



CIRRELT

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

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Cell Formation Problem Solved Exactly with the Dinkelbach Algorithm

**Bouazza Elbenani
Jacques A. Ferland**

February 2012

CIRRELT-2012-07

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

Cell Formation Problem Solved Exactly with the Dinkelbach Algorithm

Bouazza Elbenani¹, Jacques A. Ferland^{2,*}

¹ Laboratoire de Datamining et Réseaux, Mohammed V-Agdal University, B.P. 1014, Rabat, Morocco

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. The cell formation problem has been extensively studied in the literature, but very few authors have proposed exact methods to solve it. In this paper the cell formation problem is transformed into an equivalent binary linear fractional programming problem, and it is solved exactly with the Dinkelbach algorithm. At each iteration of the method, the binary linear problem is solved with the IBM ILOG CPLEX 12.2 Optimizer. A numerical experimentation is completed using 35 benchmark problems, and 27 of these problems are solved exactly. Moreover, these results allow us to validate the quality of the solution generated with metaheuristic methods proposed in the literature since the best known solutions are equal to the optimal solutions for these 27 problems. To our knowledge, this is the first time that experimentation with an exact procedure is completed for the 35 benchmarked problems used in the literature.

Keywords. Cell formation problem, binary linear fractional programming, grouping efficiency, Dinkelbach algorithm, optimal solution

Acknowledgements. This work was partly supported by the Natural Sciences and Engineering Council of Canada (NSERC) under grant OGP 0008312. This support is gratefully acknowledged.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

* Corresponding author: JacquesA.Ferland@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2012

© Copyright Elbenani, Ferland and CIRRELT, 2012

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 maximized, 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 [1]. For this reason, several metaheuristic methods have been developed over the last forty years to generate good solutions in reasonable computational time. Nevertheless, only few integer programming formulations [2, 3, 4] have been proposed to solve the smaller problems in order to evaluate the quality of the solutions proposed by the metaheuristic methods. In this paper the cell formation problem is transformed into an equivalent binary linear fractional programming problem, and it is solved exactly with the Dinkelbach algorithm. At each iteration of the method, the binary linear problem is solved with the IBM ILOG CPLEX 12.2 Optimizer. Furthermore, we can evaluate how close the solutions generated by metaheuristic methods are from optimality.

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 [5], where the authors review briefly the different methodologies used to solve the problem: cluster analysis, graph partitioning, mathematical programming, genetic (population based) algorithms, local search methods (tabu search, simulated annealing), and hybrids of these methods. More recently two additional review papers of the cell formation problem have been published. In the first paper Papaioannou and Wilson [6], the authors survey the different techniques used to solve the problem, and they classify the different solution techniques as follows:

- Cluster analysis: techniques for recognizing structure in a data set
- Graph partitioning approaches: where a graph or a network representation is used to formulate the cell formation problem
- Mathematical programming methods: the cell formation problem is formulated like a non linear or linear integer programming problem

- Heuristic, Meta-heuristic and hybrid Meta-heuristic: The most popular methods are: simulated annealing, tabu search, genetic algorithms, colony optimization, particle swarm optimization, neural networks and fuzzy theory.

In the second paper Ghosh et al. [7], the authors introduce a survey of various genetic algorithms used to solve the cell formation problem. The success of genetic algorithms in solving this problem induced researchers to consider different variants and hybrids in order to generate very robust techniques.

To determine the best-known solutions for the 35 benchmark problems, we consider the results reported in Goncalves and Resende [5], James et al. [8], Tunnukij and Hicks [9], Mahdavi et al. [10], Nokhtehdan et al. [11], Xiangyong et al. [12] 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] Chandrasekharan and Rajagopalan [13]
- GRAPHICS [clustering method] Srinivasan and Narendran [14]
- MST [clustering method] Srinivasan [15]
- GATSP [genetic algorithm solving a traveling salesman formulation] Cheng et al. [16]
- GP-GA [genetic programming] Dimopoulos and Mort [17]
- SA [simulated annealing] Zolfaghari and Liang [18], Lin et al. [19], Ying et al. [20]
- GA [genetic algorithm] Zolfaghari and Liang [18]
- TS [tabu search] Zolfaghari and Liang [18]
- EA-GA [genetic algorithm and local search] Goncalves and Resende [5]
- HGGA [hybrid grouping genetic algorithm] James et al. [8]
- EnGGA [grouping genetic algorithm and greedy heuristic] Tunnukij and Hicks [9]
- HGDE [hybrid grouping version of differential evolution] Nokhtehdan et al. [11]
- ACO-CF [ant colony optimization metaheuristic] Xiangyong et al. [12], Farahani and Hosseini [21].

