經營科學
第16卷 第2號
1999年 11月

Mathematical Programming Approaches to GT Cell Formation: A Comparative Study

Youkyung Won*

■ Abstract ■

This paper compares and evaluates the performances of the two types of mathematical programming models for solving the machine-part cell formation problem in group technology manufacturing : indirect formulation relying on surrogate measure such as similarity coefficient and direct formulation seeking to minimize the number of exceptional elements. New indirect formulation, called the generalized p-median model, is proposed. Unlike existing p-median formulations, proposed formulation includes the classical cell formation problem in which only one process plan exsits for each part as a special case. The proposed new formulation can also deal with the cell formation problem in which alternative process plans exist for a part. The indirect formulation is compared with a new direct formulation which needs much fewer extra variables and constraints than existing direct formulations. Some significant findings from comparative experiment are discussed.

1. Introduction

Group Technology (GT) has been accepted as an effective approach for improving the productivity of batch-type manufacturing system in which many different products having relatively low volumes are produced in small lot sizes. The benefits from applying GT to manufacturing are summarized in Burbidge [6]. To exploit the benefits of GT, parts are grouped into families and machines into cells, so that a family of parts can be produced completely within a cell of machines. The problem of finding part families and machine cells in GT manufacturing systems is known as the cell formation problem in literature.

The main input to the cell formation problem is the *machine-part incidence matrix*. The machine-part incidence matrix is a binary matrix **A** where the element

$$a_{ij}=\begin{cases}1 & \text{if part } j \text{ is processed by machine } i,\\ 0 & \text{otherwise.}\end{cases}$$

* School of Business Administration, Jeonju University

The objective of the cell formation is to create mutually separable machine cells so that the cells can operate independently with minimum interaction. The best block-diagonal structure from the incidence matrix means the best cells configuration with minimum intercellular part moves. However, the cell formation process often identifies exceptional elements which prevent from forming independent machine cells by creating interactions between cells. Exceptional elements are the results of bottleneck machines that are needed to process a large number of parts found in two or more part families, or bottleneck parts that require processing on machines assigned to two or more machine cells. Most of numerous studies on the cell formation are concerned with eliminating or minimizing the exceptional elements.

Many researchers have addressed the cell formation problem and proposed numerous methods for grouping machines and parts. Cheng [11], Chu [12], and Selim *et al.* [36] provide extensive classifications and reviews of the cell formation literature. However, most of the existing approaches to the cell formation problem try to extend the principles of clustering analysis. Many heuristic clustering approaches usually use similarity coefficient defined between machines or parts pair to cluster the machines and parts. Various similarity coefficients have been suggested in literature [7, 14, 25, 27, 28, 31, 35, and 44].

Mathematical programming approaches try to find the cells and families by formulating the problem into linear or nonlinear integer programming models. Kusiak [25] suggested linear integer programming models called the p-median model seeking to maximize the sum of similarity coefficients defined between pairs of parts. The author proposed two separate p-median formulations for solving the part family formation problem with a single process plan for a part and the one with alternative process plans for a part. Other formulations to the problem include the models by Kumar *et al.* [23], Srinivasan *et al.* [37], and Kusiak *et al.* [26].

However, all the above-mentioned approaches to the cell formation problem are indirect approaches in the sense that the models use indirect measures such as similarity coefficients to formulate the problem [42]. In general, there is no explicit relationship between the similarity score and the number of exceptional elements which is a direct measure of inter cellular moves of parts [43]. Boctor [3, 4] first proposed a linear integer programming model minimizing the total number of exceptional elements directly and suggested its variants. Viswanathan [42] and Adil *et al.* [1] presented quadratic integer programming models minimizing the weighted sum of the total number of exceptional elements outside the clusters and the total number of zeros within all the clusters known as voids so as to attain the minimum inter-cell movements of parts and the maximum within-cell utilization of machines simultaneously.

