Final Version (modified on November 5, 2003)

REF: JEM/CRV/IJPR 30064 "An efficient algorithm for cell formation with sequence data, machine replications and alternative process routings" by Sachin Jayaswal and Gajendra K. Adil

An efficient algorithm for cell formation with sequence data, machine replications and alternative process routings

Sachin Jayaswal[1] and Gajendra Kumar Adil[2]
[1]Industrial Engineering and Operations Research, Indian Institute of Technology, Bombay
[2]Shailesh J. Mehta School of Management, Indian Institute of Technology, Bombay

Address for correspondence:

Dr. Gajendra Kumar Adil
Shailesh J. Mehta School of Management
Indian Institute of Technology, Bombay, Powai
Mumbai 400076
India
Tel. 91-22-2576-7738
Fax: 91-22-2572-2872
Email: gajendra@som.iitb.ac.in

An efficient algorithm for cell formation with sequence data, machine replications and alternative process routings

Sachin Jayaswal and Gajendra Kumar Adil

Abstract

Cell Formation (CF) is an important problem in the design of cellular manufacturing system. Despite a large number of papers on CF been published, only a handful of them incorporate operation sequence in inter-cell move calculations and consider alternative process routings, cell size, production volume and allocating units of identical machines into different cells. Modeling the above factors makes CF problem complex but more realistic. This paper develops a model and solution methodology for a problem of Cell Formation to minimize sum of costs of inter-cell moves, machine investment and machine operating costs considering all the factors mentioned above. Algorithm comprising of Simulated Annealing and Local Search heuristics has been developed to solve the model. A limited comparison of the proposed algorithm with optimal solution generated by complete enumeration of small problems indicates that the algorithm produces solution of excellent quality. Large problems with 100 parts and 50 machine types are efficiently solved using the algorithm.

Keywords: Cell formation; alternative part routings; replicate machines; simulated annealing algorithm.

1. Introduction

In cellular manufacturing (CM) the production system is organized into smaller units called cells. Cell formation (CF) process identifies the machine group and the part family for each cell in the CM system. CF is such an important area of research that several CF approaches and algorithms have been developed in the literature considering various production factors, scenario and objectives. Implementing CM system can help organizations achieving benefits in several ways, such as, simplified planning and control procedures, reduced throughput times, reduced work-in-process inventory, reduced setup times and reduced material handling (Wemmerlov and Hyer, 1989).

Small and independent cells facilitate achieving most of the benefits that CM system realizes (Burbidge, 1991). Forming small and independent cells is, therefore, an important goal of CF. Independent cells eliminate inter-cell moves and the associated cost. Consideration of operation sequence and production volumes of parts is required to realistically capture the cost involved in inter-cell moves. Inter-cell moves can be reduced either by exploiting alternative process routings or by replicating a sufficient number of bottleneck machines into appropriate cells. This can be illustrated by considering two consecutive operations, o1 and o2, of a part assigned respectively to machine type m1 in cell c1, and machine type m2 in cell c2, thus resulting in an inter-cell move. Operation o2 needs to move to cell c2 as cell c1, it is assumed, does not have enough capacity on machine m2. If operation o2 can alternatively be performed on machine type m3 available in cell c1, it (operation o2) can be shifted to cell c1 to eliminate the inter-cell move. However, machine m3 may require higher operating cost (variable cost). Alternatively, one can eliminate the aforementioned inter-cell move by incurring investment (fixed cost) in additional machine of type m2 in cell c1. The above example although illustrates that the use of alternative machine or machine replication can help reduce inter-cell moves, same cannot be done without consideration of machine operating cost or machine investment cost. Thus there exists a trade off between operating cost (for using alternative process routings), fixed cost (of machine replication) and cost of inter-cell moves. CF procedure must, consider availability of alternative process plan, allocation of multiple units of a given machine type to more than one cell as needed and operation sequence in computation of inter-cell moves in order to meaningfully capture the above trade off. Allowing allocation of multiple units of a machine type to different cells makes the cell formation problem more realistic although it complicates modeling (Sofianopolou, 1999). The same is true in considering operation sequence in computation of inter-cell moves. Literature on CF considering sequence data, machine replication and alternative process routings and machine cost simultaneously is very limited.

Nair and Narendran (1998) and Won and Lee (2001) use indices to account for the operation sequence but do not consider replicate machines, alternative process routings or machine cost. Wu (1998), Wu and Salvendy (1999) and Lee and Chen (1997) consider inter-cell moves, operation sequence, production volume as well as replicate machines, but alternative process routing or the trade off between cost of inter-cell moves and machine cost is not considered. Lozano et al. (1999), Sofianopoulou (1999) and Zhao and Wu (2000) consider presence of alternative routing but do not take into account trade off between cost of inter-cell moves and machine cost.

Su and Hsu (1998) solve a cell formation problem to minimize costs of machines, intra-cell and inter-cell moves. Seifoddini (1989) uses the duplication cost and the associated reduction in inter-cell material handling cost to arrive at a decision on machine duplication. However, alternative process routing is not considered in either of the papers.

