The forty-second Annual Conference of the Italian Operational Research Society
(AIRO 2011 Conference)

# A Column Generation Heuristic for Machine-Part Cell Formation

Geraldo Ribeiro Filho[1]
Luiz Antonio N. Lorena[2]

[1] UNISUZ – Faculdade Unida de Suzano
Rua José Correia Gonçalves 57
08675-130 Suzano – SP
Brazil
geraldorf@uol.com.br

[2] LAC/INPE- Instituto Nacional de Pesquisas Espaciais
Caixa Postal 515
12227-010 São José dos Campos – SP
Brazil
lorena@lac.inpe.br

## Abstract

The Machine-Part Cell Formation is the problem of creating manufacture cells aiming best production flow of manageable sub-systems. Systems automation and control can be improved by the aggregation of similar parts into families, and machines into independent cells that completely manufactures families of parts. The objective of the problem is to form a given number of disjoint parts-machines groups in which products do not have to move from one cell to the other to be processed. This problem be viewed as a clustering problem, and can be modeled as a *p*-median location problem. This paper presents a column generation approach to *p*-median problem, adapted to produce feasible assignments of parts into families. A further heuristic step assigns machines to families of parts to form the manufacturing cells. Experimental tests were made using instances from the literature. The computational results obtained with the heuristic were as good as in the literature for the majority of the instances, and even better in some cases.

**Keywords**: manufacturing cells; column generation; heuristic

## 1 Introduction

The international competition and its consequent needs for quick answers to the market demands have lead several companies to consider non-traditional approaches to control and design the manufacturing systems. The “group technology” as described by Burbidge (1963) decompose manufacturing systems into manageable sub-systems, or groups, by aggregating similar parts into families and machines into cells. The production flow analysis of Burbidge (1969) is one of the first and well-known methodologies associated with group technology. Cellular manufacturing can simplify automation and control through the creation of independent cells that completely manufactures families of parts. The objective of this problem is to form a given number of disjoint part-machines groups in which products do not have to move from one cell to the other for processing. The number of groups to be formed is the equivalent to the number of families of parts and the number of machine cells to be formed. We call this problem as the Machine-Part Cell Formation (MPCF) problem.

Several optimization approaches have being proposed in the literature to create manufacturing cells. Heuristics for MPCF generally work over machine-part binary matrices, with one of the dimensions corresponding to machines and the other corresponding to parts. The matrices elements being ones or zeros indicate, respectively, which machines are used, and not used, to produce each part. These algorithms basically change rows and columns positions to produce blocks of ones, forming families of parts and cells of machines, simultaneously. The work of McCormick Jr. et al. (1972, King (1980), Chandrasekharan and Rajagopalan (1986a), and Venugopal and Narendran (1993), analyze the binary matrices to extract properties and suggest cell formation algorithms.

Fig. 1 shows the original form of a matrix with 10 machines (rows) and 15 parts (columns).  Fig. 2 uses dashed lines and omit the zeros to better illustrate the same matrix after an attempt to form 3 part-machine groups by changing rows and columns positions. The solution shown has not produced completely independent cells as parts 0, 4 and 12 must be processed by machines in more than one cell.

```
     0  1  2  3  4  5  6  7  8  9  10 11 12 13 14
  --|---------------------------------------------
0 |0  0  0  0  0  0  0  1  0  0  1  0  1  1  0
1 |1  0  1  0  1  0  0  0  0  1  0  0  1  0  0
2 |1  1  0  0  0  0  0  0  1  0  0  0  0  0  0
3 |0  0  1  0  1  1  0  0  0  1  0  0  0  0  0
4 |0  1  0  1  1  0  1  0  1  0  0  1  0  0  0
5 |0  1  0  1  0  0  0  0  0  0  0  1  0  0  0
6 |0  1  0  1  0  0  1  0  1  0  0  1  0  0  0
7 |0  0  0  0  0  0  0  0  0  0  1  0  0  1  1
8 |0  0  0  0  0  0  0  1  0  0  0  0  1  1  1
9 |0  0  1  0  1  0  0  0  0  1  0  0  0  0  0
```

**Fig. 1** Original part-machine matrix

```
   1  3  6  8 11 | 0  2  4  5  9 | 7 10 12 13 14
-------------------------------------------------
1                | 1  1  1     1 |       1
3                |    1  1  1  1 |
9                |    1  1     1 |
-------------------------------------------------
2  1        1    | 1             |
4  1  1  1  1  1 |       1       |
5  1  1        1 |               |
6  1  1  1  1  1 |               |
-------------------------------------------------
0                |               | 1  1  1  1
7                |               |    1     1  1
8                |               | 1     1  1  1
```

**Fig. 2** Same matrix after grouping parts and machines

Many optimization techniques for MPCF have been proposed in the literature. Gupta and Seifoddini (1990) present a hierarchical clustering method, non-hierarchical clustering methods are presented in Chandrasekharan and Rajagopalan (1989), and Jayakrishnan Nair and Narendran (1998), techniques based on graphs in Rajagopalan and Batra (1975), and Lin et al. (1996), neural networks applications are shown in Malave and Ramchandran (1991), and Guerrero et al. (2002), and fuzzy logic based techniques are shown in Torkul et al. (2006), and Xu and Wang (1989). Meta-heuristics are also applied, like Simulated Annealing in Venugopal and Narendran (1992a), Tabu search in Lei and Wu (2006), and Genetic Algorithms in Venugopal and Narendran (1992b), Gonçalves and Resende (2004), and Ribeiro Filho and Lorena (2000). Jaumard et al. (1999) considered the objective of minimizing the number of bottleneck elements in the parts-machines matrix, modeling the problem as a large double assignment linear programming problem that was solved by column generation approach. The papers of Joines et al.(1995), Gonçalves and Resende (2004) and Papaioannou and Wilson (2008) present comprehensive reviews of alternative methodologies.

