

# **A Two-Stage Solution Approach Based on Mathematical Programming for Designing Manufacturing Cells**

**Maghsud Solimanpur and Ahad Foroughi**  
**Faculty of Engineering**  
**Urmia University**  
**Urmia, West Azerbaijan Province, Iran**

## **Abstract**

This paper presents a new approach for Cell Formation Problem (CFP) considering operation sequence, alternative processing routes, production volume, processing time, budget limitation and machine cost. The proposed mathematical model consists of two phases. In the first phase, it groups parts into part families based on their similarities and in the next phase, machines are assigned to part families based on bottleneck and budget limitations. The obtained results reveal the effectiveness of the proposed model.

## **Keywords**

Cellular manufacturing, cell formation problem, alternative process routes, integer programming

## **1. Introduction**

Group Technology (GT) is a manufacturing philosophy that identifies similar parts and groups them into part families in order to take advantage of the similarities in manufacturing. Cellular Manufacturing (CM) can be defined as an application of group technology in which functionally dissimilar machines are grouped to produce a family of parts. By dedicating a machine cell to produce a part family, many of the efficiencies of mass production can be realized in a less repetitive batch environment. The major advantages of CM reported in the literature are reduction in setup time, throughput time, work-in-process inventories, material handling costs, etc ([1],[2]). The fundamental problem in the design of cellular manufacturing systems is the identification of part families and the composition of machine cells. Given a set of parts, processing requirements and available resources, the objective of the cell formation problem (CFP) is to obtain a satisfactory decomposition of parts into families and machines into cells such that the resulting system performs well with respect to the relative objectives. Many methods and algorithms have been developed in the literature to solve CFP. There are a number of interesting review papers ([3]-[5]). The general effort of cell formation can be described as below:

- 1) Part family formation
- 2) Machine-cell formation
- 3) Allocating machine-cells to part families

Ballakur and Steudel [6] suggested three solution strategies based on the procedure used to form part families and manufacturing cells. They can be used as a framework to classify existing CM design methods. The three solution strategies are as follows:

- (1). Part families are formed first, and then machines are grouped into cells according to the part families. This is called the part family grouping solution strategy.
- (2). Manufacturing cells are created first based on similarity in part routings, and then the parts are allocated to the cells. This is referred to as the machine grouping solution strategy.
- (3). Part families and manufacturing cells are formed simultaneously. This is the simultaneous machine-part grouping solution strategy.

A part family grouping solution strategy is discussed in this paper including many factors like machine

requirements, machine setup time, machine utilization, workload, alternative routes, machine capacities, operation sequence, production volume and processing time. A great number of the cell formation methods available in the literature have only considered some of mentioned factors. Chu [7] have reviewed the literature and reported that only 10.17 percent of related papers have considered sequence of operations in CFP. When machines are visited by parts in a part family in the same order, a flow-line configuration of machines can be adopted within the corresponding cell to bring the advantages of a product-based layout to the system. Therefore considering the sequence of operations while designing a cellular Manufacturing System (CMS) can improve the productivity of a manufacturing system ([8],[9]). Many researchers also assume that each part can be produced only by a unique process plan. However, in real applications, a part may be producible by alternative processing routes. Considering alternative processing routes provides some additional flexibility in design of cellular manufacturing systems and therefore a lower material flow cost may be achieved. Chu [7] reports that only 20.34 percent of the surveyed papers have considered alternative processing routes. Therefore, a quite lower percent of the studies in literature have addressed both alternative processing routes and sequence of operations.

Kazerooni et al [10] developed a machine chain similarity (MCS) method by considering sequence of operations and alternative processing routes. They divided the main CFP into three sub-problems: selection of routes, grouping machines into cells and grouping parts into part families. They used a genetic algorithm to obtain the best solution of each sub-problem. Zhao and Wu [11] present a multi-objective model which forms manufacturing cells based on production data such as alternative process routings, production sequence, production volume and workload. Yin et al [12] incorporate several production factors into a nonlinear mathematical model and propose a heuristic methodology. Mahesh and Srinivasan [13] proposed two methods, namely the branch and bound technique and a heuristic based on a multistage programming approach for solving CFP.

The purpose of this paper is to solve CFP considering two significant aforementioned factors and achieve a cell formation with minimum intra-cell material movement and inter-cell material movement. The remainder of the paper is organized as follows. A dissimilarity coefficient based on Levenshtein's algorithm [14] is introduced to measure the dissimilarity among routes. Two linear mathematical programming models are proposed to solve CFP. The former determines the route to be selected for each part and forms the part families simultaneously. The latter determines the machines to be allocated to each cell considering processing time, production volume, budget and cost of machines.