Rajamani et al. (1996) and Shanker and Agarwal (1997) account for alternative process routing and cell size but do not capture the operation sequence correctly in inter-cell move. Yin and Yasuda (2002) take into account operation sequence in the computation of similarity index, but, it is not used for inter-cell move calculations. Vakharia and Chang (1997) consider the objective of minimizing total cost of inter-cell moves and investment in machines and have accounted for operation sequence, production volume, replicate machines and cell size. They, however, do not consider alternative process routing. Beaulieu et al. (1997) take into consideration all the above factors and develop a two stage heuristic algorithm. However, quality of solution produced from their algorithm has not been assessed.

A review of the literature on CF considering minimizing inter-cell moves reveals that works that consider all the important factors – production volume, operation sequence, splitting (or allocating) replicate machines to different cells, alternative process routing and cell size – simultaneously are few. Many minimize inter-cell moves without accounting for the trade off with machine costs. The objective of this paper is threefold: first, to consider all the above factors in developing a CF model; second, to develop an efficient solution approach capable of handling large problems; and third, to compare the results with optimal solution for problems that can be optimally solved with reasonable computational efforts.

The remainder of the paper is organized as follows. In § 2, the problem statement and notations are described and mathematical model is developed. The solution procedure is presented in § 3. § 4 illustrates the solution procedure using an example. § 5 compares the solution obtained by the algorithm with the optimal solution and provides computational experience on randomly generated large size problems. Conclusions are presented in § 6.

2. Problem formulation

There are M (m = 1, 2,…., M) machine types, each of which can be acquired in multiple units and can be distributed among C (c = 1, 2,…., C) cells as required. One unit of machine type m is assumed to have an annual production capacity of A_m hours. Further, a machine type m can be utilized up to a fraction MUT_m of its production capacity. There are N (i = 1, 2, 3….., N) parts with known annual demand d_i for part i. Further, each part i requires J_i (j = 1, 2, 3…, J_i) operations. This paper considers rejects and, therefore, the number of useful units of part i after each operation j decreases by the amount of rejection (R_{ij} %). Number of part i input units for operation j, d_{ij}, can be calculated using equation (1).

$$d_{ij} = \frac{d_i}{\prod_{k=j}^{J_i} (1 - R_{ik})} \quad \text{...(1)}$$

An operation j for part i can be performed using K_{ij} (k = 1, 2,…, K_{ij}) alternatives (or options) in terms of choosing a machine type from available M machine types. It is assumed that the reject rate is independent of the machine chosen. Let P_{ijk} be the time for operation j of part i using k th alternative (for machine). It needs to be indicated which machine type m is used as alternative k. A set of parameters A_{ijkm} having values of 1 when kth alternative (for machine) for operation j of part i is provided by machine type m, 0 otherwise, is defined for this purpose.

At design stage, it is assumed that splitting of demand between machine types or cells is not allowed for an operation. The cell formation decisions involve determining, for each part i and operation j, the selection of a unique alternative (for machine) k and cell c, the associated load (number of machines, not necessarily an integer) on machine of type m in each cell c, and number of machines (load rounded to the next higher integer) of type m needed to be acquired in each cell c. The following three variables are defined to represent the above decisions.

Y_{ijkc} = 1 if operation j on part i is performed in cell c using kth alternative (for machine), and 0 otherwise.

X_{mc} = load on machine type m in cell c (not necessarily having an integer value).

$\lceil X_{mc} \rceil$ = number of machine type m acquired in cell c. X rounded to the next higher integer is denoted by $\lceil X \rceil$.

The following two cost parameters for one unit of machine type m are defined.

MFC_m = Annual fixed cost (investment).

MVC_m = Annual variable cost (operating cost) of a fully loaded machine. If it is partially loaded the cost is computed pro rata.

Space restrictions on a cell limit the maximum number of machines it can accommodate to S. Cost H is attached to a unit inter-cell move and is assumed to be independent of the location of the cells in the plant.

An integer programming model can be written as follows to model the problem just described.

Mathematical model

$$Minimize\ z = \sum_{m=1}^{M}\sum_{c=1}^{C}\lceil X_{mc} \rceil * MFC_m + \sum_{m=1}^{M}\sum_{c=1}^{C} X_{mc} * MVC_m$$

$$+ d_{ij+1} * \frac{H}{2}\sum_{i=1}^{N}\sum_{j=1}^{J_i-1}\sum_{c=1}^{C}\left|\sum_{k=1}^{K_{ij}} Y_{ij+1kc} - \sum_{k=1}^{K_{ij}} Y_{ijkc}\right| \qquad (2)$$

$$\sum_{c=1}^{C}\sum_{k=1}^{K_{ij}} Y_{ijkc} = 1 \qquad \forall i, j \qquad (3)$$

$$\frac{\sum_{i=1}^{N}\sum_{j=1}^{J_i}\sum_{k=1}^{K_{ij}} A_{ijkm} * Y_{ijkc} * P_{ijk} * d_{ij}}{A_m * MUT_m} = X_{mc} \qquad \forall m, c \qquad (4)$$

