Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

An Exact Method for Solving the Manufacturing Cell Formation Problem

Bouazza Elbenani
Jacques A. Ferland

August 2010

CIRRELT-2010-37

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

An Exact Method for Solving the Manufacturing Cell Formation Problem

Bouazza Elbenani[1], Jacques Ferland[2,*]

[1] Department of Computer Science, University Mohammed V, Avenue des Nations Unies, Agdal, B.P. 554 Rabat-Chellah, Maroc

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), and Department of Computer Science and Operations Research, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7

Abstract. A linear binary programming formulation is introduced to generate a solution for the cell formation problem using CPLEX. Numerical experimentation is completed with 35 benchmark problems to evaluate the quality of the solution generated with heuristic methods proposed in the literature. This experimentation indicates that for the smaller problems, the best-known solutions are the same as those generated with CPLEX.

Keywords. Cell formation problem, linear binary programming formulation, heuristic methods, grouping efficiency.

Acknowledgements. This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), grant (OGP0008312).

---

* Corresponding author: JacquesA.Ferland@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2010

1. Introduction

In group technology or in cellular manufacturing, a system including machines and parts are interacting. To maximize the efficiency of the system, a cell formation problem is solved in order to partition the system into subsystems that are as autonomous as possible in the sense that the interactions of the machines and the parts within a subsystem are maximize and that the interactions between machines and parts of other subsystems are reduced as much as possible. This gives rise to solving a cell formation problem.

The cell formation problem is a NP hard optimization problem [13]. For this reason, several heuristic methods have been developed over the last forty years to generate good solutions in reasonable computational time. Nevertheless, only few integer programming formulations [1, 23, 27] have been proposed to solve the smaller problems in order to evaluate the quality of the solutions proposed by the heuristic methods. In this paper we try to give a partial answer to evaluate how close the solutions generated by heuristic methods are from optimality. To complete this analysis, we consider a set of 35 benchmark problems that the authors solve to compare their results.

In order to learn more about the different methods proposed to solve the cell formation problem, we refer the reader to the survey proposed in [14], where the authors review briefly the different methodologies used to solved the problem: cluster analysis, graph partitioning, mathematical programming, genetic (population based) algorithms, local search methods (tabu search, simulated annealing), and hybrids of these methods. To determine the best-known solutions for the 35 benchmark problems, we consider the results reported in [35] where the authors compare their results with other already published in the literature. Hence for each problem we consider the best-known solution among those generated by the following methods:
ZODIAC (clustering method) [10]
GRAPHICS (clustering method) [32]
MST (clustering method) [31]
GATSP (genetic algorithm solving a traveling salesman formulation) [11]
GP–GA (genetic programming) [12]
SA (simulated annealing) [37]
GA (genetic algorithm) [37]
TS (tabu search) [37]
EA–GA (genetic algorithm and local search) [14]
EnGGA (grouping genetic algorithm and greedy heuristic) [35].

In Section 2, we consider the grouping efficiency measure to formulate the cell formulation problem as a fractional binary integer programming problem. Then we approximate this problem into a binary integer programming problem in order to solve the problem with CPLEX. The numerical results summarized in Section 3 indicate that for the smaller problems, the best-known solutions are the same as those generated with CPLEX.

2. Problem formulation

To formulate the cell formation problem, consider the following two sets
$I =$ set of m machines: $i = 1, \ldots, m$
$J =$ set of n parts: $j = 1, \ldots, n.$

The production incidence matrix $A = \left[a_{ij} \right]$ indicates the interactions between the machines and the parts:

$$a_{ij} = \begin{cases} 1 \text{ if machine } i \text{ process part } j \\ 0 \text{ otherwise.} \end{cases}$$