Apart from developing the models or algorithms for solving the cell formation problem, some authors [13, 19, 29, and 40] have attempted to compare and evaluate the effectiveness and efficiency of the methods proposed so far. However, most of the existing studies compare the performances of heuristic clustering or array-based methods. Little researches have been devoted to the performance evaluation of the mathematical models for the cell formation. One of the main reasons for this is that most of mathematical models treating the cell formation require many binary decision variables. In addition, large number of extra continuous variables and constraints are often needed to linearize nonlinear terms. Large mathematical model containing many variables and constraints takes prohibitive computation time to implement the models directly even on main frame computer. According to Kaparthi and Suresh [19], mathematical programming approaches have little applicability to large size problems, for which heuristic approaches are preferred. But according to Zhu *et al.* [45], mathematical programming approaches have much applicability to medium size problems. Recently, Wang and Roze [43] compared the performances of the p-median mathematical models based on various types of similarity coefficients. The authors proposed a modified p-median formulation with much fewer constraints compared with Kusiak's original p-median formulation.

However, the authors' work has some limitations. First, their new p-median formulation can't deal with the cell formation in which alternative process plans exist for a part although the new formulation requires very few binary variables and constraints compared with Kusiak's p-median formulation. In addition, the authors do not propose any mathematical models with which we can compare the performance of the p-median model. Furthermore, the authors use only three types of data sets taken from the literature to evaluate the performance of the p-median model.

The paper is aimed at comparing and evaluating the performances of indirect mathematical formulation and direct mathematical formulation to the cell formation problem. Section 2 proposes a new indirect formulation, called the generalized p-median model, which uses the new generalized machine similarity coefficient to include the cell formation problem with a single process plan for a part as a special case. Section 3 presents a quadratic integer programming model minimizing the sum of exceptional elements directly. The quadratic integer programming model is linearized with minimal extra variables and constraints as compared with existing direct formulations. Section 4 provides performance comparison using wide range of data sets taken from the literature and the last section summarizes the conclusion.

2. Indirect formulation : generalized p-median model

Kusiak's original p-median model [25] uses the similarity coefficient defined between process plans of parts to formulate the part family formation problem. This results in very large integer linear programming model since the number of parts included in a cell formation problem is usually much more than the number of machines. Therefore, similarity coefficient defined between machines instead of process plans of parts yields much smaller integer programming model that can be solved within moderate computer runtime. This modification has been suggested by Wang and Roze. In this section, a new p-median model which is a generalization over Wang and Roze's model as well as Kusiak's model is constructed. The new p-median model use generalized machine similarity coefficient to formulate the machine cell formation problem.

New similarity coefficient between two machines h and i is defined by

$$s^*_{hi} = \begin{cases} \sum_{j=1}^{n} \sigma(h, i, j) & \text{if } h \neq i,\ i = 1, \ldots, m \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where

s^*_{hi} = generalized similarity coefficient between machines h and i

m = number of machines

n = number of parts

$$\sigma(h,i,j)=\begin{cases}1, & \text{if } a_{hjr}=a_{ijr}=1 \text{ for some } r\in R_j\\ 0, & \text{otherwise.}\end{cases}$$

$$a_{ijr}=\begin{cases}1, & \text{if part } j \text{ requires processing on machine } i \text{ in plan } r\\ 0, & \text{otherwise.}\end{cases}$$

R_j = set of process plans of part j.

From the above definition, $\sigma(h,i,j)=1$ indicates that if some process plan of part j uses both machines h and i the number of common parts processed by those machines is counted as one for that part even though remaining process plans of part j also use both machines. This idea follows from the basic assumption of the cell formation problem with alternative process plans that in final solution only one process plan is selected for each part. We can then use the generalized machine similarity coefficient to deal with not only the cell formation problem in which only a fixed process plan exists for a part but also the cell formation problem in which alternative process plans exist for a part. The term *generalized* is used in this context.