$$\sum_{m=1}^{M}\lceil X_{mc} \rceil \le S \qquad \forall c \qquad (5)$$

$$Y_{ijkc} \in [0,1] \qquad \forall i, j, k, c \qquad (6)$$

$$X_{mc} \ge 0 \qquad \forall m, c \qquad (7)$$

In this model, the objective function (2) is the sum of machine investment, machine operating cost and cost of inter-cell moves. Constraint set (3) ensures that each operation on a part is completely carried out on only one alternative machine type and in one cell. At design stage, it is assumed that splitting of demand for an operation between machine types or cells is not permitted. Constraint set (4) estimates the number of machines of each type required in each cell, based on the available annual production time and the load allocated to that machine. CF objective of forming small cells is included as a limit on the size of each cell using constraint set (5). Binary and non-negativity restrictions on the decision variables are enforced through constraint sets (6) and (7) respectively.

3. Solution algorithms

For a problem of practical size, the formulation presented in § 2 will involve too many integer variables to be solved optimally. Therefore, this paper, like many other papers on CF, resorts to a heuristic approach to solving the model (§ 2). Simulated annealing algorithm (SAA) is developed to produce a solution, which is further improved by using a local search (LS) procedure. A feasible solution has, for each part and operation, a unique selection of a machine type and a cell that satisfies constraints (3) to (7). Cost of a solution is the value of the objective function (2).

3.1 Simulated annealing algorithm (SAA)

SAA implementation of this paper uses the following scheme for generation of initial solution, neighbourhood solution and termination of the algorithm.

- *Initial Solution*: Initial solution is generated by randomly selecting values for Y variables until a feasible solution is obtained.
- *Neighbourhood solution:* A neighbourhood solution is obtained by perturbing an operation assignment of a part to a different machine type/cell.
- *Termination criteria:* The algorithm terminates if one of the following conditions holds true:
    - Number of iterations reaches a pre-specified maximum value.
    - There is no improvement in the objective function value for the last 10 iterations.
    - Acceptance ratio (number of transitions accepted/ number of transitions attempted) at any temperature reaches a minimum pre-specified value.

The detailed Simulated Annealing Algorithm (SAA) is given in the appendix.

3.2 Local search (LS)

The Local Search starts with the final solution obtained from SAA. It picks the first operation of the first part and tries allocating it to the next alternative machine within the same cell, satisfying the constraint on cell size. Once such an alternative machine for the current operation, which gives a better objective function value, is found the new solution is accepted and the whole process restarts from the first operation of the first part. If no such alternative machine to the current operation can be found within the same cell, it tries searching for the same or alternative machine in the next cell that satisfies the constraint on the cell size. It is attempted to find out move that improves the solution by sequentially considering all the operations of all the parts.

4 An illustrative example

Algorithm developed in this paper is illustrated using the problem of 8 parts and 8 machine types from Beaulieu et al. (1997), which has the required data.

----------------------- [Insert table 1 and table 2 about here] -----------------------------

Step 0: Read and Initialize

0.1 Data on process requirements (P_{ijk}), reject rates (R_{ij}), number of alternative machines (K_{ij}), demand (d_i), machine fixed cost (MFC_m) and machine variable cost (MVC_m) are given in table 1 and table 2. Other data used are as follows: Maximum number of cells, C = 2; maximum number of machines allowed in a cell, S = 6; and unit cost of inter-cell move, H = 0.

0.2 Simulated annealing parameters are read as: T_0 = 70,000; AT_{max} = 100; α = 0.97; i_{max} = 1,000; R_f = 0.0005; L_{max} = 300; NM = 100; NC=100; and NP=100.

0.3 Initial iteration counter and temperature are fixed as i = 0; and $T_i = T_0$.

0.4 The initial assignment of operations to machines and cells, SOL^0 and the calculation of the objective function value OBJ^0 are shown in table 3 and table 4. Initial solution SOL^0 and initial objective function value OBJ^0 become the best solution SOL^{best} and best objective function value OBJ^{best}, respectively.

Total cost = machine cost (fixed + variable) + cost of inter-cell moves

= (26,848.4+26,856.4+28,793.6+29,743.8)+00.00 = 112,242.3

-----------------------[Insert table 3 and table 4 about here]----------------------------

Step 1: Outer loop, i.e. steps (1.1 - 1.7), is executed for 162 iterations. The solution is terminated by the criteria of final acceptance ratio.

Step 2: The final solution is produced in table 5 and table 6.