The MPCF is a clustering problem and was modeled as a *p*-median problem in some papers, beginning with Kusiak (1987) and explored by Won and Currie (2004), among others. The search for *p* median vertices on a network (graph) is a classical location problem. The objective is to locate a given *p* number of facilities (medians) to minimize the sum of the distances from each demand vertex to its nearest facility. Recently, Senne et al. (2007) have presented a column generation approach to *p*-median problems were the clusters corresponds to columns, and new columns are generated using the linear programming relaxation of a set covering problem with a cardinality constraint.

This paper presents a new model for the MPCF as a set partitioning problem with a cardinality constraint. The column generation approach to *p*-median problems introduced by Senne et al. (2007) is adapted to produce feasible assignments of parts into families. The assignments are based in Hamming or Jaccard distances for binary strings. A further heuristic step assigns machines to families of parts to form manufacturing cells.

The remaining of this paper is organized as follows: section 2 introduces the *p*-median formulation and the column generation algorithm, section 3 examines complimentary algorithms to assign parts to cells of machines or machines to families of parts, section 4 present computational results and final considerations are made in section 5.

## 2 P-median and column generation

Column generation is a powerful tool used to solve large scale linear programming problems. This technique can be used when the columns in the problem are not known previously and a complete enumeration of all columns is not a viable option, or when the problem is transformed using Dantzig-Wolfe (1960) decomposition. Column generation is a common choice in several well-known applications, such as the cutting-stock problem, vehicle routing and crew scheduling.

In the classical form, the column generation algorithm iterates between a column generator sub-problem and a restricted master problem. Solving the master problem produces dual costs, which are used in the sub-problem to determine whether its solution variables values might be added to the master problem as a new column to improve its solution.

The *p*-median problem considered in this paper can be formulated as the following set partition problem:

(SP-Pmed): $$v(SP-Pmed) = Min \sum_{j=1}^{m} c_j x_j \qquad (1)$$

subject to $$\sum_{j=1}^{m} a_{ij} x_j = 1 \; ; i = 1,...,n \qquad (2)$$

$$\sum_{j=1}^{m} x_j = p \qquad (3)$$

$$x_j \in \{0,1\}. \qquad (4)$$

Where:

$n = |N|$ is the number of parts in the set parts $N$,

$S = \{S_1, S_2, ..., S_m\}$, is a set of subsets of N,

$[a_{ij}]_{nxm}$ , is a matrix with $a_{ij} = 1$ if $i \in S_j$ , and $a_{ij} = 0$ otherwise;

$$c_j = \underset{i \in S_j}{Min} \left( \sum_{k \in S_j} d_{ik} \right); \qquad (5)$$

$[d_{ij}]_{nxn}$ is a symmetric distance (Hamming or Jaccard) matrix;

$m$ is the number of columns; and

$p$ is the number of families of parts to be created.

The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different.

The Jaccard index, also known as the Jaccard similarity coefficient, measures similarity between sample sets, like binary strings, and is defined as the size of the intersection divided by the size of the union of the sample sets. The Jaccard distance, which measures dissimilarity between sample sets, is complementary to the Jaccard coefficient and is obtained by subtracting the Jaccard coefficient from 1.

Given two binary strings, both with the same number of binary values, for $i$ and $j$ in {0,1}, $M_{ij}$ is the total number of positions $k$ in the strings A and B such that $a_k=i$ and $b_k=j$.

The Jaccard distance between A and B is given by:

$$d_{A,B} = \frac{M_{01} + M_{10}}{M_{01} + M_{10} + M_{11}} \tag{6}$$

The (SP-Med) formulation above was used for location problems in Senne et al. (2007). If $S$ is the set of all subsets of $N$, the formulation can give an optimal solution to the $p$-median problem. But the number of subsets can be huge, and therefore a partial set of columns is considered instead. Another approximation considers a problem to be solved by column generation as the following set covering formulation of SP-Pmed:

(SC-Pmed): $$v(SC - Pmed) = Min \sum_{j=1}^{m} c_j x_j \tag{7}$$

subject to $$\sum_{j=1}^{m} a_{ij} x_j \geq 1 \; ; i = 1,...n \tag{8}$$

$$\sum_{j=1}^{m} x_j = p \tag{9}$$

$$x_j \in [0,1]. \tag{10}$$

The SC-Pmed problem is also known as the restricted master problem in the column generation context. After defining an initial pool of columns, this problem is solved and the final dual costs $\pi_i$, $i$ = 1,...,$n$ and $\alpha$ associated to constraints (8) and (9), respectively, are used to generate new columns solving the following sub-problem:

(Sub-Pmed): $$v(Sub - Pmed) = \underset{i \in N}{Min} \left[ \underset{y_j \in \{0,1\}}{Min} \sum_{j \in N} (d_{ij} - \pi_j) y_j \right]. \tag{11}$$

The Sub-Pmed problem is easily solved setting $y_j = 1, \; if \; d_{ij} - \pi_j \leq 0$ and $y_j = 0, \; if \; d_{ij} - \pi_j > 0$, for each $i$ = 1,...,$n$. The column $\begin{bmatrix} y_j \\ 1 \end{bmatrix}$ is added to SC-Pmed when v(Sub-Pmed) < $\alpha$ . In fact, for $i$ = 1,...,$n$, all the corresponding columns satisfying $\left[ \underset{y_j \in \{0,1\}}{Min} \sum_{j \in N} (d_{ij} - \pi_j) y_j \right]$ < $\alpha$ can be added to the pool of columns, accelerating the column generation process.