In order to construct the new p-median model using the generalized machine similarity coefficient, define the variable

$$x_{hi}=\begin{cases}1 & \text{if machine } h \text{ belongs to cell } i,\ h,\ i=1,\dots,m\\ 0 & \text{otherwise.}\end{cases}$$

The generalized p-median model can then be stated as

(GP)

$$\text{Max}\ \sum_{h=1}^{m}\sum_{i=1}^{m} s^*_{hi}x_{hi} \qquad (2)$$

$$\text{s.t.}\ \sum_{i=1}^{m} x_{hi}=1,\ h=1,\dots,m \qquad (3)$$

$$\sum_{h=1}^{m} x_{hi} \geq L_c x_{ii},\ i=1,\dots,m \qquad (4)$$

$$\sum_{h=1}^{m} x_{hi} \leq U_c x_{ii},\ i=1,\dots,m \qquad (5)$$

$$\sum_{i=1}^{m} x_{ii}=p \qquad (6)$$

$$x_{hi}=0 \text{ or } 1,\ h,\ i=1,\dots,m. \qquad (7)$$

The objective is to maximize the sum of machine similarities. Constraint (3) ensures that each machine belongs to exactly one machine cell. In constraints (4) and (5) L_c and U_c represent the minimum and the maximum numbers of machines allowable to each cell, respectively. The lower cell size restriction is added to avoid singleton machine cells. Note that in constraint (4) at least L_c machines can be clustered with machine i only when $x_{ii}=1$. Similarly, constraint (5) ensures that at most U_c machines are clustered with machine i only when $x_{ii}=1$. Constraint (6) specifies the required number of machine cells. Constraint (7) ensures the binary solution.

The generalized p-median model (GP) is much smaller problem than the original p-median model in terms of the number of constraints as well as the number of binary variables. The original p-median model based on the similarity coefficient between process plans of parts contains $\left(\sum_{j=1}^{n}|R_j|\right)^2$ binary variables, whereas the generalized p-median model contains only m^2 binary variables. This leads to a significant reduction of binary variables for large-scale cell formation problems containing large number of parts. Moreover, the model (GP) contains only a total of $3m+1$ constraints which is very small compared with $\left(\sum_{j=1}^{n}|R_j|\right)^2+\left(\sum_{j=1}^{n}|R_j|\right)+1$ cons traints in the original p-median model. Wang and Roze's formulation does not contain the lower cell size constraint (4) that is needed to avoid singleton machine cells. Therefore, the generalized p-median model (GP) contains the same number of binary variables and constraints as in Wang and Roze's formulation if the constraint (4) is omitted from the formulation.

3. Direct formulation

3.1. Quadratic integer programming model

In this section a quadratic integer programming model is developed for simultaneous grouping of machines and parts. The proposed quadratic integer programming model is aimed at minimizing the sum of exceptional elements. In order to develop new direct formulation for minimizing the sum of exceptional elements, define the following binary variables :

$$x_{ik}=\begin{cases} 1 & \text{if machine } i \text{ belongs to cell } k \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{jl}=\begin{cases} 1 & \text{if part } j \text{ is assigned to cell (family) } l \\ 0 & \text{otherwise.} \end{cases}$$

Assuming that under a given machine-part incidence matrix $\mathbf{A} = (a_{ij})$ machine i processes part j, an entry $a_{ij}=1$ becomes an exceptional element if and only if machine i and part j belong to mutually different cells, i.e., $x_{ik}=y_{jl}=1$ for $k \neq l$. Therefore, the sum of exceptional elements is given by

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1, l\neq k}^{p} a_{ij}x_{ik}y_{jl} \quad (8)$$

Then, the model which seeks to minimize the sum of exceptional elements is as follows :

(Q0) Min (8)

$$\text{s.t. } \sum_{k=1}^{p} x_{ik}=1, \ i=1, \ldots, m \quad (9)$$

$$L_c \leq \sum_{i=1}^{m} x_{ik} \leq U_c, \ k=1, \ldots, p \quad (10)$$

$$\sum_{l=1}^{p} y_{jl}=1, \ j=1, \ldots, n \quad (11)$$

$$L_f \leq \sum_{j=1}^{n} y_{jl} \leq U_f, \ l=1, \ldots, p \quad (12)$$

$$x_{ik}, y_{jl}=0 \text{ or } 1 \text{ for all } i, j, k, l. \quad (13)$$

The objective function (8) is equivalent to the one given in the Boctor model. Constraint (9) indicates that each machine is assigned to exactly one cell. Constraints (10) are necessary to satisfy the upper and lower limits on cell size. Constraint (11) ensures that each part is also allocated to one and only one cell (family). Constraints (12) impose the upper and lower limits on family size. L_f and U_f represent the minimum and the maximum numbers of parts assigned to each cell (family), respectively. Constraint (13) guarantees the binary solution.