Total cost = machine cost (fixed + variable) + cost of inter-cell moves

= (20,285.46+20,285.46+33,084.47+30,473.39) +00.00 = 111,296.35

-----------------------[Insert table 5 and table 6 about here]----------------------------

The Local Search could not find any further improvement upon the solution generated by SAA for this problem. The objective function value obtained as reported by Beaulieu et al. (1997) is 111,286, which is very close to the value obtained using our algorithm. The small difference of 10.35 (< 0.01 %) in the objective function value can be attributed to the rounding error. However, for this problem, on verifying the results obtained by Beaulieu et al. (1997), it appears that there is a typographical error in the case of processing time for OP51 on machine M3. The processing time of 20 seems to have been mistyped as 2. For the comparison of the algorithm developed in this paper, the corrected value, 20, is used, although the proposed algorithm gave even a better solution with objective function value of 106,963.891 using the value, 2, as reported in the paper.

In the next section, comparison of solution from the proposed algorithm with the optimal solution for small size problems is presented followed by the computational experience with large problems.

5 Computational experience

§ 5.1 gives the scheme for generation of data for computational experience. § 5.2 gives comparison of the solutions obtained with the optimal for small size problems. Optimal solutions are generated by complete enumeration, which becomes computationally very difficult for larger size problems. Computational experience with larger size problems, therefore, is reported separately in § 5.3.

5.1 Generation of Data

Five small size problems are solved in § 5.2 for comparison of the solution with the optimal. In § 5.3, the heuristic is evaluated for large problems. For each problem, five instances are generated randomly. The following scheme is used for data generation.

- *Max J* is defined to be the longest operation sequence amongst parts. The length of the operation sequence, J_i, for part i is fixed at *Max J* for all operations in § 5.2. For problems of larger size in § 5.3, J_i is randomly generated using a discrete uniform distribution with parameters 1 and *Max J*. The value of *Max J* is varied for each problem in § 5.2, while it is set at 10 for all the problems in § 5.3.
- The annual demand (number of completed units), d_i, for each part i, is randomly generated using discrete uniform distribution with parameters 3,000 and 6,000. The reject percentage, R_{ij}, for part i, operation j is randomly generated using the same distribution with parameters 1 and 3. The annual demand d_{ij}, for part i, operation j, is calculated using the data on part demand and reject percentage using (1).
- The number of alternative machines, K_{ij}, for part i, operation j is randomly generated using a discrete uniform distribution with parameters 1 and *Max K*. The value of *Max K* is set at 2 for problems in § 5.2 and at 3 for problems in § 5.3. The particular machine type m used as kth alternative to carry out operation j of part i is randomly generated using a discrete uniform distribution with parameters 1 and M. This fixes the value of A_{ijkm}. It is possible that a machine type may appear more than once in the operation sequence. However, consecutive operations for a part are not allowed on the same

machine type (consecutive operations on the same machine type are considered as a single operation).

- The processing time, P_{ijk}, for part *i*, operation *j* on machine alternative *k* for machine type is randomly generated using a discrete uniform distribution with parameters 12 and 25 (minutes).

- The fixed and variable machine costs, MFC_m and MVC_m for machine type *m* are randomly generated using discrete uniform distribution with parameters 2,000 and 10,000, and 3,000 and 20,000, respectively.

- The maximum utilization ratio MUT_m for all machine types *m* is fixed at 0.90. The cost of inter-cell move, *H* is fixed at 0.5.An annual production capacity *Am*, of 2,000 and 4,000 hours are assumed for problems in § 5.2 and § 5.3, respectively.

5.2 Comparison with optimal solution

- **Implementing the heuristic**

  Problem characteristics are shown in table 7. The values of Simulated Annealing parameters are selected as given below:

  T_0 = 20,000; α = 0.95; i_{max} = 1,000; AT_{max} = 50; L_{max} = 100; R_f = 0.01; *NM* = 10; *NC* = 20; *NP* = 25.

------------------------[Insert table 7 about here]------------------------------

- **Performance measures**

  - *Optimality gap*: The percentage gap in the objective function value from the optimal is reported in table 7.
  - *Computation time*: The computation time (cpu time in seconds) on Sun Ultra spark 10, 400 MHz using Solaris 2.8 for Simulated Annealing Algoroithm, Local Search and Optimal Solution by complete enumeration are reported in table 7.

- **Results**

It is observed from table 7 that in 12 out 25 cases, SAA gives optimal solution. In one case, the optimal solution is achieved after Local search. The worst optimality gaps obtained are 9.691% and 4.687% for SAA and Local Search, respectively. The total number of enumerations to obtain an optimal solution can be expressed as:

$$\Pi_{ij} [K_{ij} * C]$$

Table 7 also shows that the cpu time for the optimal solution by complete enumeration increases rapidly with the size of the problem. This time increases from as low as 0.3 seconds for a problem with 5 parts, 2 operations, 2 cells to as high as 421.78 seconds for a problem with 6 parts, 2 operations, 3 cells. This time increases exponentially with number of parts and operations. The computation time for SAA, however, shows little variation with the size of problem. SAA, therefore, proves to be computationally more efficient with the increase in the size of problem.

5.3 Computational experience with large problems

Large problems with 100 parts and 50 machine types are considered. Maximum number of cells, *C*, allowed is fixed at 4 for all the problems Maximum number of machines allowed in a cell, *S*, is fixed at 50. All other data are generated as described in § 5.1.