The column generation algorithm is summarized as:

(i) Set an initial pool of columns to SC-Pmed;

(ii) Solve SC-Pmed and return the dual prices $\pi_j$ , $j$ = 1,...,$n$ and $\alpha$ ;

(iii) Solve the corresponding sub-problem (Sub-Pmed) appending to SC-Pmed the columns $\begin{bmatrix} y_j \\ 1 \end{bmatrix}$ satisfying $\left[ \underset{y_j \in \{0,1\}}{Min} \sum_{j \in N} (d_{ij} - \pi_j) y_j \right] < \alpha$, $i$=1,…,$n$ ;

(iv) If no columns are found in step (iii) then stop, otherwise return to (ii);

(v) Solve the SP-Pmed problem with the generated columns.

Solving the SP-Pmed problem in step (v) gives us a feasible assignment of parts to families. A further final step consists of the assignment of machines to cells that produce the part families. The machines assignment is described in the next section of this document.

The relaxed intermediary problems (SC-Pmed) in step (ii) and the set partition problem in step (v) are usually solved by software tools like CPLEX (2006). The stop condition in step (iv) is generally combined with another criterion, like a maximum limit for number of columns to be generated in the process.

The initial pool of columns in step (i) can be generated in several ways, simple or with higher processing cost. For instance, columns could be generated randomly choosing a number of parts for each column and randomly choosing each one these parts, or the generation process could implement some heuristic approach to create better initial columns, and perhaps accelerate the column generation phase.

In this work, the number of 1's (parts) in each initial column was given by a random number with a normal distribution with then mean given by the ratio between the number of parts in the problem ($n$) and the number of cells or groups to be formed ($p$), $E = \frac{n}{p}$, and standard deviation given by $s = \frac{E}{4}$. The parts with value set to 1 in the columns were then randomly chosen.

## 3 The assignments

Solving a MPCF problem by column generation involves a further step to form machines cells. The objective of each machine cell is to produce a family of parts. Therefore, the assignment consists of the association of each machine to one of the families of parts obtained in the previous column generation process.

The assignments are performed by a local search heuristic presented by Gonçalves and Resende (2004). Their work, on the opposite way, assigns parts to machine cells generated by a genetic algorithm.

The heuristic consists of an improvement procedure that is repeatedly applied. Each iteration $k$ of the procedure starts with a given initial set of part families $P_k^{INITIAL}$, and produces a set of machine cells $M_k^{FINAL}$, and another set of part families $P_k^{FINAL}$. Two block-diagonal matrices can be obtained by combining $P_k^{INITIAL}$ with $M_k^{FINAL}$ and $P_k^{FINAL}$ with $M_k^{FINAL}$. From these two matrices, the one with the highest grouping efficacy (as described at the end of this section) is chosen as the resulting block-diagonal matrix of the iteration $k$. The procedure stops if $P_k^{FINAL} = P_k^{INITIAL}$ or if the grouping efficacy of the block-diagonal matrix resulting from iteration $k$ is not greater than the grouping efficacy of the block-diagonal matrix resulted from the previous iteration $k$-1, for $k$>2. Otherwise, the procedure sets $P_{k+1}^{INITIAL} = P_k^{FINAL}$ and continues to iteration $k$+1.

Each iteration k of the local search heuristic consists of the following two steps:

(1) Assignment of machines to the initial set of part families $P_k^{INITIAL}$. The machines are assigned to a family of parts one at a time, in any order. A machine is assigned to the family of parts that maximizes an approximation of the grouping efficacy. A machine is assigned to the part family $P^*$, given by:

$$P^* = \arg\max_{P} \left\{ \frac{N_1 - N_{1,P}^{Out}}{N_1 + N_{0,P}^{In}} \right\} \tag{12}$$

Where

argmax is the argument that maximizes expression;

$N_1$ is the total number of 1's in matrix;

$N_{1,P}^{Out}$ is the total number of 1's outside the diagonal blocks if the machine is assigned to part family *P*;

$N_{0,P}^{In}$ is the total number of 0's inside the diagonal blocks if the machine is assigned to part family *P*.

The ideal blocks in the matrix should be as dense as possible, with few 0's in it and, also ideally, the solution should not have 1's outside the blocks. Therefore, the parameters $N_1$, $N_{1,P}^{Out}$ and $N_{0,P}^{In}$ in equation (12) are directly related to the quality of the solution and are used to obtain the group efficacy measure used in the literature.

In this step, the heuristic generates a set of machine cells $M_k^{FINAL}$. Let $\mu_k^1$ be the efficacy of the block-diagonal matrix defined by $P_k^{INITIAL}$ and $M_k^{FINAL}$.

(2) Assignment of parts to the set of machine cells $M_k^{FINAL}$, obtained in step (1). Parts are assigned to machine cells, one at a time, in any order. A part is assigned to the machine cell that maximizes an approximation of the grouping efficacy. A part is assigned to the machine cell *M**, given by:

$$M^* = \arg\max_{M} \left\{ \frac{N_1 - N_{1,M}^{Out}}{N_1 + N_{0,M}^{In}} \right\} \tag{13}$$

Where

argmax is the argument that maximizes expression;

$N_1$ is the total number of 1's in matrix;

$N_{1,M}^{Out}$ is the total number of 1's outside the diagonal blocks if the part is assigned to machine cell *M*;

$N_{0,M}^{In}$ is the total number of 0's inside the diagonal blocks if the part is assigned to machine cell *M*.

