73</td>
</tr>
<tr>
<td>19</td>
<td>Boctor (1991)-7</td>
<td>16 × 30</td>
<td>3</td>
<td>5</td>
<td>71.50</td>
<td>5</td>
<td>71.50</td>
</tr>
<tr>
<td>20</td>
<td>Boctor (1991)-8</td>
<td>16 × 30</td>
<td>3</td>
<td>1</td>
<td>65.18</td>
<td>13</td>
<td>59.71</td>
</tr>
<tr>
<td>21</td>
<td>Boctor (1991)-9</td>
<td>16 × 30</td>
<td>3</td>
<td>8</td>
<td>71.78</td>
<td>9</td>
<td>69.49</td>
</tr>
<tr>
<td>22</td>
<td>Srinivasan et al. (1990)</td>
<td>16 × 30</td>
<td>3</td>
<td>15</td>
<td>64.81</td>
<td>15</td>
<td>64.81</td>
</tr>
<tr>
<td>23</td>
<td>Mosier and Taube (1985)</td>
<td>20 × 20</td>
<td>2</td>
<td>42</td>
<td>49.13</td>
<td>22</td>
<td>51.10</td>
</tr>
</tbody>
</table>
A tabu search strategy to solve cell formation problem with ratio level data

Conclusions and future scope

In this work a Tabu Search is used to solve the cell formation problem using the work load data (non-binary real value) as an input matrix. This proposed algorithm is tested with benchmark problems found in the literature and the results are compared with the existing algorithms mainly K-means clustering algorithm, modified ART1 algorithm and the genetic algorithm. In addition to the commonly used measure of performance that is the number of exceptional elements, MGE is also applied to evaluate the efficiency of the proposed algorithm. The Tabu Search may be suitably modified and employ to solve the cell formation problem with other non-binary real value data like, production volume, machine duplication, machine capacity and product sequence.

References