- **Implementing the heuristic**

Values of the following Simulated Annealing parameters are fixed:

T_0 = 500,000; i_{max} = 2,000; R_f = 0.01; *NM* = 10; *NC* = 20; *NP* = 500.

SAA is run for different length of time and its performance in terms of solution quality recorded. This is achieved by varying the cooling rate (α), maximum accepted transition at each temperature (AT_{max}) or maximum transitions at each temperature (L_{max},) as shown in table 8.

------------------------[Insert table 8 about here]-----------------------------

Performance measures

- *Percentage improvement*: Percentage improvement in the objective function value achieved using each of SAA and Local Search is reported in table 8. Percentage improvement is reported relative to the randomly generated initial solution.
- *Computation time*: The computation time (in seconds) on Sun Ultra spark 10, 400 MHz using Solaris 2.8 for SAA and Local Search are reported in table 8.

- **Results**

The maximum, minimum and the average values of computation times and % improvements obtained from solution of five instances of the nine problems are presented in table 8. Percentage improvement as large as 30.25 and 10.68 in the objective function value are achieved for SAA and Local Search, respectively. With increase in values of α (decreasing cooling rate), AT_{max} and L_{max} SAA runs longer as indicated by the higher cpu times in table 8. However, solution quality from SAA also improves as indicated by greater percentage improvement in the objective function values. The counterpart LS heuristic exhibits the opposite trends in solution time and quality. However, the quality of final solution obtained by running, both SAA and LS sequentially does not vary significantly for these nine parameter settings, and the total improvement in the objective function value remains around 35%, although the total computation time varies. The total computation time, as well as, SAA computation time, increases with the increase in values of α, AT_{max} and $L_{max.}$ Thus running SAA for a shorter time appears to be a better strategy for the problem tested.

6 Conclusions

There are very few papers that simultaneously consider operation sequence, machine replications and alternative process routings in cell formation. Modeling these factors makes Cell Formation problem complex but more realistic. In this paper, a model and solution methodology for a problem of Cell Formation to minimize sum of costs of inter-cell moves, machine investment and machine operating costs are developed. The model takes into account operation sequence in computation of inter-cell moves, allocating replicate machines of same type to different cells, alternate process routes, production volumes and cell size, amongst other factors. Solution algorithm comprising of Simulated Annealing and Local Search heuristics has been developed to solve the model. Computational experiences show that the algorithm generates good quality solution and capable of solving large problems.

Acknowledgements

We thank authors of paper (Beaulieu et al., 1997) for providing us the data set reported in table 1 and table 2 and the two anonymous referees for their valuable suggestions, which have helped improve the paper.

Appendix: Simulated Annealing Algorithm (SAA)

Step 0: Read and Initialize

0.1 Read process requirements: demands, d_{ij}; alternative machines and processing times, P_{ijk} ; machine capacities (A_m) and utilization limits (MUT_m); machine costs (FMC_m and VMC_m); maximum number of cells, C; maximum cell size, S; and unit cost of inter-cell move, H.

0.2 Define the annealing parameters: initial temperature, T_0; maximum transitions at each temperature, L_{max}; maximum accepted transition at each temperature, AT_{max}; decrementing factor, α; maximum number of iterations, i_{max}; and final acceptance ratio, R_f. Also, define values of parameters *NM*, *NC*, *NP* to be used in the neighbourhood search in steps 1.3.2c, 1.3.2e, 1.3.2f.

0.3 Initialize iteration counter: $i = 0$, temperature $T_i = T_0$.

0.4 Generate initial feasible solution, SOL^0 and determine its objective function value, OBJ^0. Initialize the best solution: $SOL^{best} = SOL^0$ and the best objective function value: $OBJ^{best} = OBJ^0$.

Step 1: Execute outer loop, i.e., steps (1.1 - 1.7) until conditions in step 1.7 are met.

1.1 Initialize inner loop counter $l = 0$, and accepted number of transitions $AT = 0$.

1.2 Initialize solution for inner loop, $SOL^i_0 = SOL^i$, $OBJ^i_0 = OBJ^i$.

1.3 Execute inner loop, i.e., steps (1.3.1 – 1.3.5) until condition in step 1.3.5 is met.

1.3.1 Update $l = l + 1$

1.3.2 Generate a feasible neighbourhood solution by perturbing an operation assignment of a part to a machine/cell and obtaining new machine/cell allocation for the operation (get SOL^i_l, OBJ^i_l) following steps 1.3.2 a- 1.3.2 g.

a) Randomly select a part, say *i*, and an operation, say *j*.

b) If the selected operation has alternative machine(s), go to c), else go to d).

c) Select randomly a different alternative machine that does not violate the constraint on cell size. If no alternative machine is found in a pre-specified number of trials (*NM*) that satisfies the constraint on cell size, go to d).

d) Assign the operation to a randomly selected different cell that satisfies the constraint on cell size.

e) If for the selected operation no alternative machine/cell can be found in a pre-specified number of iterations (*NC*) that satisfies the constraint on cell size, go to a).

f) If no part can be found in a pre-specified number of iterations (*NP*) for which there is possible any operation with alternative machine/cell satisfying the constraint on cell size, go to step 2.

g) Generate new solution for this operations assignment SOL^{i}_{l}, and calculate new objective function value, OBJ^{i}_{l}.