In this step, the local search heuristic generates a new set of families of parts $P_k^{FINAL}$. Let $\mu_k^2$ be the efficacy of the block-diagonal matrix defined by $P_k^{FINAL}$ and $M_k^{FINAL}$.

The block-diagonal matrix resulting from this iteration has a grouping efficacy given by $\mu_k = \max(\mu_k^1, \mu_k^2)$. If $P_k^{FINAL}$ = $P_k^{INITIAL}$ or $\mu_k \leq \mu_{k-1}$, the iterative process stops and the block-diagonal matrix of iteration $k$-1 is take as the result. Otherwise, the procedure sets $P_{k+1}^{INITIAL}$ = $P_k^{FINAL}$ and continues to step (1) of iteration $k$+1.

Therefore, give an initial set of families of parts $P^{INITIAL}$, obtained by solving the SP-Pmed problem, the local search algorithm to associate machines to families of parts can be represented as follows:

While the stop condition is false
    Obtain $M^{FINAL}$ from $P^{INITIAL}$, assigning machines to families of parts according the criterion given by expression 12;
    Obtain $P^{FINAL}$ from $M^{FINAL}$, assigning parts to cells of machines according to the criterion given by expression 13;
    $B_{IF}$ = block-diagonal matrix obtained with $P^{INITIAL}$ and $M^{FINAL}$;
    $B_{FF}$ = block-diagonal matrix obtained with $P^{FINAL}$ and $M^{FINAL}$;
    $e_{IF}$ = efficacy of the solution represented by $B_{IF}$;
    $e_{FF}$ = efficacy of the solution represented by $B_{FF}$;
    If $e_{IF} > e_{FF}$ then
        $B_{sol} = B_{IF}$ is the final solution matrix;
        $e_{sol} = e_{IF}$;
    Else
        $B_{sol} = B_{FF}$ is the final solution matrix;
        $e_{sol} = e_{FF}$;
    End If;
    Stop condition = ($P^{FINAL} == P^{INITIAL}$), or from the second iteration, $e_{sol}$ is not greater than $e_{sol}$ of previous iteration;
End While;
If the stop condition is false, $P^{INITIAL}$ = $P^{FINAL}$;
Return the block-diagonal matrix $B_{sol}$ with efficacy $e_{sol}$.

The above assignment procedure may lead to what the work of Gonçalves and Resende (2004) call singletons, cells having less than two parts or two machines. They discard it these cells by assign zero efficacy. To avoid this effect of the procedure we have implemented an additional step, after the assignment of machines to cells and parts to families, and before the efficacy calculation.

This additional procedure makes a list of eventually singleton machine cells and, while this list is not empty, repeatedly selects one cell from the list at random and looks for every other cell with at least three machines, and transfer to the singleton cell the machine that produces the larger increase in the solution efficacy. If the singleton cell becomes valid with the transfer, it is removed from the invalid cells list. An analog procedure takes place for families of parts.

The grouping efficacy measure used in the literature and in this work is given by a coefficient that takes into account the number of zeros inside the clusters and the number of ones outside the clusters, respectively, representing the cluster compactness and intercellular movement:

$$Coef = \frac{e - e_1}{e + e_0} \qquad (14)$$

Where:

$e$ is the number of 1's in the matrix
$e_0$ is the number of 0's inside the clusters
$e_1$ is the number of 1's outside the clusters

The ideal coefficient value is 1, that means no zeros inside and no ones outside the clusters, and a better clustering has a higher coefficient value.

## 4 Experimental Results

The experimental results were obtained with some instances used in the paper of Gonçalves and Resende (2004), and one additional instance with a 20x35 matrix of machines and parts from Burbidge (1963). Table 1 shows a list of the test instances, its origin in the literature, and its number of machines, parts and cells to be formed.