To complete this analysis, we consider a set of 35 benchmark problems that the authors solve to compare their results. In a previous study, the authors in [22] solve a linear approximation of the cell formation problem using the CPLEX Optimizer reaching an optimal solution of their approximation problem in 22 of the 35 benchmarked problems. In this paper, we introduce an exact method to solve the cell formation problem based on the Dinkelbach algorithm [23]. The numerical experimentation indicates the optimal solution is obtained for 27 problems.

In Section 2, we describe the cell formation problem using the grouping efficacy measure used in literature, and we introduce an equivalent formulation as a binary linear fractional programming problem. The Dinkelbach algorithm is described in section 3. The numerical results are summarized in Section 4.

2 Problem formulation

To formulate the cell formation problem, consider the following notation:

M : number of production machines.

P : number of production parts.

C : number of production cells.

I : set of M machines ($i = 1, \dots, M$),

J : set of P parts ($j = 1, \dots, P$),

K : set of C cells ($k = 1, \dots, C$),

The production incidence matrix $A = [a_{ij}]$ 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 ($k = 1, \dots, C$) includes a subset (group) of machines $I_k \subset I$ and a subset (family) of parts $J_k \subset J$. The problem is to determine a solution including C production cells $(I, J) = \{(I_1, J_1), \dots, (I_C, J_C)\}$ as autonomous as possible. Note that the C production cells induces partitions of the machines set and of the parts set:

$$I_1 \cup \dots \cup I_C = I \quad \text{and} \quad J_1 \cup \dots \cup J_C = J$$

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

$$I_{k_1} \cap I_{k_2} = \phi \quad \text{and} \quad J_{k_1} \cap J_{k_2} = \phi.$$

To illustrate the production cells concept, consider a machine-part incidence matrix in Table 1. Table 2 indicates a partition into 3 different cells illustrated in the gray zones. The solution includes the 3 machine groups $\{(1,4,6), (3,5), (2)\}$ and the 3 part families $\{(2,4,6,8), (1,7), (3,5)\}$.

Table 1. Incidence matrix

Parts		1	2	3	4	5	6	7	8
Machines	1	0	1	0	1	1	1	0	1
	2	1	0	1	0	1	0	0	0
	3	1	0	1	0	0	0	1	0
	4	0	1	0	1	0	1	0	1
	5	1	0	0	0	0	0	1	1
	6	1	1	0	0	0	1	1	1

Table 2. Matrix solution

Parts		2	4	6	8	1	7	3	5
Machines	1	1	1	1	1	0	0	0	1
	4	1	1	1	1	0	0	0	0
	6	1	0	1	1	1	1	0	0
	3	0	0	0	0	1	1	1	0
	5	0	0	0	1	1	1	0	0
	2	0	0	0	0	1	0	1	1

The *exceptional elements* (1,5), (6,1), (6,7), (3,3), (5,8) and (2,1) 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, Sarker and Khan carry out a comparative study in [24]. In this paper we consider the grouping efficacy (*Eff*) that is usually used by the authors to compare the efficiency of their methods (see [25]):

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

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

To specify a mathematical programming formulation for the cell formation problem, we introduce the following binary variables:

for each pair $i = 1, \dots, M; k = 1, \dots, C$

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

for each pair $j = 1, \dots, P; k = 1, \dots, C$

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

It is easy to verify that

$$a_1^{out} = a - \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P a_{ij} x_{ik} y_{jk}$$

$$a_0^{in} = \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - a_{ij}) x_{ik} y_{jk},$$

and it follows that the objective function *ComEff* is a fractional function in x and y .