1.3.3 Let $\delta = OBJ^{i}_{l} - OBJ^{i}_{l-1}$.

1.3.4 If $\delta \leq 0$ or random $(0,1) \leq e^{\frac{-\delta}{T_i}}$ is true, then, accept SOL^{i}_{l} and OBJ^{i}_{l} and increment AT by 1. Otherwise, reject the solution and assign, $SOL^{i}_{l} = SOL^{i}_{l-1}$, $OBJ^{i}_{l} = OBJ^{i}_{l-1}$.

1.3.5 If one of the following conditions holds true: $AT \geq AT_{max}$ or $l \geq L_{max}$, then assign l to L_i (length of Markov chain), terminate the inner loop and go to 1.4, else continue the inner loop and go to 1.3.1.

1.4 Update: $i = i + 1$

1.5 Update $SOL^{i} = SOL^{i-1}_{l-1}$, $OBJ^{i} = OBJ^{i-1}_{l-1}$. If $OBJ^{i} < OBJ^{best}$ then $SOL^{best} = SOL^{i}$ and $OBJ^{best} = OBJ^{i}$.

1.6 Reduce the cooling temperature: $T_i = \alpha * T_{i-1}$.

1.7 If one of the following conditions holds true: $i \geq i_{max}$; or the acceptance ratio (defined as $AT / L_i) \leq R_f$; or the objective function value for the last 10 iterations remains the same, then terminate the outer loop and go to 2, else continue the outer loop and go to 1.1.

Step 2: Print the best solution obtained and terminate the procedure.

References

Beaulieu, A., Gharbi, A., and Ait-Kadi, D., 1997, “An algorithm for the cell formation and the machine selection problems in the design of a cellular manufacturing system”, *International Journal of Production Research*, 35 (7), 1857-1874.

Burbidge., J. L., 1991, “Production flow analysis in planning group technology”, *Journal of Operations Management*, 10(1), 5-27

Lozano, S., Guerrero. F., Eguia, L., and Onieva. L., 1999, “Cell design and loading in the presence of alternative routing”, *International Journal of Production Research*, 37 (14), 3289-3304.

Lee, S.-D., and Chen, Y.-L., 1997, “A weighted approach for cellular manufacturing design: minimizing inter-cell movement and balancing workload among duplicate machines”, *International Journal of Production Research,* 35 (4), 1125-1146.

Nair, G.J., and Narendran, T.T., 1998, “CASE: A clustering algorithm for cell formation with sequence data”, *International Journal of Production Research*, 36 (1), 157-179.

Rajamani, D., Singh, N., and Aneja, Y. P., 1996, “Design of cellular manufacturing systems”, *International Journal of Production Research*, 34 (7), 1917-1928.

Seifoddini, H., 1989, “Duplication process in machine cells formation in Group Technology”, *IEE Transactions*, 21(4), 382-388.

Shanker, K., and Agarwal, A. K., 1997, “Models and solution methodologies for the generalized grouping problem in cellular manufacturing”, *International Journal of Production Research*, 35 (2), 513-538.

Sofianopoulou, S., 1999, “Manufacturing cells design with alternative process plans and/or replicate machines”, *International Journal of Production Research*, 37 (3), 707-720.

Su, C.-T., and Hsu, C.-M., 1998, “Multi-objective machine-part cell formation through parallel simulated annealing”, *International Journal of Production Research,* 36 (8), 2185-2207.

Vakharia, A. J., and Chang, Y.L., 1997, “Cell formation in group technology: A combinatorial search approach”, *International Journal of Production Research*, 35 (7), 2025-2043.

Wemmerlov, U. and Hyer, N.L., 1989, “Cellular manufacturing in the U.S. industry-a survey of users”, International Journal of Production Research, 27, 1511-1530.

Won, Y., and Lee, K.C., 2001, “Group technology cell formation considering operation sequences and production volumes”, *International Journal of Production Research*, 39 (13), 2755-2768.

Wu, N., 1998, “A concurrent approach to cell formation and assignment of identical machines in group technology”, *International Journal of Production Research*, 36 (8), 2099-2114.

Wu, N., and Salvendy. G., 1999, “An efficient heuristic for the design of cellular manufacturing systems with multiple identical machines”, International *Journal of Production Research*, 37 (15), 3519-3540.

Yin, Y. and Yasuda, K., 2002, “Manufacturing cells’ design in consideration of various production factors, *International Journal of Production Research*, 40(4), 885-906.

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.