**Table 1** Test instances

| | Instance Origin | Machines | Parts | Cells |
|---|---|---|---|---|
| 1 | King and Nakornchai, 1982 | 5 | 7 | 2 |
| 2 | Waghodekar and Sahu, 1984 | 5 | 7 | 2 |
| 3 | Seifoddini, 1989 | 5 | 18 | 2 |
| 4 | Kusiak and Cho, 1992 | 6 | 8 | 2 |
| 5 | Kusiak and Chow, 1987 | 7 | 11 | 3 |
| 6 | Boctor, 1991 | 7 | 11 | 3 |
| 7 | Seifoddini and Wolfe, 1986 | 8 | 12 | 3 |
| 8 | Chandrasekharan and Rajagopalan, 1986a | 8 | 20 | 3 |
| 9 | Chandrasekharan and Rajagopalan, 1986b | 8 | 20 | 2 |
| 10 | Mosier and Taube, 1985a | 10 | 10 | 3 |
| 11 | Chan and Milner, 1982 | 10 | 15 | 3 |
| 12 | Askin and Subramanian, 1987 | 14 | 24 | 5 |
| 13 | Stanfel, 1985 | 14 | 24 | 5 |
| 14 | McCormick et al., 1972 | 16 | 24 | 5 |
| 15 | Srinivasan et al., 1990 | 16 | 30 | 4 |
| 16 | King, 1980 | 16 | 43 | 5 |
| 17 | Carrie, 1973 | 18 | 24 | 6 |
| 18 | Mosier and Taube, 1985b | 20 | 20 | 5 |
| 19 | Kumar et al., 1986 | 20 | 23 | 5 |
| 20 | Carrie, 1973 | 20 | 35 | 4 |
| 21 | Boe and Cheng, 1991 | 20 | 35 | 5 |
| 22 | Chandrasekharan and Rajagopalan, 1989 | 24 | 40 | 7 |
| 23 | Chandrasekharan and Rajagopalan, 1989 | 24 | 40 | 7 |
| 24 | Chandrasekharan and Rajagopalan, 1989 | 24 | 40 | 7 |
| 25 | Chandrasekharan and Rajagopalan, 1989 | 24 | 40 | 9 |
| 26 | Chandrasekharan and Rajagopalan, 1989 | 24 | 40 | 9 |
| 27 | Chandrasekharan and Rajagopalan, 1989 | 24 | 40 | 9 |
| 28 | McCormick et al., 1972 | 27 | 27 | 4 |
| 29 | Carrie, 1973 | 28 | 46 | 9 |
| 30 | Kumar and Vannelli, 1987 | 30 | 41 | 11 |
| 31 | Stanfel, 1985 | 30 | 50 | 12 |
| 32 | Stanfel, 1985 | 30 | 50 | 11 |
| 33 | McCormick et al., 1972 | 37 | 53 | 2 |
| 34 | Burbidge, 1969 | 30 | 35 | 4 |

Tests were made with both distances, Hamming and Jaccard, in the Column Generation (CG) algorithm. The performance measure was made with clustering efficacy coefficient. Table 2 shows the computational results obtained using Hamming distance, and Table 3 shows the results using the Jaccard distance.

The columns in the tables 2 and 3 shows the instance number as listed in Table 1, the clustering efficacy found in the literature (Gonçalves and Resende, 2004), maximum, minimum and average efficacy found with 10 runs of the algorithm, average total time, average time for the column generation phase and average time for the integer programming problem solution phase, the average of the number of iterations in the column generation loop, and the average number of generated columns in the ten runs of the algorithm. The tables have a last row with average values for each table column.

Bold font style in the maximum efficacy column and literature efficacy column indicates algorithm results equal or better than the literature. Particularly, tables 2 and 3 shows that for instances 21 and 33 the algorithm obtained better results than found in literature, using both distance measures.

The experiments, for all instances, used 300 randomly made initial columns for the columns generation phase, and a limit of 15 thousand generated columns was used as an additional stop condition for the process.

In table 2, using Hamming distance, we see that the algorithm was able to obtain equal or better results than those found in the literature for 20 out of the 34 instances (58.8%), and in table 3, using Jaccard distance, the same occurs for 21 out of 34 instances (61.8%). Tests were made using a 3 GHz Pentium IV microcomputer, with code written in C#, and using the Concert 2.6 library of the CPLEX 11 solver.

## 5 Final considerations

This paper examined a new heuristic for MPCF problems. The MPCF is modeled as a clustering problem assigning parts to families in a set partitioning problem with a cardinality constraint. The cells of machines and families of parts are created based on assignments obtained during column generation and a local search process.

The computational results obtained with the heuristic were as good as in the literature for the majority of the instances, and even better in some cases. Complimentary research is in course to deal with large scale instances, such as the effective management of columns, removing unproductive columns along the process.

**Acknowledgments**

The second author acknowledges Conselho Nacional de Desenvolvimento Científico e Tecnológico -CNPq (proc. 476862/2012-4 and 300692/2009-9)