Denote the numerator and the denominator of *ComEff* as

$$N(x, y) = a_0^{in} + a_1^{out} = \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - a_{ij}) x_{ik} y_{jk} + a - \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P a_{ij} x_{ik} y_{jk}$$

$$= a + \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - 2a_{ij}) x_{ik} y_{jk},$$

and

$$D(x, y) = a + a_0^{in} = a + \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - a_{ij}) x_{ik} y_{jk}, \text{ respectively.}$$

Hence the objective function can be written as:

$$ComEff(x, y) = \frac{N(x, y)}{D(x, y)} = \frac{a + \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - 2a_{ij}) x_{ik} y_{jk}}{a + \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - a_{ij}) x_{ik} y_{jk}} \quad (2)$$

Then the cell formation problem can be formulated as in the following $M(x, y)$ model:

$$\text{Min } ComEff = \frac{N(x, y)}{D(x, y)} = \frac{a + \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - 2a_{ij}) x_{ik} y_{jk}}{a + \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - a_{ij}) x_{ik} y_{jk}} \quad (2)$$

$$\text{Subject to } \sum_{k=1}^C x_{ik} = 1 \quad i = 1, \dots, M \quad (3)$$

$$\sum_{k=1}^C y_{jk} = 1 \quad j = 1, \dots, P \quad (4)$$

$$\sum_{i=1}^M x_{ik} \geq 1 \quad k = 1, \dots, C \quad (5)$$

$$\sum_{j=1}^P y_{jk} \geq 1 \quad k = 1, \dots, C \quad (6)$$

$$x_{ik} = 0 \text{ or } 1 \quad i = 1, \dots, M; k = 1, \dots, C \quad (7)$$

$$y_{jk} = 0 \text{ or } 1 \quad j = 1, \dots, P; k = 1, \dots, C \quad (8)$$

The constraints (3) and (4) ensure that each machine and each part is assigned to exactly one cell, respectively. The constraints (5) and (6) ensure that each cell includes at least one machine and one part (i.e., no empty cell). Finally, the variables are binary in (7) and (8). In our numerical experimentation we fix the number C of cells for each problem to its value in the best-known solution.

Now we use a similar approach as in Mahdavi et al. [4] to transform the problem $M(x, y)$ into a linear binary fractional problem using additional binary variables w_{ij}^k to replace the product $x_{ik} y_{jk}$:

$$w_{ij}^k = x_{ik} y_{jk} \quad i = 1, \dots, M; j = 1, \dots, P; k = 1, \dots, C \quad (9)$$

$$w_{ij}^k = 0 \text{ or } 1 \quad i = 1, \dots, M; j = 1, \dots, P; k = 1, \dots, C \quad (10)$$

To complete the linear formulation of the cell formation problem, we use the following linear relations equivalent to the quadratic relations (9):

$$w_{ij}^k - x_{ik} \leq 0 \quad i = 1, \dots, M; j = 1, \dots, P; k = 1, \dots, C \quad (11)$$

$$w_{ij}^k - y_{jk} \leq 0 \quad i = 1, \dots, M; j = 1, \dots, P; k = 1, \dots, C \quad (12)$$

$$-w_{ij}^k + x_{ik} + y_{jk} - 1 \leq 0 \quad i = 1, \dots, M; j = 1, \dots, P; k = 1, \dots, C \quad (13)$$

The binary fractional programming problem $M(x, y, w)$ is obtained by including the additional constraints (10), (11), (12), (13) into problem $M(x, y)$

$$\text{Min } ComEff(x, y, w) = \frac{N(x, y, w)}{D(x, y, w)} = \frac{a + \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - 2a_{ij}) w_{ij}^k}{a + \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P (1 - a_{ij}) w_{ij}^k} \quad (2)$$

$$\text{Subject to } \sum_{k=1}^C x_{ik} = 1 \quad i = 1, \dots, M \quad (3)$$

$$\sum_{k=1}^C y_{jk} = 1 \quad j = 1, \dots, P \quad (4)$$

$$\sum_{i=1}^M x_{ik} \geq 1 \quad k = 1, \dots, C \quad (5)$$

$$\sum_{j=1}^P y_{jk} \geq 1 \quad k = 1, \dots, C \quad (6)$$

$$w_{ij}^k - x_{ik} \leq 0 \quad i = 1, \dots, M; j = 1, \dots, P; k = 1, \dots, C \quad (11)$$

$$w_{ij}^k - y_{jk} \leq 0 \quad i = 1, \dots, M; j = 1, \dots, P; k = 1, \dots, C \quad (12)$$

$$-w_{ij}^k + x_{ik} + y_{jk} - 1 \leq 0 \quad i = 1, \dots, M; j = 1, \dots, P; k = 1, \dots, C \quad (13)$$

$$x_{ik} = 0 \text{ or } 1 \quad i = 1, \dots, M; k = 1, \dots, C \quad (7)$$

$$y_{jk} = 0 \text{ or } 1 \quad j = 1, \dots, P; k = 1, \dots, C \quad (8)$$

$$w_{ij}^k = 0 \text{ or } 1 \quad i = 1, \dots, M; j = 1, \dots, P; k = 1, \dots, C \quad (10)$$

We can now rely on the Dinkelbach [23] approach to solve the linear binary fractional problem $M(x, y, w)$.