Furthermore, a part j may be processed by several machines. A production cell k $(k = 1, \ldots, K)$ includes a subset (group) of machines $C_k \subset I$ and a subset (family) of parts $F_k \subset J$. The problem is to determine a solution including K production cells $(C, F) = \{(C_1, F_1), \ldots, (C_K, F_K)\}$ as *autonomous* as possible. Note that the K production cells induces partitions of the machines set and of the parts set:

$$C_1 \cup \ldots \cup C_K = I \quad \text{and} \quad F_1 \cup \ldots \cup F_K = J$$

and for all pairs of different cell indices k_1 and $k_2 \in \{1, \ldots, K\}$

$$C_{k_1} \cap C_{k_2} = \phi \quad \text{and} \quad F_{k_1} \cap F_{k_2} = \phi.$$

In the following example [3], the second matrix indicates a partition into 3 different cells illustrated in the gray zones. The solution includes the 3 machine groups {(6,7), (1,2), (3,4,5)} and the 3 part families {(4,5,8,10), (1,2,6,9), (3,7,11)}.

Machines \ Parts	1	2	3	4	5	6	7	8	9	10	11
1	1	1	0	0	0	1	0	0	0	0	0
2	0	1	0	0	0	1	0	0	1	0	0
3	1	0	1	0	0	0	1	0	0	0	1
4	0	0	1	0	0	0	1	0	0	0	0
5	0	0	1	1	0	0	0	0	0	0	1
6	0	0	0	1	1	0	0	0	0	1	0
7	0	0	0	0	1	0	0	1	0	1	0

(a) Incidence Matrix

Machines \ Parts	4	5	8	10	1	2	6	9	3	7	11
6	1	1	0	1	0	0	0	0	0	0	0
7	0	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	0	0	0	0
2	0	0	0	0	0	1	1	1	0	0	0
3	0	0	0	0	1	0	0	0	1	1	1
4	0	0	0	0	0	0	0	0	1	0	0
5	1	0	0	0	0	0	0	0	1	1	1

(b) Matrix Solution Figure 1.a

Figure 1. Boctor Matrix

The *exceptional elements* (5,4) and (3,1) correspond to entries having a value 1 that lay outside of the gray diagonal blocks.

To measure the *autonomy* of a solution, different measures have been proposed, and Sarker and Khan carry out a comparative study in [28]. In this paper we consider the grouping efficacy *Eff* [18] that is usually used by the authors to compare the efficiency of their methods;

$$Eff = \frac{a - a_1^{Out}}{a + a_0^{In}} = 1 - (\frac{a_0^{In} + a_1^{Out}}{a + a_0^{In}}) \tag{1}$$

where $a = \sum_{i=1}^{M} \sum_{j=1}^{P} a_{ij}$ denotes the total number of entries equal to 1 in the matrix *A*, a_1^{Out} denotes the number of *exceptional* elements, and a_0^{In} is the number of zero entries in the gray diagonal blocks. The objective function of maximizing *Eff* is then equivalent to minimize

$$ComEff = (\frac{a_0^{In} + a_1^{Out}}{a + a_0^{In}}) .$$

To formulate the mathematical formulation of the problem, we introduce the following binary variables:
for each pair $i = 1, \ldots, m;\ k = 1, \ldots, K$

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

for each pair $j = 1, \ldots, n;\ k = 1, \ldots, K$

$$y_{jk} = \begin{cases} 1 & \text{if part } j \text{ belongs to cell } k \\ 0 & \text{otherwise.} \end{cases}$$

To evaluate the objective function *ComEff*, it is easy to verify that

$$a_1^{out} = a - \sum_{k=1}^{K} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_{ik} y_{jk}$$

$$a_0^{In} = \sum_{k=1}^{K} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(1 - a_{ij}\right) x_{ik} y_{jk},$$

and it follows that the objective function *ComEff* is a fractional function in *x* and *y*. In order to formulate an integer programming formulation for the cell formation problem, we approximate *ComEff* with another function *PE* specified in terms of a_0^{In} (the number of zero entries in the gray diagonal blocks) and a_1^{Out} (the number of *exceptional* elements):

$$PE = a_0^{In} + \beta a_1^{Out}$$

where $\beta > 0$ is used to give different weights to the two terms. Note that reducing the two terms included in the function *PE* should improve the grouping efficiency and induce more autonomous solution for the cell formation problem.