**Table 2** Results obtained using Hamming distance

| Inst | Efficacy | | | | Time (sec) | | | Loop | Cols |
|---|---|---|---|---|---|---|---|---|---|
| | Liter | Max | Min | Avg | Tot | CG | IP | | |
| 1 | **0.7368** | **0.7368** | 0.7368 | 0.7368 | 0.1 | 0.0 | 0.0 | 0.3 | 0.6 |
| 2 | **0.6250** | **0.6250** | 0.6087 | 0.6103 | 0.1 | 0.0 | 0.0 | 1.0 | 1.6 |
| 3 | **0.7959** | **0.7959** | 0.7959 | 0.7959 | 1.1 | 0.2 | 0.8 | 75.6 | 907.6 |
| 4 | **0.7692** | **0.7692** | 0.7692 | 0.7692 | 0.1 | 0.0 | 0.0 | 1.2 | 4.2 |
| 5 | **0.5313** | **0.5313** | 0.4688 | 0.5157 | 0.3 | 0.1 | 0.2 | 61.7 | 459.0 |
| 6 | **0.7037** | **0.7037** | 0.7037 | 0.7037 | 0.2 | 0.1 | 0.1 | 32.6 | 283.0 |
| 7 | **0.6830** | **0.6830** | 0.6830 | 0.6830 | 0.2 | 0.1 | 0.1 | 34.1 | 297.9 |
| 8 | **0.8525** | **0.8525** | 0.8525 | 0.8525 | 0.1 | 0.1 | 0.1 | 11.9 | 173.7 |
| 9 | 0.5872 | 0.5833 | 0.5833 | 0.5833 | 1.2 | 0.4 | 0.8 | 216.2 | 1198.1 |
| 10 | **0.7059** | **0.7059** | 0.7059 | 0.7059 | 0.1 | 0.0 | 0.0 | 2.3 | 9.1 |
| 11 | **0.9200** | **0.9200** | 0.9200 | 0.9200 | 0.1 | 0.0 | 0.1 | 6.4 | 64.2 |
| 12 | 0.6986 | 0.6533 | 0.6026 | 0.6313 | 1.1 | 0.3 | 0.7 | 126.7 | 1379.5 |
| 13 | **0.6933** | **0.6933** | 0.6282 | 0.6776 | 0.7 | 0.2 | 0.5 | 68.6 | 857.9 |
| 14 | 0.5258 | 0.4851 | 0.4112 | 0.4524 | 0.2 | 0.1 | 0.1 | 44.4 | 191.2 |
| 15 | **0.6783** | **0.6783** | 0.6783 | 0.6783 | 1.6 | 0.5 | 1.1 | 164.6 | 1692.0 |
| 16 | **0.5486** | **0.5486** | 0.5353 | 0.5426 | 3.3 | 0.9 | 2.4 | 273.2 | 2505.2 |
| 17 | 0.5446 | 0.5273 | 0.3966 | 0.4547 | 0.2 | 0.1 | 0.1 | 39.8 | 193.6 |
| 18 | 0.4296 | 0.4109 | 0.3769 | 0.3987 | 0.2 | 0.1 | 0.1 | 40.1 | 257.4 |
| 19 | 0.4965 | 0.4887 | 0.3778 | 0.4564 | 1.0 | 0.3 | 0.7 | 128.3 | 1163.3 |
| 20 | 0.7622 | 0.7614 | 0.7614 | 0.7614 | 3.1 | 0.8 | 2.3 | 148.9 | 2625.2 |
| 21 | **0.5807** | **0.5815** | 0.5397 | 0.5547 | 1.1 | 0.3 | 0.7 | 107.9 | 1206.1 |
| 22 | **1.0000** | **1.0000** | 1.0000 | 1.0000 | 0.3 | 0.1 | 0.2 | 14.8 | 421.1 |
| 23 | **0.8511** | **0.8511** | 0.8511 | 0.8511 | 0.7 | 0.2 | 0.5 | 40.1 | 1046.1 |
| 24 | **0.7351** | **0.7351** | 0.7351 | 0.7351 | 0.7 | 0.3 | 0.5 | 82.0 | 1093.1 |
| 25 | 0.5197 | 0.5063 | 0.4383 | 0.4730 | 0.4 | 0.1 | 0.3 | 27.8 | 494.3 |
| 26 | 0.4706 | 0.4545 | 0.4167 | 0.4422 | 0.8 | 0.3 | 0.5 | 109.4 | 1103.2 |
| 27 | 0.4487 | 0.4348 | 0.3765 | 0.4217 | 0.3 | 0.1 | 0.2 | 45.9 | 408.2 |
| 28 | 0.5427 | 0.5180 | 0.4448 | 0.5010 | 1.2 | 0.3 | 0.8 | 158.9 | 1100.7 |
| 29 | 0.4462 | 0.4367 | 0.3577 | 0.4025 | 4.2 | 0.6 | 3.5 | 215.2 | 2288.7 |
| 30 | 0.5848 | 0.5283 | 0.4667 | 0.4950 | 1.5 | 0.2 | 1.2 | 41.5 | 1090.4 |
| 31 | **0.5966** | **0.5966** | 0.5525 | 0.5764 | 0.8 | 0.2 | 0.6 | 73.3 | 915.8 |
| 32 | 0.5051 | 0.5025 | 0.4697 | 0.4865 | 2.6 | 0.8 | 1.8 | 216.5 | 2107.5 |
| 33 | **0.5642** | **0.5672** | 0.5616 | 0.5633 | 48.5 | 17.5 | 31.0 | 284.7 | 7638.0 |
| 34 | **0.7571** | **0.7571** | 0.7571 | 0.7571 | 1.8 | 0.5 | 1.3 | 102.9 | 1960.0 |
| Avg | 0.6438 | 0.6360 | 0.6048 | 0.6232 | 2.3 | 0.8 | 1.6 | 88.2 | 1092.3 |