## 2. Dissimilarity Coefficient

A processing route is treated as a definite sequence of operations to be done successively to manufacture the corresponding part. At each route, it is assumed that each operation is performed by a certain machine. Thus, a route can be treated as an ordered sequence of machines. In other words, since a machine is denoted by a unique symbol, a processing route can be imagined as an ordered string of symbols. A distance index based on Levenshtein's (1966) algorithm is introduced in this section to measure the dissimilarity between the processing routes. In information theory and computer sciences, the Levenshtein's distance is a metric for measuring the amount of difference between two sequences. The Levenshtein's distance between two strings is given by the minimum number of operations needed to transform a string into another, where an operation is an insertion, deletion, or substitution of a single character. These operations are defined as follows.

Insertion transformation

$$A_1A_2 \rightarrow A_1O_1A_2$$

Deletion transformation

$$A_1O_1A_2 \rightarrow A_1A_2$$

Substitution transformation

$$A_1O_1A_2 \rightarrow A_1O_2A_2$$

The pseudo code for Levenshtein's algorithm that takes two strings, S of length m, and T of length n, and computes the Levenshtein's distance between them is as follows.

```

int LevenshteinDistance(char s[1..m], char t[1..n])
// d is a table with m+1 rows and n+1 columns
declare int d[0..m, 0..n]
for i from 0 to m
  d[i, 0] := i
for j from 0 to n
  d[0, j] := j
for j from 1 to n
  for i from 1 to m {
    if s[i] = t[j] then
      cost := 0
    else
      cost := 1
    d[i, j] := minimum(d[i-1, j] + 1, // insertion
                      d[i, j-1] + 1, // deletion
                      d[i-1, j-1] + cost) // substitution
  }
return d[m, n]

```

---

To show calculation of this index, a problem has been adopted from Sankaran and Kasilingam [15] as given in Table 1. This problem includes ten parts to be processed by six machines and 20 routes in total. In this problem, each machine is denoted by a unique letter. For example, the calculation of dissimilarity between route 1 of part 1 and route 2 of part 8 is shown in Figure 1.

Table 1. An illustrative example adopted from Sankaran and Kasilingam [15]

Part	Production	Route	Machine (Processing time)	Part	Production	Route	Machine (Processing time)		
1	100	1	B(2),E(2),F(1)	5	30	4	C(2),D(1),E(3),		
		2	B(2),C(3),F(1)			6	40	1	B(4),D(2)
2	80	1	B(1),D(1),E(2),F(1)	7	80			2	D(2),F(3)
		2	B(2),D(1),F(1)			8	90	1	A(2),C(1)
3	20	1	A(1),D(4)	9	110			1	B(2),D(1),E(2),F(1)
4	40	1	B(2),D(2),E(2)			10	60	2	A(1),B(2),D(1),F(1)
		2	B(2),D(2),F(3)	10	60			1	B(2),D(1),F(2)
5	30	1	A(1),C(2),D(1)			10	60	2	B(2),E(3),F(2)
		2	A(1),C(2),E(3)	10	60			1	C(2),D(1)
		3	C(2),D(1)					2	D(1),E(4)

		<b>B</b>	<b>E</b>	<b>F</b>
	0	1	2	3
<b>A</b>	1	1	2	3
<b>B</b>	2	1	2	3
<b>D</b>	3	2	2	3
<b>F</b>	4	3	3	2

Figure 1. Calculation of dissimilarity

### 3. Problem Formulation

The proposed solution approach consists of two stages. In the first stage, optimum processing route for each part is decided and the part families are accordingly formed. Machines are assigned to the identified part families in the second stage. A mathematical programming model is proposed at each stage.

#### 3.1 Part Family Formation and Route Selection

The following notations are used to formulate the part family formation and process routing selection problem at this stage.

- $P$  Number of parts,
- $C$  Number of cells,
- $R_i$  Number of routings for part  $i$ ,
- $D_{ikjl}$  Dissimilarity between the route  $k$  of part  $i$  and route  $l$  of part  $j$ ,
- $x_{ikc} = \begin{cases} 1 & \text{if part } i \text{ is to be processed by route } k \text{ in cell } c \\ 0 & \text{Otherwise} \end{cases}$

The processing route of each part is to be selected in such a way that the dissimilarity between the parts in the same family be minimum. Therefore, the following mathematical programming model can be formulated.

#### MODEL 1:

$$\text{Minimize } \sum_{i=1}^{P-1} \sum_{k=1}^{R_i} \sum_{j=i+1}^P \sum_{l=1}^{R_j} \sum_{c=1}^C D_{ikjl} * x_{ikc} * x_{jlc} \quad (1)$$

s. t.

$$\sum_{k=1}^{R_i} \sum_{c=1}^C x_{ikc} = 1 \quad \forall i = 1, 2, 3, \dots, P \quad (2)$$

$$x_{ikc} \in \{0, 1\} \quad \forall i, k, j \quad (3)$$

It is worth noting that the number of cells should be set by designer before running the model. Since the objective function expressed by (1) is non-linear, the following variable is introduced to linearize MODEL 1.

$$y_{ikjlc} = x_{ikc} * x_{jlc}$$

Where

$$y_{ikjlc} = \begin{cases} 1 & \text{if parts } i \text{ and } j \text{ are processed by routes } k \text{ and } l \text{ in cell } c, \text{ respectively} \\ 0 & \text{Otherwise} \end{cases}$$

Thus, the following mixed-integer linear programming model is derived.