The family size constraint set is added to prevent abnormally small or large families from being formed. This leads to producing more compact cells and families compared with the generalized p-median model (GP) which contains only the cell size restriction. Existing approaches to direct formulation except Boctor's formulation use subjective weighting factor to minimize the weighted sum of voids and exceptional elements [1, 42]. In those models, changing weights are selected by the cell designer. As a result of this, however, the final objective value of the models does not represent the actual inter cellular moves of parts. In contrast, the objective value of the new formulation really represents the inter cell part moves.

Relative to the cell and family size constraints, Boctor's formulations include only cell size constraint, but the model (Q0) includes additional restriction on the family size. Therefore, the quadratic model (Q0) is entirely equivalent to Boctor's formulations adjusted with family size constraint.

3.2 Efficient linearization of (Q0)

Quadratic terms in the objective function of (Q0) can be linearized by using Oral and Kettani's efficient linearization technique [33]. According to the authors' technique, the quadratic term in (Q0) can be linearized with extra continuous variable z_{ik} as follows :

(QL)

$$\text{Min} \ \sum_{i=1}^{m} \sum_{k=1}^{p} z_{ik} \tag{14}$$

s.t. (9)~(13) and

$$z_{ik} \geq \sum_{j=1}^{n} \sum_{l \neq k}^{p} a_{ij} y_{jl} - \left(\sum_{j=1}^{n} a_{ij} \right)(1 - x_{ik}), \quad i=1, \ldots, m; \ k=1, \ldots, p \tag{15}$$

$$z_{ik} \geq 0, \ i=1, \ldots, m; \ k=1, \ldots, p. \tag{16}$$

Then the linear formulation (QL) is equivalent to the quadratic integer formulation (Q0) and hence equivalent to Boctor's formulation adjusted with family size constraint.. Note that the integrality of variable z_{ik} is guaranteed since in constraint (15) the right-hand part of the inequality is always integral and all the coefficients of z_{ik} in the objective function are 1. The variable z_{ik} indicates the number of exceptional elements generated by machine i assigned to cell k. The linear model (QL)contains only mp extra continuous variables and the same number of extra constraints. Recalling that $p \ll m \ll n$ in a typical cell formation problem, computational efforts required additionally are maintained minimal in terms of the model size.

The model (QL) is compared with Boctor's 1996 formulation. Both models have the same number of binary variables. While the Boctor model contains $p|A|$ continuous variables and $m+n+2p+p(m+|A|)$ constraints, the model (QL) contains pm continuous variables and $m+n+2p+pm$ constraints. Note that the number of extra continuous variables and constraints needed by the model (QL) are very small compared with those needed by Boctor's formulations since $pm \ll p|A|$. In order to solve the problem in Chandrasekharan and Rajagopalan [8] with $m=40$, $n=100$, $p=10$ and $|A|=422$, for example, Boctor's 1996 formulation needs 4220 continuous variables and 4780 constraints, whereas the proposed formulation needs no more than 400 continuous variables and 560 constraints.

4. Comparative experiment

In this section, the performances of both the indirect mathematical formulation and the direct mathematical formulation are compared. Although several measures have been presented for evaluating the performance of cell formation methods, computation time and the number of exceptional elements which have been considered to be the most important by several authors are selected as the criteria of comparison.