**Table 3** Results obtained using Jaccard distance

| Inst | Efficacy | | | | Time (sec) | | | Loop | Cols |
|---|---|---|---|---|---|---|---|---|---|
| | Liter | Max | Min | Avg | Tot | CG | IP | | |
| 1 | **0.7368** | **0.7368** | 0.7368 | 0.7368 | 0.8 | 0.6 | 0.2 | 300.0 | 993.6 |
| 2 | **0.6250** | **0.6250** | 0.6087 | 0.6120 | 0.5 | 0.4 | 0.1 | 270.0 | 482.2 |
| 3 | **0.7959** | **0.7959** | 0.7959 | 0.7959 | 4.7 | 1.1 | 3.6 | 300.0 | 3658.0 |
| 4 | **0.7692** | **0.7692** | 0.7692 | 0.7692 | 0.9 | 0.6 | 0.3 | 240.6 | 1207.2 |
| 5 | **0.5313** | **0.5313** | 0.4848 | 0.5205 | 0.9 | 0.7 | 0.3 | 300.0 | 1568.2 |
| 6 | **0.7037** | **0.7037** | 0.7037 | 0.7037 | 0.6 | 0.4 | 0.2 | 270.3 | 971.8 |
| 7 | **0.6830** | **0.6830** | 0.6830 | 0.6830 | 1.1 | 0.7 | 0.4 | 300.0 | 1662.6 |
| 8 | **0.8525** | **0.8525** | 0.8525 | 0.8525 | 1.5 | 0.6 | 0.9 | 271.6 | 1859.6 |
| 9 | **0.5872** | **0.5872** | 0.5766 | 0.5815 | 2.1 | 0.8 | 1.3 | 300.0 | 2000.3 |
| 10 | **0.7059** | **0.7059** | 0.7059 | 0.7059 | 0.5 | 0.4 | 0.1 | 152.0 | 582.2 |
| 11 | **0.9200** | **0.9200** | 0.9200 | 0.9200 | 0.5 | 0.2 | 0.3 | 65.1 | 658.2 |
| 12 | 0.6986 | 0.6625 | 0.5062 | 0.6357 | 1.8 | 0.7 | 1.1 | 243.2 | 1980.1 |
| 13 | **0.6933** | **0.6933** | 0.6711 | 0.6853 | 1.4 | 0.7 | 0.7 | 272.1 | 1522.6 |
| 14 | 0.5258 | 0.5152 | 0.4356 | 0.4822 | 2.1 | 0.8 | 1.3 | 300.0 | 3094.6 |
| 15 | **0.6783** | **0.6783** | 0.6783 | 0.6783 | 2.1 | 0.9 | 1.2 | 300.0 | 2329.9 |
| 16 | **0.5486** | **0.5486** | 0.5486 | 0.5486 | 7.1 | 1.8 | 5.3 | 300.0 | 4908.0 |
| 17 | 0.5446 | 0.5321 | 0.4727 | 0.4982 | 2.0 | 0.8 | 1.2 | 300.0 | 2684.1 |
| 18 | 0.4296 | 0.4074 | 0.3688 | 0.3959 | 1.4 | 0.7 | 0.6 | 300.0 | 2021.3 |
| 19 | 0.4965 | 0.4545 | 0.4514 | 0.4520 | 1.2 | 0.7 | 0.4 | 300.0 | 1295.5 |
| 20 | 0.7622 | 0.7614 | 0.7614 | 0.7614 | 6.6 | 1.8 | 4.8 | 298.3 | 4898.0 |
| 21 | **0.5807** | **0.5815** | 0.5285 | 0.5691 | 3.4 | 1.0 | 2.5 | 300.0 | 3693.3 |
| 22 | **1.0000** | **1.0000** | 1.0000 | 1.0000 | 0.3 | 0.1 | 0.2 | 10.7 | 385.6 |
| 23 | **0.8511** | **0.8511** | 0.8511 | 0.8511 | 4.5 | 1.1 | 3.4 | 203.5 | 4034.2 |
| 24 | **0.7351** | **0.7351** | 0.7351 | 0.7351 | 3.5 | 1.1 | 2.3 | 297.7 | 3835.6 |
| 25 | 0.5197 | 0.5188 | 0.4417 | 0.4905 | 3.9 | 1.1 | 2.9 | 298.7 | 4604.0 |
| 26 | 0.4706 | 0.4277 | 0.3631 | 0.4042 | 4.2 | 1.1 | 3.2 | 297.5 | 4320.9 |
| 27 | 0.4487 | 0.4251 | 0.3869 | 0.4070 | 3.9 | 1.1 | 2.9 | 300.0 | 4265.4 |
| 28 | 0.5427 | 0.5228 | 0.4729 | 0.4992 | 1.6 | 0.7 | 0.9 | 300.0 | 1333.3 |
| 29 | 0.4462 | 0.4309 | 0.3690 | 0.4014 | 6.4 | 1.3 | 5.1 | 296.1 | 4708.3 |
| 30 | 0.5848 | 0.5370 | 0.4798 | 0.5094 | 5.4 | 1.3 | 4.1 | 300.0 | 5917.7 |
| 31 | **0.5966** | **0.5966** | 0.5385 | 0.5700 | 4.1 | 1.1 | 3.1 | 300.0 | 3959.5 |
| 32 | 0.5051 | 0.5025 | 0.4488 | 0.4696 | 4.3 | 1.2 | 3.2 | 299.0 | 4059.4 |
| 33 | **0.5642** | **0.5672** | 0.5672 | 0.5672 | 56.4 | 17.8 | 38.5 | 300.0 | 8698.3 |
| 34 | **0.7571** | **0.7571** | 0.7571 | 0.7571 | 6.7 | 1.8 | 4.8 | 290.5 | 4797.6 |
| Avg | 0.6438 | 0.6358 | 0.6080 | 0.6250 | 4.4 | 1.4 | 3.0 | 269.9 | 2911.5 |

## References


Askin RG, Subramanian S (1987) A cost-based heuristic for group technology configuration. International Journal of Production Research 25:101–113.

Boctor F (1991) A linear formulation of the machine-part cell formation problem. International Journal of Production Research 29(2):343-356.

Boe W, Cheng CH (1991) A close neighbor algorithm for designing cellular manufacturing systems. International Journal of Production Research 29(10):2097–2116.

Burbidge JL (1963) Production flow analysis. Production Engineer 42:742-752.

Burbidge JL (1969) An introduction of group technology. In: Seminar on group Technology, Turin.

Carrie S. Numerical taxonomy applied to group technology and plant layout. International Journal of Production Research; 1973; 11:399–416.

Chan HM, Milner DA (1982) Direct clustering algorithm for group formation in cellular manufacture. Journal of Manufacturing System 1:65–75.

Chandrasekharan MP, Rajagopalan R (1986a) An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. International Journal of Production Research 24(2):451-464.