Now the objective function *PE* can be formulated in terms of the variables *x* and *y*:

$$
\begin{aligned}
PE &= \sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(1-a_{ij}\right)x_{ik}y_{jk} + \beta\left(a-\sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}x_{ik}y_{jk}\right)\\
&= \beta a + \sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(1-\left(1+\beta\right)a_{ij}\right)x_{ik}y_{jk}\\
&= \left(\alpha-1\right)a + \sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(1-\alpha a_{ij}\right)x_{ik}y_{jk}
\end{aligned}
$$

where $\alpha = 1+\beta.$ Hence in this paper we are considering the following binary programming problem $M(x,y)$ of the cell partitioning problem:

$$M\left(x,y\right) \qquad \text{Min } PE = \left(\alpha-1\right)a + \sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(1-\alpha a_{ij}\right)x_{ik}y_{jk}$$

$$\text{Subject to } \sum_{k=1}^{K}x_{ik}=1 \qquad i=1,\dots,m \tag{2}$$

$$\sum_{k=1}^{K}y_{jk}=1 \qquad j=1,\dots,n \tag{3}$$

$$\sum_{i=1}^{m}x_{ik}\geq 1 \qquad k=1,\dots,K \tag{4}$$

$$\sum_{j=1}^{n}y_{jk}\geq 1 \qquad k=1,\dots,K \tag{5}$$

$$x_{ik}=0 \text{ or } 1 \qquad i=1,\dots,m; k=1,\dots,K \tag{6}$$

$$y_{jk}=0 \text{ or } 1 \qquad j=1,\dots,n; k=1,\dots,K \tag{7}$$

The constraints (2) and (3) ensure that each machine and each part is assigned to exactly one cell, respectively. The constraints (4) and (5) ensure that each cell includes at least one machine and one part. Finally, the variables are binary in (6) and (7). In our numerical experimentation we fix the number K of cells for each problem to its value in the best-known solution, and constraints (4) and (5) eliminate any empty cell.

Referring to [23], we now transform $M(x,y)$ into a linear binary programming problem using additional binary variables w_{ij}^{k} to replace the product $x_{ik}y_{jk}$:

$$w_{ij}^{k} = x_{ik}y_{jk} \quad i=1,\dots,m;\ j=1,\dots,n; k=1,\dots,K. \tag{8}$$

To complete a linear formulation of the cell formation problem, we use the following linear relations equivalent to the quadratic relations (8):

$$-w_{ij}^{k}+x_{ik}+y_{jk}-1.5\leq 0 \qquad i=1,\dots,m;\ j=1,\dots,n; k=1,\dots,K \tag{9}$$

$$1.5w_{ij}^{k}-x_{ik}-y_{jk}\leq 0 \qquad i=1,\dots,m;\ j=1,\dots,n; k=1,\dots,K. \tag{10}$$

The linear binary programming problem $M(x,y,w)$ is obtained by including the additional constraints (9) and (10) into problem $M(x,y)$ and replacing the objective function PE by

$$PE\left(w\right)=\left(\alpha-1\right)a+\sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(1-\alpha a_{ij}\right)w_{ij}^{k}.$$

This problem can be solved using integer linear programming software like CPLEX.

3. Numerical results

In this paper we consider the 35 benchmark problems given in [14] and [35] where the results obtained with several different methods are reported. Our purpose is to solve the corresponding $M(x,y,w)$ associated problems with CPLEX 10.11, and to evaluate the grouping efficiency *Eff* of these solutions. Now since CPLEX is an exact procedure, then comparing the *Eff* value of CPLEX with the grouping efficiency of a solution generated with a heuristic method should indicate how close this solution is to optimality.