Part, *i*	**Part demand, d_i**	**Operation, *j***	**% Rejects (R_{ij})**	**Operation demand, d_{ij}**	**No. of alternative machine types, K_{ij}**	**[Alternative *k*] Machine types, *m* (Process times P_{ijk} on these machines, expressed in minutes)**
P1	5,000	OP11	1	5,205.633	2	[1] M4 (15), [2] M6 (15)
		OP12	1	5,153.577	1	[1] M5 (17)
		OP13	2	5,102.041	2	[1] M1 (6), [2] M2 (6)
P2	4,750	OP21	2	4,945.856	1	[1] M3 (23)
		OP22	2	4,846.939	1	[1] M1 (15)
P3	4,000	OP31	2	4,250.372	1	[1] M3 (18)
		OP32	1	4,165.365	2	[1] M4 (16); [2] M6 (13.5)
		OP33	3	4,123.711	1	[1] M7 (25)
P4	3,750	OP41	1	3,864.788	2	[1] M2 (15.5), [2] M4 (15)
		OP42	1	3,826.14	1	[1] M6 (19)
		OP43	1	3,787.879	1	[1] M5 (14)
P5	5,500	OP51	2	5,726.781	2	[1] M3 (20)*, [2] M8 (18.5)
		OP52	2	5,612.245	2	[1] M2 (12), [2] M4 (13)
P6	3,500	OP61	1	3,643.943	2	[1] M3 (15.6), [2] M8 (13)
		OP62	2	3,607.504	2	[1] M2 (18.5), [2] M4 (17.75)
		OP63	1	3,535.354	1	[1] M6 (14.3)
P7	4,000	OP71	2	4,250.372	2	[1] M3 (12), [2] M8 (10.3)
		OP72	1	4,165.365	2	[1] M2 (11), [2] M4 (17)
		OP73	3	4,123.711	1	[1] M7 (26)
P8	5,500	OP81	1	5,785.299	2	[1] M2 (18), [2] M4 (17)
		OP82	3	5,727.377	1	[1] M1 (12)
		OP83	1	5,555.556	3	[1] M4 (22.5), [2] M5 (21), [3] M6 (22)

*value reported in the paper is 2 which has been corrected.

Table 1: Parts demand and operation requirements (extracted from Beaulieu et al., 1997)

Machine type, m	M1	M2	M3	M4	M5	M6	M7	M8
Fixed Cost, MFC_m	2,386.52	3,579.79	5,369.68	5,966.31	8,509.99	6,712.10	6,264.63	6,562.94
Variable Cost, MVC_m	3,289.63	6,438.27	11,376.54	6,974.54	8,301.64	6,920.94	17,760.74	11,473.68
Am (Hours)	3,983.6	3,950.8	3,950.8	3,942.4	3,934.4	3,942.4	3,926.4	3,947.2

Table 2: Annual cost and available hours of machines (extracted from Beaulieu et al., 1997

Cell, *c*	Machines assigned in the cell		Operations (OP*ij*)assigned to machine type *m* in cell *c*
	Machine type, *m*	**No. of units, ⌈X*mc*⌉**	
1	M1	1	OP13, OP22, OP82
	M2	2	OP41, OP52, OP62, OP72, OP81
	M3	1	OP21, OP31
	M4	2	OP11, OP32, OP83
2	M3	1	OP51, OP61, OP71
	M5	1	OP12, OP43
	M6	1	OP42, OP63
	M7	1	OP33, OP73

Table 3: Initial assignment of operations to machines and cells (step 0.4)

Machine Type,*m*	Machine Req., ⌈X*mc*⌉		Load on machines, X*mc*		Fixed Cost, MFC_m		Variable Cost, MVC_m	
	Cell, c=1	**Cell, c=2**	**Cell, c=1**	**Cell, c=2**	**Cell, c=1**	**Cell, c=2**	**Cell, c=1**	**Cell, c=2**
M1	1		0.72		2,386.52		2,367.89	
M2	2		1.45		7,159.58		9,341.56	
M3	1	1	0.80	0.47	5,369.68	5,369.68	9,131.13	5,336.43
M4	2		1.14		11,932.62		7,953.04	
M5		1		0.60		8,509.99		4,945.91
M6		1		0.52		6,712.1		3,606.18
M7		1		0.90		6,264.63		15,855.3
Total					26,848.4	26,856.4	28,793.6	29,743.8

Table 4: Annual cost on machines for initial solution.

Cell, *c*	Machines assigned in the cell		Operations (OP*ij*)assigned to machine type *m* in cell *c*
	Machine type, *m*	No. of units, ⌈X*mc*⌉	
1	M1	1	OP13, OP22, OP82
	M2	1	OP52, OP81
	M5	1	OP12, OP43
	M6	1	OP32, OP42, OP63
	M7	1	OP33, OP73
2	M2	1	OP41, OP62, OP72
	M3	2	OP21, OP31, OP51, OP61, OP71
	M4	1	OP11, OP83

Table 5: Final assignment of operations to machines and cells.