3 Dinkelbach procedure

To simplify the notation, denote

$$(X, Y, W) = \{(x, y, w) : \text{solutions satisfying constraints (3)–(8), (10)–(13)}\}$$

The Dinkelbach procedure to solve $M(x, y, w)$ requires solving a sequence of problems where the objective function becomes binary linear by including the denominator $D(x, y, w)$ into the numerator $N(x, y, w)$:

$$M(\lambda, x, y, w) \quad \text{Min}_{(x, y, w) \in (X, Y, W)} N(x, y, w) - \lambda D(x, y, w)$$

for different values of λ . This linear binary problem is solved with CPLEX 12.2 Optimizer problem.

The Dinkelbach procedure can now be summarized as follows:

Dinkelbach procedure

- *Initialization.*
 - Start with an initial solution $(x^0, y^0, w^0) \in (X, Y, W)$
 - Take $\lambda_0 := \frac{N(x^0, y^0, w^0)}{D(x^0, y^0, w^0)}$, and $\zeta := 1$.
- *Step*
 - Solve the problem

$$M(\lambda_{\zeta-1}, x, y, w)$$

Let $(x^\zeta, y^\zeta, w^\zeta)$ be an optimal solution of this problem.
Let $F(\lambda_\zeta)$ be the optimal of this problem.
 - *Stopping rule.* If $F(\lambda_\zeta) < \delta$ (optimality tolerance), then STOP: $(x^\zeta, y^\zeta, w^\zeta)$ is an optimal solution of $M(x, y, w)$ and $Eff(x^\zeta, y^\zeta)$ is the maximal grouping efficacy.
 - Otherwise, let $\lambda_\zeta := \frac{N(x^\zeta, y^\zeta, w^\zeta)}{D(x^\zeta, y^\zeta, w^\zeta)}$ Let $\zeta := \zeta + 1$, and go back to *Step*

Since we expect the best-known efficacy value to be close to the optimal value, then we use this value to initialize λ^0 in order to reduce the number of iterations of the algorithm. Thus for each problem $\lambda^0 = \left(1 - \frac{\text{best-known solution}}{100}\right)$.

Finally, the algorithm converges since the sequence $\{\lambda_\zeta\}$ is strictly decreasing (Crouzeix et al.[26]).

4 Numerical results

In this paper we consider the 35 benchmark problems that are usually used by authors to compare the efficiency of their solution approaches. Referring to Goncalves and Resende [5], James et al. [8], Tunnuvij and Hicks [9], Mahdavi et al. [10], Nokhtehdan et al. [11], Xiangyong et al. [12], and Elbenani et al. [27] papers summarizing the results obtained by several authors, we can identify the best-known solution for each of the 35 problems. Our purpose is to solve the models $M(x, y, w)$ for the cell formation problem and the sub-problems $M(\lambda, x, y, w)$ with the Dinkelbach algorithm and the CPLEX 12.2 Optimizer, respectively. The optimal solutions obtained for the grouping efficacy Eff are compared with the grouping efficiency of the best-known solutions of 35 benchmark problems.

The problems are summarized in Table 3. The problem number and the reference where it is specified are included in the first two columns of the table 3. The size of the problems (values of M and P) and the number of cells C are in columns 3, 4, and 5. The best-known value of the grouping efficacy for each of the 35 problems is included in column 6 of the table 3.

The computational tests are performed on a AMD processor running at 2.2 GHz with 4096 Kilobytes of central memory and 1024 Kilobytes of memory cache. The relative optimality tolerance δ in the Dinkelbach algorithm is set at the value 10^{-6} . The grouping efficacy Eff obtained with the Dinkelbach algorithm and the solution time are reported in columns 7 and 8 of the Table 3, respectively.