In order to complete this analysis, we consider the numerical results for the methods mentioned in the introduction. Referring to the result exhibited in [14] and [35], we can identify the best-known solution for each problem. The problems are summarized in Table 1. The problem number and the source are included in the first two columns of the table. Then the size of the problems (values of m and n) and the number of cells K are in columns 3, 4, and 5. The best known value of the grouping efficiency for each of the 35 problems is included in column 6 of the table.

The $M(x,y,w)$ problems are solved with CPLEX 10.11, and the computational testing are performed on a AMD processor running at 2.2 GHz with 4096 Kilobytes of central memory and 1024 Kilobytes of memory cache. Each problem is solved twice for two different values of the parameter α $(1 \text{ and } 2)$. The grouping efficiency *Eff* is evaluated for both solutions, and the value reported in column 7 of the table correspond to the best solution. The solution time to obtain this solution is included in column 8.

The results in the table indicate that the first 11 problems are easy to solve since CPLEX requires less than 2 seconds to obtain the solution, with the exception of problem P9 requiring 5.03 seconds. The other problems are more difficult to solve since they require more than 78 seconds. For the larger problems (P18, P21, and P25 to P34), the time limit of 86400 seconds is reached before reaching the solutions. This is indicated by a * in column of the table. For the other larger problems P14, P19, and P35, we reach the optimal solution, but the computational time indicated in the table exceeds 86400 seconds.

Problem number	Problem source	*m*	*n*	*K*	Best-know solution	CPLEX *Eff*	Solution time (sec)
P1	[17]	5	7	2	82.35	82.35	0.33
P2	[36, fig.4a]	5	7	2	69.57	69.57	0.35
P3	[29]	5	18	2	79.59	79.59	0.47
P4	[21]	6	8	2	76.92	76.92	0.37
P5	[22]	7	11	5	60.87	60.87	1.67
P6	[3]	7	11	4	70.83	70.83	1.17
P7	[30]	8	12	4	69.44	69.44	1.69
P8	[7]	8	20	3	85.25	85.25	2.00
P9	[8]	8	20	2	58.72	58.72	5.03
P10	[25]	10	10	5	75	75	1.82
P11	[6]	10	15	3	92	92	1.72
P12	[2]	14	24	7	72.06	72.06	78.11
P13	[34]	14	24	7	71.83	71.83	103.68
P14	[24]	16	24	8	53.26	53.26	* 101783.88
P15	[33]	16	30	6	68.99	**69.53**	752.01
P16	[16]	16	43	8	57.23	57.23	86400
P17	[5]	18	24	9	57.73	57.73	14909.77
P18	[26]	20	20	5	43.45	*39.66*	*
P19	[19]	20	23	7	50.81	50.81	* 869103
P20	[5]	20	35	5	77.91	77.91	144.63
P21	[4]	20	35	5	57.98	*55.49*	*
P22	[7]	24	40	7	100	100	108.58
P23	[7]	24	40	7	85.11	85.11	496.95
P24	[7]	24	40	7	73.51	73.51	6556.99
P25	[7]	24	40	11	53.29	*47.95*	*
P26	[7]	24	40	12	48.95	*39.39*	*
P27	[7]	24	40	12	46.58	*41.84*	*
P28	[24]	27	27	5	54.82	*50.57*	*
P29	[5]	28	46	10	46.91	*35.68*	*
P30	[20]	30	41	14	63.12	*59.70*	*
P31	[34, fig. 5]	30	50	13	60.12	*49.69*	*
P32	[34, fig. 6]	30	50	14	50.83	*41.42*	*
P33	[17]	36	90	17	46.67	*43.77*	*
P34	[24]	37	53	3	60.64	*57.47*	*
P35	[10]	40	100	10	84.03	84.03	* 1572184.5