While the direct formulation presented in the previous section can find the cells and families simultaneously, the p-median model requires additional procedure assigning the parts to the corresponding machine cells once the cells are

〈Table 1〉 Computational result for the generalized p-median model and the new direct formulation

Problem source	Problem size	p	U_c	U_f	CPU (seconds)		# of exceptions	
					GP	QL	GP	QL
1. Carrie(1973a)[7]	18×24	3	7	10	482.56	*	20	18
2. Carrie(1973b)[7]	20×35	4	6	10	6.32	25.45	2	2
3. De Witte(1980)[14]	12×19	2	9	12	4.13	2.08	9	9
4. King(1980a)[20]	14×24	3	6	12	52.01	0.57	4	0
		4	5	10	13.67	98.64	7	2
5. King(1980b)[21]	16×43	3	7	20	11.07	317.74	28	17
		4	6	17	14.79	*	30	40
6. King(1982)[22]	30×90	3	12	50	*	*	71	81
7. Mosier(1985)[31]	20×20	2	14	14	1.92	594.33	31	23
8. Stanfel(1985a)[38]	14×24	4	5	10	5.48	60.72	6	2
9. Stanfel(1985b)[38]	30×50	4	15	20	*	5.92	12	0
		5	18	25	*	*	24	26
10. Stanfel(1985c)[38]	30×50	3	18	25	*	*	40	35
		4	15	20	*	*	46	41
11. Kumar(1986)[23]	23×20	2	15	15	922.72	320.97	26	13
		3	10	10	0.78	*	34	40
12. Kumar(1987)[24]	30×41	2	20	30	2.47	46.19	13	3
		3	14	20	*	*	11	5
		4	12	15	*	*	10	17
		5	10	10	*	*	10	29
13. Chandrasekharan(1987)[8]	40×100	8	12	30	*	*	117	99
		9	9	25	*	*	70	202
		10	6	20	*	*	43	164
14. Tabucanon(1987)[39]	30×40	4	10	14	*	*	21	39
		5	8	12	*	*	24	37
15. Chandrasekharan(1989a)[9]	24×40	7	5	8	57.14	35.08	0	0
16. Chandrasekharan(1989b)[9]	24×40	7	5	8	2.76	*	10	20
17. Chandrasekharan(1989c)[9]	24×40	7	5	8	1.87	*	20	61
18. Chandrasekharan(1989d)[9]	24×40	7	5	8	1.86	*	20	61
19. Chandrasekharan(1989e)[9]	24×40	6	6	12	3514.51	*	47	52
20. Chandrasekharan(1989f)[9]	24×40	6	7	12	266.75	*	54	58
21. Chandrasekharan(1989g)[9]	24×40	5	8	12	74.79	*	53	59
22. Seifoddini(1989a)[34]	5×20	2	3	13	0.07	0.65	7	5
23. Seifoddini(1989b)[35]	11×22	2	7	14	6.82	0.61	5	5
		3	4	10	0.43	15.34	10	10
24. Harhalakis(1990)[16]	20×20	4	7	8	41.71	*	11	29
		5	6	6	6.38	*	14	31
25. Srinivasan(1990)[37]	16×30	3	7	14	0.55	876.83	16	16
		4	6	10	0.40	*	19	19
26. Ventura(1990)[41]	27×27	2	17	17	15.72	178.44	23	23
27 Askin(1991)[2]	14×24	3	6	12	6.86	0.51	4	0
		4	4	10	4.85	72.69	6	2
28. Boe(1991)[5]	20×35	3	10	20	8.66	*	24	24
		4	5	12	0.77	*	35	69
29 Kao(1991)[18]	24×30	7	5	6	342.78	*	12	41
30. Moon(1992)[30]	12×19	4	4	6	2.96	*	11	11
31. Chen(1995)[10]	20×60	4	10	30	445.52	*	25	31
		5	5	15	4.86	*	30	42
32. Gindy(1995)[15]	45×120	8	15	40	*	*	40	89
		9	12	30	*	*	48	60
		10	9	20	*	*	58	102
33. Joines(1996)[17]	20×35	3	10	20	711.51	20.66	2	1
		4	5	10	1.24	*	2	42
34. Nair(1996)[32]	46×100	8	15	40	*	*	149	168
		9	10	30	*	*	181	210
		10	6	20	*	*	151	253

Note : * indicates that the problem takes more than 3600 CPU seconds to solve optimally.