The results in the Table 3 indicate that the first 11 problems, and the problems P20 and P22 are easy to solve since they requires less than 32 seconds of computational time. The solution approach can also reach the optimal solution for 14 of the larger problems (P12 to P17, P19, P21, P23, P24, P28, P30, P34, and P35), but the solution time can become very large. Unfortunately, it was not possible to solve the 8 other problems (P18, P25 to P27, P29, and P31 to P33) because the memory limit is reached before getting the optimal solutions. For these problems we place a * in the last column of the Table 3, and in column 7, we indicate in italic the value of Eff reached by our method before stopping.

Table 3: Comparing the best-known solutions with Eff for Dinkelbach algorithm

Problem number	Problem source	M	P	C	Best-known solution	Dinkelbach Eff	Solution time (sec)
P1	King and Nakornchai [28]	5	7	2	82.35	82.35	2.3
P2	Waghodekar and Sahu [29]	5	7	2	69.57	69.57	1.6
P3	Seifoddini [30]	5	18	2	79.59	79.59	3.1
P4	Kusiak and Cho [31]	6	8	2	76.92	76.92	2.0
P5	Kusiak and Chow [32]	7	11	5	60.87	60.87	30.6
P6	Boctor [33]	7	11	4	70.83	70.83	4.3
P7	Seifoddini and Wolfe [34]	8	12	4	69.44	69.44	9.6
P8	Chandrasekharan and Rajagopalon [35]	8	20	3	85.25	85.25	3.1
P9	Chandrasekharan and Rajagopalon [36]	8	20	2	58.72	58.72	3.5
P10	Mosier and Taube [37]	10	10	5	75	75	1.1
P11	Chan and Milner [38]	10	15	3	92	92	1.63
P12	Askin and Subramanian [39]	14	24	7	72.06	72.06	2188.7
P13	Stanfel [40]	14	24	7	71.83	71.83	593.2
P14	McCormick [41]	16	24	8	53.26	53.26	15130.5

P15	Srinivasan et al. [42]	16	30	6	69.53	69.53	252.5
P16	King [43]	16	43	8	57.53	57.53	183232.5
P17	Carrie [44]	18	24	9	57.73	57.73	2345.6
P18	Mosier and Taube [45]	20	20	5	43.45	43.06	*
P19	Kumar et al. [46]	20	23	7	50.81	50.81	131357.5
P20	Carrie [44]	20	35	5	77.91	77.91	31.1
P21	Boe and Cheng [47]	20	35	5	57.98	57.98	14583.6
P22	Chandrasekharan and Rajagopalon [48]	24	40	7	100	100	11.3
P23	Chandrasekharan and Rajagopalon [48]	24	40	7	85.11	85.11	230.7
P24	Chandrasekharan and Rajagopalon [48]	24	40	7	73.51	73.51	1101.1
P25	Chandrasekharan and Rajagopalon [48]	24	40	11	53.29	53.29	*
P26	Chandrasekharan and Rajagopalon [48]	24	40	12	48.95	48.28	*
P27	Chandrasekharan and Rajagopalon [48]	24	40	12	47.26	45.70	*
P28	McCormick [41]	27	27	5	54.82	54.82	958714.1
P29	Carrie [44]	28	46	10	47.06	46.75	*
P30	Kumar and Vannelli [49]	30	41	14	63.31	63.31	378300.0
P31	Stanfel [40]	30	50	13	60.12	60.12	*
P32	Stanfel [40]	30	50	14	50.83	50.55	*
P33	King and Nakornchai [28]	36	90	17	47.75	44.67	*
P34	McCormick [41]	37	53	3	60.64	60.64	268007.6
P35	Chandrasekharan and Rajagopalon [13]	40	100	10	84.03	84.03	7365.3

The approach in [22] allows reaching the optimal solution of 22 problems (P1 to P15, P17, P19, P20, P22 to P24, and P35) even if it is solving exactly a binary linear approximation of the cell formation problem with an earlier version of CPLEX . For the additional 5 problems solved exactly with the Dinkelbach methods (P16, P21, P28, P30 and P34), the optimal value is indicated in bold in Table 3. Finally, note that for the 8 problems (P18, P25 to P27, P29, and P31 to P33) where the method stops because of memory limit, the average value of their best known solutions grouping is better by a factor of 0.78 % better over the average corresponding values of *Eff* reached by our method.

It follows that the metaheuristic methods reaching the best known solutions seem to be very efficient since they can reach the optimal value *Eff* for the 27 problems solved to optimality by the Dinkelbach algorithm. But nevertheless, even if the Dinkelbach algorithm may require longer solution time than metaheuristic, it allows to get the optimal solutions for several problems, and to verify the efficiency of the metaheuristic methods.