Table 1: Comparing the best-known solutions with *Eff* for CPLEX

For the problems solved to optimality with CPLEX, the value of *Eff* is equal to the best-known solutions, except for problem P15 where the value is larger by a factor of 0.8%. This observation indicates that for these problems the heuristic methods are very efficient since they can reach the same solutions as those obtained by solving the linear binary programming problem $M(x,y,w)$ where *ComEff* is approximated by the function *PE*. For the problem where CPLEX reaches the solution time limit (indicated by *), we indicate in italic the value of *Eff* of the best solution reached by CPLEX. Furthermore, for these 12 problems, the average value of the grouping efficiency of the heuristic methods is better by a factor of 11.17% better over the average corresponding values of *Eff* reached by CPLEX.

4. Conclusion

We introduce a binary linear programming problem for the cell formation problem where the value of the grouping efficiency is replaced by another function *PE* specified in terms of a_0^{In} (the number of zero entries in the gray diagonal blocks) and a_1^{Out} (the number of *exceptional* elements). CPLEX 10.11 is used to solve 35 benchmark problems. For the 22 smaller problems solved to optimality, the corresponding values of the grouping efficiency *Eff* are equal to the values of the best-known solutions obtained with the heuristic methods, and for an additional problem, the *Eff* is better than the best-known solution. Hence this indicates that the best-known solutions are fairly closed to the optimal solutions.

Acknowledgement

This research was supported by NSERC grant (OGP0008312) from CANADA.

References

[1] G.K. Adil, D. Rajamani, D.A. Strong, Mathematical model for cell formation considering investment and operational costs, European Journal of Operational Research 69 (3) (1993) 330–341.

[2] R.G. Askin, S.P. Subramanian, A cost-based heuristic for group technology configuration, International Journal of Production Research 25 (1) (1987) 101–113.

[3] F.F. Boctor, A linear formulation of the machine-part cell formation problem, International Journal of Production Research 29 (1991) 343–356.

[4] W.J. Boe, C.H. Cheng, A close neighbour algorithm for designing cellular manufacturing systems, International Journal of Production Research 29 (10) (1991) 2097–2116.

[5] A.S. Carrie, Numerical taxonomy applied to group technology and plant layout, International Journal of Production Research 11 (4) (1973) 399–416.

[6] H.M. Chan, D.A. Milner, Direct clustering algorithm for group formation in cellular manufacture, Journal of Manufacturing Systems 1 (1) (1982) 65-75.

[7] M.P. Chandrasekharan, R. Rajagopalan, GROUPABILITY: an analysis of the properties of binary data matrices for group technology, International Journal of Production Research 27(6) (1989) 1035–1052.

[8] M.P. Chandrasekharan, R. Rajagopalan, MODROC: an extension of rank order clustering for group technology, International Journal of Production Research 24 (5) (1986a) 1221–1233.

[9] M.P. Chandrasekharan, R. Rajagopalan, An ideal seed non-hierarchical clustering algorithm for cellular manufacturing, International Journal of Production Research 24 (2) (1986b) 451–464.

[10] M.P. Chandrasekharan, R. Rajagopalan, ZODIAC: an algorithm for concurrent formation of part-families and machine-cells, International Journal of Production Research 25 (6) (1987) 835–50.

[11] C.H. Cheng, Y.P. Gupta, W.H. Lee, K.F. Wong, TSP-based heuristic for forming machine groups and part families, International Journal of Production Research 36 (1998) 1325–1337.

[12] C. Dimopoulos, N. Mort, A hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems, International Journal of Production Research 39 (2001) 1–19.

[13] C. Dimopoulos, A.M.S. Zalzala, Recent developments in evolutionary computations for manufacturing optimization: problems, solutions, and comparisons, IEEE Transactions on Evolutionary Computations 4 (2000) 93–113.