#### MODEL 2:

$$\text{Minimize } \sum_{i=1}^{P-1} \sum_{k=1}^{R_i} \sum_{j=i+1}^P \sum_{l=1}^{R_j} \sum_{c=1}^C D_{ikjl} * y_{ikjlc} \quad (4)$$

ST:

$$\sum_{k=1}^{R_i} \sum_{c=1}^C x_{ikc} = 1 \quad \forall i = 1, 2, 3, \dots, P \quad (5)$$

$$y_{ikjlc} \geq x_{ikc} + x_{jlc} - 1 \quad \forall i, k, j, l, c; i \neq j \quad (6)$$

$$x_{ikc} \in \{0,1\} \quad \forall i, k, c \quad (7)$$

$$y_{ikjlc} \geq 0 \quad \forall i, k, j, l, c \quad (8)$$

The objective function minimizes the total dissimilarity between the processing routes of the parts being in the same cell. Constraint (5) ensures that each part will be processed by only one route at only one cell. Constraint (6) establishes the relationship between variables  $x$  and  $y$  and guarantees that variable  $y_{ikjlc} = 1$  if  $x_{ikc} = 1$  and  $x_{jlc} = 1$ , and it will be zero in otherwise. This is because of the fact that the coefficients of variables  $y$  in the objective function are positive.

The proposed linear model (MODEL 2) was coded in Lingo 8.0 Software and run on a Pentium IV PC with 2.0 GHz speed and 2 GB of RAM. This code was used to solve the illustrative problem given in Section 2 to test validity of MODEL 2. The final globally optimum solution reported in Table 2 was obtained using a branch-and-bound algorithm after 3 seconds.

Table 2. The result of optimum routings and part family formation

Part	Part family(1)							Part family (2)		
	1	2	4	6	8	9	10	3	5	7
Route	1	2	2	1	1	1	1	1	3	1

### 3.2 Machine Assignment

After selection of processing routes and determination of part families, the machines are to be assigned to manufacturing cells. The set of machines can be partitioned into two categories, namely non-bottleneck and bottleneck. A non-bottleneck machine is a machine that is required by only one part family whereas a bottleneck machine is needed by more than one part family. A non-bottleneck machine is assigned to the part family which needs this machine. However, since bottleneck machines are required by several part families, assignment of these machines is controversial to some extent. A mathematical programming model is proposed in this section to assign this type of machines.

To assign a bottleneck machine to a part family, the proposed method considers the potential benefit of this assignment with respect to the processing times and production volumes. If bottleneck machine  $m$  is not assigned to cell  $c$ , it will induce the cost  $w_{mc}$  expressed by

$$w_{mc} = \sum_{p \in c} D_p * t_{mp} * A_p \quad \forall c, \forall m \in B. \quad (9)$$

Where

$D_p$  is the production volume of part  $p$ ,

$t_{mp}$  is the processing time of part  $p$  on machine  $m$ ,

$A_p$  is the average cost of one inter-cell movement of part  $p$ , and

$B$  is the set of bottleneck machines.

Let  $r_m$  denote the procurement cost of machine  $m$ . Therefore, the saving induced by the assignment of machine  $m$  to cell  $c$ ,  $v_{mc}$ , can be calculated by

$$v_{mc} = w_{mc} - r_m \quad (10)$$

Based on the solution obtained from MODEL 2 for the illustrative example, machines C and D are bottleneck. Table 3 shows the procurement cost of each machine in the illustrative example. Table 4 shows the potential saving due to assignment of each bottleneck machine to each cell. We assume that  $A_p$  is the same for all parts.

**Table 3. Procurement cost of machines in the illustrative example**

Machine	A	B	C	D	E	F
Cost (\$)	40	20	30	90	75	35

Table 4. Potential saving due to assignment of bottleneck machines to different cells

	Bottleneck machine	
	C	D
Part family 1	90	410
Part family 2	110	20

We propose a linear 0-1 programming model for cost-effective assignment of bottleneck machines to the part families formed by MODEL 2. The proposed model allows duplication of bottleneck machines with considering budget limitation. Let  $z_m$  denote the number of bottleneck machine  $m$  to be procured and  $u_{mc}$  is a binary decision variable which is 1, if machine  $m$  is assigned to cell  $c$  and 0, otherwise. The proposed model is expressed as follows.