5 Conclusion

The cell formation problem can be transformed into an equivalent binary linear fractional programming. We introduce a solution approach based on the Dinkelbach algorithm where the CPLEX 12.2 Optimizer is used to solve the binary linear programming problem at each iteration. The optimal value of the grouping efficacy Eff is obtained for 27 out of the 35 benchmarked problems often used in the literature. The best known solution of each of these 27 problems is equal to the optimal value of Eff . Hence our exact procedure may require longer solution time, but it allows to solve exactly several benchmarked problems, and to show the efficiency of the metaheuristic reaching the best known solutions.

Acknowledgement

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

References

- [1] Dimopoulos C, Zalzala AMS (2000). Recent developments in evolutionary computations for manufacturing optimization: problems, solutions, and comparisons. *IEEE Transactions on Evolutionary Computations* 4:93–113.
- [2] Rajamani D, Singh N, Aneja YP (1992). Selection of parts and machines for cellularization: a mathematical programming approach. *European Journal of Operational Research* 62:47–54.
- [3] Adil GK, Rajamani D, Strong DA (1993). Mathematical model for cell formation considering investment and operational costs. *European Journal of Operational Research* 69 (3):330–341.
- [4] Mahdavi I, Javadi BF, Alipour K, Slomp J (2007). Designing a new mathematical model for cellular manufacturing system based on cell utilization. *Applied mathematics and Computation* 190:662-670.
- [5] Goncalves J, Resende MGC (2004). An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering* 47:247–273.
- [6] Papaioannou G, Wilson JN (2010). the evolution of cell formation problem methodologies based on recent studies (1997-2008): Review and directions for future research. *European Journal of Operational Research* 206:509–521.
- [7] Ghosh T, Dan PK, Sengupta S, Chattopadhyay M (2010). Genetic rule based techniques in cellular manufacturing (1992-2010): a systematic survey. *International Journal of Engineering, Science and technology* 2 (5):198–215.
- [8] James TL, Brown EC, Keeling KB (2007). A hybrid Grouping Genetic Algorithm for the cell formation problem. *Computers & Operations Research* 34:2059–2079.

- [9] Tunnukij T, Hicks C (2009). An Enhanced Genetic Algorithm for solving the cell formation problem. *International Journal of Production research* 47:1989–2007.
- [10] Mahdavi I, Paydar MM, Solimanpur M, Heidarzade A (2009). Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. *Expert Systems With Applications* 36:6598-6604.
- [11] Noktehdan A, Karimi B, Kashan AH (2010). A differential evolution algorithm for the manufacturing cell formation problem using group based operators. *Expert Systems With Applications* 37:4822-4829.
- [12] Xiangyong Li, Baki MF, Aneja YP (2010). An ant colony optimization metaheuristic for machine-part cell formation problems. *Computers & Operations Research* 37:2071-2081.
- [13] Chandrasekharan MP, Rajagopalan R (1987). ZODIAC: an algorithm for concurrent formation of part-families and machine-cells. *International Journal of Production Research* 25(6):835–50.
- [14] Srinivasan G, Narendran TT (1991). GRAPHICS. A nonhierarchical clustering algorithm for group technology. *International Journal of Production Research* 29:463–478.
- [15] Srinivasan G (1994). A clustering-algorithm for machine cell-formation in group technology using minimum spanning-trees. *International Journal of Production Research* 32:2149–2158.
- [16] Cheng CH, Gupta YP, Lee WH, Wong KF (1998). TSP-based heuristic for forming machine groups and part families. *International Journal of Production Research* 36:1325–1337.
- [17] Dimopoulos C, Mort N (2001). A hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems. *International Journal of Production Research* 39:1–19.
- [18] Zolfaghari S, Liang M (2002). Comparative study of simulated annealing, genetic algorithms and tabu search for solving binary and comprehensive machine-grouping problems. *International Journal of Production research* 40:2141–2158.
- [19] Lin SW, Ying KC, Lee ZJ (2010). Part-machine cell formation in group technology using a simulated annealing-based meta-heuristic. *International Journal of Production Research* 48(12):3579-3591.
- [20] Ying KC, Lin SW, Lu CC (2011). Cell formation using a simulated annealing algorithm with variable neighbourhood. *European Journal of Industrial Engineering* 5:22-42.
- [21] Farahani MH, Hosseini L (2011). An ant colony optimization approach for the machine-part cell formation problem. *International Journal of Computational Intelligence Systems* 4:486-496.
- [22] Elbanani B, Ferland, JA (2012). An exact method for solving the manufacturing cell formation problem. *International Journal of Production Research* (Online).