<Table 2> Contingency table showing comparative advantage of both formulations

CPU time	# of exceptional elements: p-median formulation ‡	# of exceptional elements: direct formulation	sum
p-median formulation †	36	12	48
direct formulation	3	5	8
sum	39	17	56

Note : † indicates the category in which the p-median formulation produces the solutions that are equal to or better than the direct formulation in terms of the computation time.
‡ indicates the category in which the p-median formulation produces the solutions that are equal to or better than the direct formulation in terms of the number of exceptional elements.

obtained. Unlike the objective function value of the direct formulation, the objective function value of the p-median model provides no information on the number of exceptional elements. In order to minimize the sum of exceptional elements, a part is assigned to the cell in which that part needs most operations.

In order to evaluate the performance of both mathematical models, 34 medium-sized incidence matrices are taken from the literature. <Table 1> shows the list of the data set selected. In solving the models, the lower limits to the cell and family sizes are set equal to 2. For each formulation, 56 optimization problems have been solved using CPLEX version 2.1 on an HP 9000/715 workstation. Time limit to each problem is set equal to 3600 CPU seconds. For each formulation, CPU time in seconds and the number of exceptional elements are reported in <Table 1>. Contingency <Table 2> summarizes comparative advantage of both formulations in terms of the CPU time and the number of exceptional elements.

From the result shown in <Table 1> and <Table 2> some interesting findings deserve to be mentioned. First, it can be noticed from <Table 2> that in terms of computation time the p-median formulation obtained the solutions that were equal to or better than the direct formulation for 48 problems (about 86%) out of 56 problems and in terms of the number of exceptional elements the p-median formulation obtained the solutions that were equal to or better than the direct formulation for 39 problems (about 70%) out of 56 problems. It can also be noticed that only for 5 problems (about 9%) out of 56 problems the direct formulation produced the solutions that are strictly better than the p-median formulation in terms of both the computation time and the solution quality. This implies usefulness of the generalized p-median mathematical model for medium-sized cell formation problems.

Second, as the value of p increases the computation time of direct formulation increases drastically since the number of binary variables, $(m+n)p$, increases in proportion to the value of p. However, as the value of p increases the p-median formulation took decreasing computation time for 9 problems except 2 problems (problems 5 and 12) taking increasing computation time. Noting that the number of binary variables in the p-median model does not increase even if the value of p increases, this result gives very interesting implication in implementing the p-median model. It has been commonly recommended that in implementing the p-median model, cell designer should start with a small value of p and resolve the model for increasing values of p until satisfactory solution is found [24, 36]. But the result from comparative experiment presented in this paper leads to a policy contrary to such existing recommendation. New recommendation for implementation of the p-median model is that cell designer should start with a sufficiently large value of p to find rough cells quickly and resolve the model for decreasing values of p until satisfactory solution is found.

5. Concluding remarks

This paper compares the performances of indirect mathematical formulation and direct mathematical formulation for solving the cell formation problem in group technology. New indirect formulation, called the generalized p-median model, based on the machine similarity coefficient is proposed to deal with the cell formation problem in which alternative process plans exist for a part as well as the cell formation problem in which only a fixed process plan exists for a part. In addition, new direct formulation is developed which contains much fewer variables and constraints as compared with existing direct formulation.

The results from comparative experiment show two significant findings : first, the generalized p-median formulation is of better applicability to medium-sized cell formation problem than direct formulation and second, on the contrary to existing policy which attempts to resolve the model for increasing values of p starting from a small value of p until satisfactory solution is found, cell designer is recommended to resolve the model for decreasing values of p starting from a sufficiently large value of p.

The direct formulation proposed in the paper, however, gives worse solutions than the generalized p-median formulation in terms of the number of exceptional elements and comuptation time for most problems since its linear model still contains many binary variables. Development of efficient linear model containing fewer binary variables remains as another future research area.

REFERENCES

[1] Adil, G. K., Rajamani, D. and Strong, D. "Assignment allocation and simulated annealing algorithms for cell formation," *IIE Trans.*, 29(1997), pp.53-67.