**MODEL 3:**

$$\text{Maximize } \sum_{\forall m \in B} \sum_{c=1}^c v_{mc} * u_{mc} \quad (11)$$

ST:

$$z_m \geq 1 \quad \forall m \in B \quad (12)$$

$$\sum_{\forall m \in B} r_m * z_m \leq \text{Budget} \quad (13)$$

$$\sum_{c=1}^c u_{mc} = z_m \quad \forall m \in B \quad (14)$$

$$u_{mc} \in \{0,1\} \quad (15)$$

The objective function expressed by (11) maximizes the total saving in the assignment of bottleneck machines. Constraint (12) ensures that at least one bottleneck machine of type  $m$  should be assigned. Constraint (13) satisfies the budget limitation and Constraint (14) enumerates the number of machines of type  $m$  assigned to different cells. Suppose the amount of budget is 50 units in the illustrative example presented in Section 2. The MODEL 3 was solved using LINGO 8 within less than hundredth of a second and resulted in the solution shown in Table 5. In this table, the machine shown in boldface (machine C) has been duplicated. Machine D is still a bottleneck machine and parts 3 and 5 which need this machine have to visit cell 1 for completion.

Table 5. The final optimum result

Part	Part family(1)							Part family (2)		
	1	2	4	6	8	9	10	3	5	7
route	1	2	2	1	1	1	1	1	3	1
Machine	B	E	F	D	C			C	A	

#### 4. Conclusion

A few papers simultaneously have considered operation sequence, production volume, alternative process routings, processing time, budget limitation and machine cost in the design of cellular manufacturing systems. Modeling of these factors makes the CFP complex but more realistic. In this paper, a new approach is proposed for solving CFP which minimizes the sum of dissimilarity and maximizes the total saving in the assignment of bottleneck machines. The proposed approach solves the problem in two phases. Because this approach considers most production factors in CFP, it can be significant and leads to efficient cell formation, better job routing, scheduling, process control, flexible inter-cell layout and makespan of jobs. Decomposition of the proposed comprehensive approach into two stages (each stage with a smaller problem) reduces the computation time and makes it efficient to tackle large-sized real problems. In the future studies, this model can be extended by consideration of capacity of machines. Also, application of meta-heuristic algorithms can be attempted in subsequent researches.

#### References

1. Heragu, S. S., 1994, "Group technology and cellular manufacturing". *IEEE Transactions on Systems, Man and Cybernetics*, 24(2), 203–214.
2. Wemmerlov, U., and Hyer, N., 1989, "Cellular Manufacturing in the US Industry: A Survey of Users". *International Journal of Production Research*, 27(9), 1511-1530.
3. King, J. R., 1980, "Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm". *International Journal of Production Research*, 18, 213–232.
4. Singh, N., 1993, "Design of cellular manufacturing systems: An invited review". *European Journal of Operational Research*, 69, 284–291.
5. Selim, H. M., Askin, R. G., & Vakharia, A. J., 1998, "Cell formation in group technology: Review, evaluation and directions for future research". *Computers and Industrial Engineering*, 34, 3–20.
6. Ballakur, A., and Steudel, H., 1987, "A Within-Cell Utilization Based Heuristic for Designing Cellular Manufacturing Systems". *International Journal of Production Research*, 25(5), 639-665.
7. Chu, C., 1995, "Recent Advances in Mathematical Programming for Cell Formation". In: Kamrani A.K, Parsaei H.R, and Liles D.H, "Planning, Design, and Analysis of Cellular Manufacturing Systems". Amsterdam, NL: Elsevier, 18.
8. Choobineh F., 1988, "A framework for the design of cellular manufacturing systems". *International Journal of Production Research*, 26(7), 1161–72.
9. Tam KY., 1988, "An operation sequence based similarity coefficient for part family formations". *Journal of Manufacturing Systems*, 9, 55–68.
10. Kazerooni M., Luong HS., Abhary K., 1997, "A genetic algorithm based cell design considering alternative routing". *Computer Integrated Manufacturing Systems*, 10(2), 93–108.
11. Zhao C., 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.
12. Yin Y., Yasuda K., Hu L., 2005, "Formation of manufacturing cells based on material flows". *International Journal of Advanced Manufacturing Technology*, 27, 159–165.
13. Mahesh O., Srinivasan. 2002, "Incremental cell formation considering alternative machines". *International Journal of Production Research*, 40(14), 3291–3310.
14. Levenshtein V.I., 1996, "Binary codes capable of correcting deletions, insertions, and reversals". *Soviet Physics Doklady* 10,707–710.
15. Sankaran S., Kasilingam KG., 1990, "An integrated approach to cell formation and part routing in group technology". *Engineering Optimization*, 16, 235–245.