- [23] Dinkelbach W (1967). On nonlinear fractional programming. *Management Science* 13(7):492–498.
- [24] Sarker B, Khan M (2001). A comparison of existing grouping efficiency measures and a new grouping efficiency measure. *IIE Transactions* 33:11–27.
- [25] Kumar C, Chandrasekharan M (1990). Grouping efficiency: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research* 28:233–243.
- [26] Crouzeix, JP, Ferland JA, Nguyen VH (2008). Revisiting Dinkelbach-type algorithms for generalized fractional programs. *Operational Research Society of India*, 45: 96-110.
- [27] Elbenani B, Ferland JA, Bellemare J (2012). Genetic algorithm and large neighborhood search to solve the cell formation problem. *Expert Systems with Applications* 39: 2408 – 2414.
- [28] King JR, Nakornchai V (1982). Machine-component group formation in group technology: review and extension. *International Journal of Production Research* 20 (2):117–133.
- [29] Waghodekar PH, Sahu S (1984). Machine-component cell formation in group technology MACE. *International Journal of Production Research* 22:937–948.
- [30] Seifoddini H (1989). 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):1161–1165.
- [31] Kusiak A, Cho M (1992). Similarity coefficient algorithm for solving the group technology problem. *International Journal of Production Research* 30 (11):2633–2646.
- [32] Kusiak A, Chow WS (1987). Efficient solving of the group technology problem. *Journal of Manufacturing Systems* 6 (2):117–124.
- [33] Boctor EF (1991). A linear formulation of the machine-part cell formation problem. *International Journal of Production Research* 29:343–356.
- [34] Seifoddini H, Wolfe PM (1986). Application of the similarity coefficient method in group technology. *IIE Transactions* 18 (3):271–277.
- [35] Chandrasekharan MP, Rajagopalan R (1986a). MODROC: an extension of rank order clustering for group technology. *International Journal of Production Research* 24 (5):1221–1233.
- [36] Chandrasekharan MP, Rajagopalan R (1986b). An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research* 24 (2):451–464.
- [37] Mosier CT, Taube L (1985a). The facets of group technology and their impact on implementation. *OMEGA* 13 (5):381–391.

- [38] Chan HM, Milner DA (1982). Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing Systems* 1 (1):65-75.
- [39] Askin RG, Subramanian SP (1987). A cost-based heuristic for group technology configuration. *International Journal of Production Research* 25 (1):101-113.
- [40] Stanfel LE (1985). Machine clustering for economic production. *Engineering Costs and Production Economics* 9:73-81.
- [41] McCormick WT, Schweitzer PJ, White TW (1972). Problem decomposition and data reorganization by a clustering technique. *Operations Research* 20:993-1009.
- [42] Srinivasan G, Narendran TT, Mahadevan B (1990). An assignment model for the part-families problem in group technology. *International Journal of Production Research* 28 (1):145-152.
- [43] King JR (1980). Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. *International Journal of Production Research* 18:213-232.
- [44] Carrie AS (1973). Numerical taxonomy applied to group technology and plant layout. *International Journal of Production Research* 11 (4):399-416.
- [45] Mosier CT, Taube L (1985b). Weighted similarity measure heuristics for the group technology machine clustering problem. *OMEGA* 13 (6):577-83.
- [46] Kumar KR, Kusiak A, Vannelli A (1986). Grouping of parts and components in flexible manufacturing systems. *European Journal of Operations Research* 24:387-397.
- [47] Boe WJ, Cheng CH (1991). A close neighbour algorithm for designing cellular manufacturing systems. *International Journal of Production Research* 29 (10):2097-2116.
- [48] Chandrasekharan MP, Rajagopalan R (1989). GROUPABILITY: an analysis of the properties of binary data matrices for group technology. *International Journal of Production Research* 27(6):1035-1052.
- [49] Kumar KR, Vannelli A (1987). Strategic subcontracting for efficient disaggregated manufacturing. *International Journal of Production Research* 25 (12):1715-1728.