[2] Askin, R. G., Gresswell, S. H., Goldberg, J. B. and Vakharia, A. J. "A Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing," *Int. J. Prod. Res.*, 29(1991), pp.1081-1100.

[3] Boctor, F. "A linear formulation of the machine-part cell formation problem," *Int. J. Prod. Res.*, 29(1991), pp.343-356.

[4] Boctor, F. "The minimum cost, machine-part cell formation problem," *Int. J. Prod. Res.*, 34(1996), pp.1045-1063.

[5] Boe, W. J. and Cheng, C. H. "A close neighbour algorithm for designing cellular manufacturing systems," *Int. J. Prod. Res.*, 29(1991), pp.2097-2116.

[6] Burbidge, J. L. "Production flow analysis for planning group technology," *J. Opns. Mgmt.*, 10(1991), pp.5-27.

[7] Carrie, A. S. "Numerical taxonomy applied to group technology and plant layout," *Int. J. Prod. Res.*, 11(1973), pp.399-416.

[8] Chandrasekharan, M. P. and Rajagopalan, R. "ZODIAK-an algorithm for concurrent for-

mation of part-families and machine cells," *Int. J. Prod. Res.*, 25(1987), pp.835-850.

[9] Chandrasekharan, M. P. and Rajagopalan, R. "Groupability : an analysis of the properties of binary data matrices for group technology," *Int. J. Prod. Res.* 27(1989), pp. 1035-1052.

[10] Chen, C. L., Cotruvo, N. A. and Baek, W. "A simulated annealing solution to the cell formation problem," *Int. J. Prod. Res.*, 33 (1995), pp.2601-2614.

[11] Cheng, C. H. "Algorithms for grouping machine groups in group technology," *Omega*, 20(1992), pp.493-501.

[12] Chu, C. H. "Recent advances in mathematical programming for cell formation," *Planning, Design, and Analysis of Cellular Manufacturing Systems*, A. K. Kamrani, H. R. Parsaei and D. H. Liles(ed). Elsevier ScienceB. V. Amsterdam, 1995, pp.1-46.

[13] Chu, C. H. and Tsai, M. "A comparison of three array-based clustering techniques for manufacturing cell formation," *Int. J. Prod. Res.*, 28(1990), pp.1417-1433.

[14] De Witte, J. "The use of similarity coefficients in production flow analysis," *Int. J. Prod. Res.*, 18(1980), pp.504-514.

[15] Gindy, N. N. Z., Ratchev, T. M. and Case, K. "Component grouping for GT applications—a fuzzy clustering approach with validity measure," *Int. J. Prod. Res.*, 33(1995), pp.2493-2509.

[16] Harlarakis, G., Nagi, R. and Proth, J. M. "An efficient heuristic in manufacturing cell formation for group technology applications," *Int. J. Prod. Res.*, 28(1990), pp.185-198.

[17] Joines, J. A., Culbreth, C T. and King, R. E. "Manufacturing cell design : an integer programming model employing genetic algorithms," *IIE Trans.*, 28(1996), pp.69-85.

[18] Kao, Y. and Moon, Y. B. "A unified group technology implementation using the back-propagation learning rule of neural networks," *Computers. Ind. Engng.*, 20(1991), pp.425-437.

[19] Kaparthi, S. and Suresh, N. C. "Performance of selected part-machine grouping techniques for data sets of wide ranging sizes and imperfection," *Dec. Sci.*, 25(1994), pp. 515-539.

[20] King, J. R. "Machine-component group formation in group technology," *Omega*, 8 (1990), pp.193-199.

[21] King, J. R. "Machine-component grouping in production flow analysis : an approach using rank order clustering," *Int. J. Prod. Res.*, 18(1980), pp.213-232.

[22] King, J. R. and Nakornchai, V. "Machine-component group for mationin group technology-review and extension," *Int. J. Prod. Res.*, 20(1982), pp.117-133.

[23] Kumar, K. R., Kusiak, A. and Vannelli, A. "Grouping of parts and components in flexible manufacturing systems," *Eur. J. Opl. Res.*, 24(1986), pp.387-397.