Machine Type, *m*	Machine Req., ⌈X*mc*⌉		Load on machines, X*mc*		Fixed Cost, MFC_m		Variable Cost, MVC_m	
	Cell, *c*=1	Cell, *c*=2	Cell, *c*=1	Cell, *c*=2	Cell, *c*=1	Cell, *c*=2	Cell, *c*=1	Cell, *c*=2
M1	1		0.72		2,386.52		2,368.53	
M2	1	1	0.72	0.73	3,579.79	3,579.79	4,654.87	4,687.06
M3		2		1.74		10,739.36		19,795.2
M4		1		0.86		5,966.31		5,991.13
M5	1		0.60		8,509.99		4,947.78	
M6	1		0.76		6,712.10		5,252.99	
M7	1		0.89		6,264.63		15,860.30	
Total					27,453.03	20,285.46	33,084.47	30,473.39

Table 6: Annual cost on machines for final solution

Prob No.	Prob instance	No. of Parts, *N*	No. of M/cs types, *M*	Max No. of Oprn *MaxJ*	Max No. of Alt-mach, *MaxK*	No. of Cells *C*	Cell size, *S*	Optimality gap (%)		Optimal value	Computational time (cpu time in sec)		
								SAA	LS		SAA	LS	Optimal
1		5	5	2	2	2	5						
	1							1.13	1.13	77,494	0.56	0.02	0.31
	2							0	0	37,617	0.33	0.01	0.17
	3							0	0	93,840	0.42	0.01	0.17
	4							2.13	2.13	49,402	0.23	0.01	0.34
	5							0	0	124,357	4.68	0.01	0.30
2		4	5	3	2	2	6						
	1							0	0	103,854	0.34	0.01	5.12
	2							9.69	3.43	58,387	4.93	0.02	5.39
	3							2.27	0	138,168	0.26	0.02	0.72
	4							0	0	54,036	0.28	0.02	0.43
	5							2.80	2.80	114,018	0.29	0.01	1.42
3		5	5	2	2	3	4						
	1							0.15	0.15	78,366	0.44	0.02	22.08
	2							0	0	37,617	0.37	0.02	11.09
	3							0	0	93,840	0.65	0.02	10.8
	4							7.13	3.15	50,173	0.35	0.02	26.71
	5							0	0	124,358	0.44	0.02	21.33

Table 7: Comparison of solution with optimal for small size problems (continued in the next page)

Prob No.	Prob instance	No. of Parts, *N*	No. of M/cs types, *M*	Max No. of Oprn *MaxJ*	Max No. of Alt-mach, *MaxK*	No. of Cells *C*	Cell size, *S*	Optimality gap (%)		Optimal value	Computational time (cpu time in sec)		
								SAA	LS		SAA	LS	Optimal
4		5	5	3	2	2	6						
	1							0	0	127,306	5.72	0.02	83.78
	2							4.34	1.45	109,612	0.35	0.02	56.82
	3							1.03	1.03	174,498	10.68	0.02	10.52
	4							0.28	0.28	101,115	3.65	0.02	5.89
	5							4.17	4.17	95,263	0.42	0.02	47.12
5		6	5	2	2	3	4						
	1							0	0	105,634	0.6	0.01	382.5
	2							0	0	44,339	0.34	0.01	403.4
	3							0	0	109,257	32.44	0.02	185.6
	4							4.69	4.69	61,691	0.38	0.02	421.8
	5							3.60	1.35	133,171	6.25	0.02	372.4

Table 7: Comparison of solution with optimal for small size problems.

α ↓	L_{max} , AT_{max} →		L_{max}=100, AT_{max}=50		L_{max}=100, AT_{max}=50		L_{max}=100, AT_{max}=50	
	Algorithm →		SAA	LS	SAA	LS	SAA	LS
0.97	% improvement in objective function value.	Max	25.97	10.16	27.89	8.36	29.76	7.50
		Min	21.04	8.18	21.53	6.36	22.46	6.31
		Avg	23.02	9.31	24.49	7.48	26.10	6.91
	Computation time (cpu in sec)	Max	41.38	278.58	84.45	236.84	174.58	206.89
		Min	31.32	196.68	60.72	184.34	136.35	139.63
		Avg	35.96	235.69	73.46	204.64	155.52	183.15
0.98	% improvement in objective function value.	Max	26.87	10.68	28.71	8.19	29.43	9.70
		Min	20.82	6.74	23.42	5.21	22.89	4.76
		Avg	23.39	8.34	25.90	6.86	26.14	6.83
	Computation time (cpu in sec)	Max	65.68	251.98	135.73	218.38	251.70	269.82
		Min	42.13	181.36	101.26	164.98	183.31	133.55
		Avg	51.21	218.31	110.66	188.14	213.72	187.12
0.99	% improvement in objective function value.	Max	27.82	8.27	28.87	8.34	30.25	6.73
		Min	20.84	6.69	22.97	6.01	24.29	4.71
		Avg	24.95	7.75	26.10	6.86	27.44	5.52
	Computation time (cpu in sec)	Max	113.03	231.61	227.39	216.20	504.79	190.68
		Min	84.52	198.16	167.48	162.91	365.12	114.69
		Avg	98.92	209.00	194.38	186.23	433.53	154.77

Table 8: Computational results for large size problems.