[14] J. Goncalves, M.G.C. Resende, An evolutionary algorithm for manufacturing cell formation, Computers&Industrial Engineering 47 (2004) 247–273.

[15] T.L. James, E.C. Brown, K.B. Keeling, A hybrid Grouping Genetic Algorithm for the cell formation problem, Computers & Operations Research 34 (2007) 2059–2079.

[16] J.R. King, Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm, International Journal of Production Research 18 (1980) 213–232.

[17] J.R. King, V. Nakornchai, Machine-component group formation in group technology: review and extension, International Journal of Production Research 20 (2) (1982) 117–133.

[18] C. Kumar,M. Chandrasekharan, Grouping efficiency: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, International Journal of Production Research 28 (1990) 233–243.

[19] K.R. Kumar, A. Kusiak, A.Vannelli, Grouping of parts and components in flexible manufacturing systems, European Journal of Operations Research 24 (1986) 387–397.

[20] K.R. Kumar, A. Vannelli, Strategic subcontracting for efficient disaggregated manufacturing, International Journal of Production Research 25 (12) (1987) 1715–1728.

[21] A. Kusiak, M. Cho, Similarity coefficient algorithm for solving the group technology problem, International Journal of Production Research 30 (11) (1992) 2633–2646.

[22] A. Kusiak, W.S. Chow, Efficient solving of the group technology problem, Journal of Manufacturing Systems 6 (2) (1987) 117–124.

[23] I. Mahdavi, B.F. Javadi, K. Alipour, J. Slomp, Designing a new mathematical model for cellular manufacturing system based on cell utilization, Applied mathematics and Computation 190 (2007) 662 – 670.

[24] W.T. McCormick, P.J. Schweitzer, T.W. White, Problem decomposition and data reorganization by a clustering technique, Operations Research 20 (1972) 993–1009.

[25] C.T. Mosier, L. Taube, The facets of group technology and their impact on implementation, OMEGA 13 (5) (1985) 381–391.

[26] C. Mosier, L. Taube, Weighted similarity measure heuristics for the group technology machine clustering problem, OMEGA 13 (6) (1985) 577–83.

[27] D. Rajamani, N. Singh, Y.P. Aneja, Selection of parts and machines for cellularization: a mathematical programming approach, European Journal of Operational Research 62 (192) 47–54.

[28] B. Sarker, M. Khan, A comparison of existing grouping efficiency measures and a new grouping efficiency measure, IIE Transactions 33 (2001) 11–27.

[29] H. Seifoddini, A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications, International Journal of Production Research 27 (7) (1989) 1161–1165.

[30] H. Seifoddini, P.M. Wolfe, Application of the similarity coefficient method in group technology, IIE Transactions 18 (3) (1986) 271–277.

[31] G. Srinivasan, A clustering-algorithm for machine cell-formation in group technology using minimum spanning-trees, International Journal of Production Research 32 (1994) 2149–2158.

[32] G. Srinivasan, T.T. Narendran, GRAPHICS. A nonhierarchical clustering algorithm for group technology, International Journal of Production Research 29 (1991) 463–478.

[33] G. Srinivasan, T.T. Narendran, B. Mahadevan, An assignment model for the part-families problem in group technology, International Journal of Production Research 28 (1) (1990) 145–152.

[34] L.E. Stanfel, Machine clustering for economic production, Engineering Costs and Production Economics 9 (1985) 73–81.

[35] T. Tunnukij, C. Hicks, An Enhanced Genetic Algorithm for solving the cell formation problem, International Journal of Production research 47 (2009) 1989–2007.

[36] P.H. Waghodekar, S. Sahu, Machine-component cell formation in group technology MACE, International Journal of Production Research 22 (1984) 937–948.

[37] S. Zolfaghari, M. Liang, Comparative study of simulated annealing, genetic algorithms and tabu search for solving binary and comprehensive machine-grouping problems, International Journal of Production research 40 (2002) 2141–2158.