[24] Kumar, K. R. and Vannelli, A. "Strategic subcontracting for efficient disaggregated manufacturing," *Int. J. Prod. Res.*, 25(1987), pp.1715-1728.

[25] Kusiak, A. "The generalized group technology concept," *Int. J. Prod. Res.*, 25(1987), pp.561-569.

[26] Kusiak, A., Boe, W. J. and Cheng, C. H. "Designing cellular manufacturing systems: branch-and-bound and $\mathbf{A}^*$ approaches," *IIE Trans.*, 25(1993), pp.46-56.

[27] Kusiak, A and Cho, M. "Similarity coefficient algorithms for solving the group technology problem," *Int. J. Prod. Res.*,

30(1992), pp.2633-2646.

[28] McAuley, J. "Machine grouping for efficient production," *Prod. Engnr.*, 52(1972), pp.53-57.

[29] Miltenburg, J. and Zang, W. "A comparative evaluation of nine well-known algorithms for solving the cell formation problem in group technology," *J. Opns. Mgmt.*, 10(1972), pp.44-72.

[30] Moon, Y. B. and Chi, S. C. "Generalized part family formation using neural network techniques," *J. Manuf. Syst.*, 11(1992), pp. 149-159.

[31] Mosier, C. T. and Taube, L. "Weighted similarity measure heuristics for the group technology machine clustering problem," *Omega*, 13(1985), pp.577-579.

[32] Nair, G. J. K. and Narendran, T. T. "Grouping index : a new quantitative criterion for goodness of block-diagonal forms in group technology," *Int. J. Prod. Res.*, 34(1996), pp.2767-2782.

[33] Oral, M. and Kettani, O. "A linearization procedure for quadratic and cubic mixed-integer problems," *Opns. Res.*, 40(1992), pp. S109-S116.

[34] Seifoddini, H. "A note on the similarity coefficient method and the problem of improper machine assignment in group technology problem," *Int. J. Prod. Res.*, 27(1989), pp. 1161-1165.

[35] Seifoddini, H. "Single linkage versus average linkage clustering in machine cells formation applications," *Computers Ind. Engng.*, 16 (1989), pp.419-426.

[36] Selim, H. M., Askin, R. G. and Vakharia, A. J. "Cell formation in group technology: review, evaluation and directions for future research," *Computers Ind. Engng.*, 34(1998), pp.3-20.

[37] Srinivasan, G., Narendran, T. T. and Mahadevan, B. "An assignment model for the part-families problem in group technology," *Int. J. Prod. Res.*, 28(1990), pp.145-152.

[38] Stanfel, L. "Machine clustering for economic production," *Engng. Costs Prod. Econ.*, 9(1985), pp.73-81.

[39] Tabucanon, M. T. and Ojha, R. "ICRMA—a heuristic approach for intercellular flow reduction in cellular manufacturing systems," *Material flow*, 4(1987), pp.189-197.

[40] Vakharia, A. J. and Wemerlöve, U. "A comparative investigation of hierarchical clustering techniques and dissimilarity measures appled to the cell formation problem," *J. Opns. Mgmt.*, 13(1995), pp.117-138.

[41] Ventura, J. A., Chen, F. F. and Wu, C. H. "Grouping parts and tools in flexible manufacturing systems," *Int. J. Prod. Res.*, 28 (1990), pp.1039-1056.

[42] Viswanathan. S. "Configuring cellular manufacturing systems : a quadratic integer programming formulation and a simple interchange heuristic," *Int. J. Prod. Res.*, 33(1995), pp.361-376.

[43] Wang, J. and Roze, C. "Formation of machine cells and part families : a modified *p*-median model and a comparative study," *Int. J. Prod. Res.*, 35(1997), pp.1259-1286.

[44] Wei, J. C. and Kern, G. M. "A linear cell clustering algorithm for group technology," *Int. J. Prod. Res.*, 27(1989), pp.2053-2062.

[45] Zhu, Z. W., Heady, R. B. and Reiners, S. "An efficient zero-one formulation of the cell formation problem," *Computers. Ind. Engng.*, 28(1995), pp.911-916.