Chandrashekharan MP, Rajagopalan R (1986b) MODROC: An extension of rank order clustering for group technology. International Journal of Production Research 24(5):1221–1233.

Chandrasekharan MP, Rajagopalan R (1989) Groupability: Analysis of the properties of binary data matrices for group technology. International Journal of Production Research 27(6):1035-1052.

Dantzig G, Wolfe P (1960) Decomposition Principle for Linear Programs. Operations Research 8:101-111.

Gonçalves JF, Resende MGC (2004) An evolutionary algorithm for manufacturing cell formation. Computers and Industrial Engineering 47: 247-273.

Guerrero F, Lozano S, Smith KA, Canca D, Kwok T (2002) Manufacturing cell formation using a new self-organizing neural network. Computers and Industrial Engineering 42: 377-382.

Gupta T, Seifoddini H (1990) Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system. International Journal of Production Research 28(7):1247-1269.

ILOG (2006) ILOG CPLEX 10.0: user's manual.

Jayakrishnan NJG, Narendran TT (1998) CASE: a clustering algorithm for cell formation with sequence data. International Journal of Production Research 36(1): 157-179.

Jaumard B, Labit P, Ribeiro CC (1999) A column generation approach to cell formation problems in cellular manufacturing. Le Cahiers du GERAD pp.99-20.

Joines JA, King RE, Culbreth CT (1995) A comprehensive review of production-oriented manufacturing cell formation techniques. International Journal of Flexible Automation and Intelligent Manufacturing 3:254-264.

King JR, Nakornchai V. Machine-component group formation in group technology: Review and extension. International Journal of Production Research; 1982; 20(2):117–133.

King JR (1980) Machine-component grouping formation in group technology. International Journal of management Science 8(2):193-199.

Kumar KR., Kusiak A, Vannelli A (1986) Grouping of parts and components in flexible manufacturing systems. European Journal of Operations Research 24:387–397.

Kumar KR, Vannelli A (1987) Strategic subcontracting for efficient disaggregated manufacturing. International Journal of Production Research 25(12):1715–1728.

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

Kusiak A (1987) The generalized group technology concept, International Journal of Production Research 25(4):561-569.

Kusiak A, Cho M (1992) Similarity coefficient algorithm for solving the group technology problem. InternationalJournal of Production Research 30:2633-2646.

Lei D, Wu Z (2006) Tabu search for multiple-criteria manufacturing cell design. International Journal of Advanced Manufacturing Technology 28: 950-956.

Lin TL, Dessouky MM, Kumar KR, Ng SM (1996) A heuristic based procedure for the weighted production - cell formation problem. IIE Transactions 28:579-589.

Malave CO, Ramachandran A (1991) A neural network based design of cellular manufacturing system. Journal of Intelligent Manufacturing 2:305-314.

McCormick Jr WT, Schweitzer PJ, White TW (1972) Problem decomposition and data reorganization by a cluster technique. Operations Research 20(5):993-1009.

Mosier CT, Taube L (1985a) The facets of group technology and their impact on implementation, OMEGA 13(6):381–391.

Mosier CT, Taube L (1985b) Weighted similarity measure heuristics for the group technology machine clustering problem. OMEGA 13(6):577–583.

Papaioannou G, Wilson JM (2008) Fuzzy extensions to integer programming models of cell-formation problems in machine scheduling. Annals of Operations Research.

Rajagopalan R, Batra JL (1975) Design of cellular production systems: A graph theoretic approach. International Journal of Production Research 13(6):567-579.

Ribeiro Filho G, Lorena LAN (2000) A Constructive Evolutionary Approach to the Machine-Part Cell Formation Problem . In: Fleury A, Yoshisaki H, Guimaraes LBM, Ribeiro JLD, editors. Buildings Competencies for International Manufacturing - Perspectives for developing countries. Porto Alegre: UFRGS/FEENG pp.340-348.

Seifoddini H (1989) Single linkage versus average linkage clustering in machine cells formation applications. Computers and Industrial Engineering 16(3):419–426.

Seifoddini H, Wolfe PM (1986) Application of the similarity coefficient method in group technology. IIE Transactions 18(3):266-270.

Senne ELF, Lorena LAN, Pereira MA (2007) A simple stabilizing method for column generation heuristics: an application to p-median location problems. International Journal of Operations Research 4:1-9.

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

Stanfel L (1985) Machine clustering for economic production. Engineering Costs and Production Economics 9:73-8.

Torkul O, Cedimoglou IH, Geyik AK (2006) An application of fuzzy clustering to manufacturing cell design. Journal of Intellegent & Fuzzy Systems 17:173-181.

Venugopal V, Narendran TT (1992a) Cell formation in manufacturing systems through simulated annealing: an experimental evaluation. European Journal of Operational Research 63(3):409-422.

Venugopal V, Narendran TT (1992b) A genetic algorithm approach to the machine-component grouping problem with multiple objectives. Computers and Industrial Engineering22(4):469-480.

Venugopal V, Narendran TT (1993) Design of cellular manufacturing systems based on asymptotic forms of a Boolean matrix. European Journal of Operational Research 67:405-417.

Waghodekar PH, Sahu S (1984) Machine-component cell formation in group technology MACE. Lnternational Journal of Production Research 22:937-948.

Won Y, Currie KR (2004) Efficient p-median mathematical programming approaches to machine-part grouping in group technology manufacturing. Engineering Optimization 36(5):555–573.

Xu H, Wang HP (1989) Part family formation for gt applications based on fuzzy mathematics. International Journal of Production Research 27(9):1637-1651.