Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Computers & Industrial Engineering 47 (2004) 247–273

computers & industrial engineering

www.elsevier.com/locate/dsw

An evolutionary algorithm for manufacturing cell formation☆

José Fernando Gonçalves[a], Mauricio G.C. Resende[b,*]

[a]*Faculdade de Economia do Porto, Rua Dr Roberto Frias, 4200-464 Porto, Portugal*
[b]*Internet and Network Systems Research, AT&T Labs Research, 180 Park Avenue, Bldg. 103, Room C241, Florham Park, NJ 07932, USA*

Received 8 January 2003; accepted 14 July 2004
Available online 11 September 2004

Abstract

Cellular manufacturing emerged as a production strategy capable of solving the certain problems of complexity and long manufacturing lead times in batch production. The fundamental problem in cellular manufacturing is the formation of product families and machine cells. This paper presents a new approach for obtaining machine cells and product families. The approach combines a local search heuristic with a genetic algorithm. Computational experience with the algorithm on a set of group technology problems available in the literature is also presented. The approach produced solutions with a grouping efficacy that is at least as good as any results previously reported in literature and improved the grouping efficacy for 59% of the problems.

Keywords: Cellular manufacturing; Group technology; Genetic algorithms; Random keys

1. Introduction

Cellular manufacturing emerged as a production strategy capable of solving certain problems of complexity and long manufacturing lead times in batch production systems in the beginning of the 1960s. Burbidge (1979) defined group technology (GT) as an approach to the optimization of work in which the organizational production units are relatively independent groups, each responsible for the production of a given family of products.

☆ AT & T Labs Research Technical Report TD-5FE6RN, October 29, 2002. Revised February 13, 2004.

* Corresponding author. Tel.: +1-973-360-8444; fax: +1-973-360-8178.
E-mail addresses: jfgoncal@fep.up.pt (J.F. Gonçalves), mgcr@research.att.com (M.G.C. Resende).

0360-8352/$ - see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.cie.2004.07.003

One fundamental problem in cellular manufacturing is the formation of product families and machine cells. The objective of this product-machine grouping problem is to form perfect (i.e. disjoint) groups in which products do not have to move from one cell to the other for processing.

At the conceptual level cell formation models ignore many manufacturing factors and only consider the machining operations of the products, so that a manufacturing system is represented by a binary machine-part incidence matrix [A], which is a zero-one matrix of order $\boldsymbol{P}\times\boldsymbol{M}$ where $\boldsymbol{P}$=number of products and $\boldsymbol{M}$=number of machines. Element $\boldsymbol{a_{p,m}}=1$ indicates the visit of product p to machine $\boldsymbol{M}$ and $\boldsymbol{a_{p,m}}=0$ indicates otherwise.

Many methods of cell formation have been developed and published. Wemmerlov and Hyer (1989) and Selim, Askin, and Vakharia (1998) provide extensive reviews of prior research. In the next subsections we briefly review procedures based on the type of general solution methodology used (Cluster analysis, Graph partitioning, Mathematical programming and other.)

1.1. Procedures based on cluster analysis

Array-based clustering methods perform a series of column and row permutations to form product and machine cells simultaneously. King (1980) and later King and Nakornchai (1982) developed array-based methods. Chandrasekharan and Rajagopalan (1987), Khator and Irani (1987), King and Nakornchai (1982), and Kusiak and Chow (1987) proposed other algorithms. A comprehensive comparison of three array-based clustering techniques is given in Chu and Tsai (1990). The quality of the solution given by these methods depends on the initial configuration of the zero-one matrix.

McAuley (1972) and Carrie (1973) developed algorithms using clustering and similarity coefficients. Since then, Gupta and Seifoddini (1990), Khan, Islam, and Sarker (2000), Mosier and Taube (1985a,b), Seifoddini (1989), and Yasuda and Yin (2001) proposed hierarchical methods. These methods have the disadvantage of not forming product and machine cells simultaneously, so additional methods must be employed to complete the design of the system.

GRAFICS, developed by Srinivasan and Narendran (1991), and ZODIAC, which is a modular version of MacQueen's clustering method, developed by Chandrasekharan and Rajagopalan (1987), are examples of non-hierarchical methods. Miltenburg and Zhang (1991) present a comprehensive comparison of nine clustering methods where non-hierarchical methods outperform both array-based and hierarchical methods.

1.2. Graph partitioning approaches

Rajagopalan and Batra (1975) used graph theory to solve the grouping problem. They developed a machine graph with as many vertices as the number of machines. Two vertices were connected by an edge if there were parts requiring processing on both the machines. Cliques obtained from the graph were used to determine machine cells. The limitation of this method is that machine cells and part families are not formed simultaneously. Kumar et al. (1986) solved a graph decomposition problem to determine machine cells and part families for a fixed number of groups and with bounds on cell size. Their algorithm for grouping in flexible manufacturing systems is also applicable in the context of GT.

Vannelli and Kumar (1986) developed graph theoretic models to determine machines to be duplicated so that a perfect block diagonal structure can be obtained. Kumar and Vannelli (1987) developed a similar procedure for determining parts to be subcontracted in order to obtain a perfect block diagonal structure. These methods are found to depend on the initial pivot element choice.

Vohra et al. (1990) suggested a network-based approach to solve the grouping problem. They used a modified form of the Gomory-Hu algorithm to decompose the part-machine graph. Askin, Creswell, Goldberg, and Vakharia (1991) proposed a Hamiltonian-path algorithm for the grouping problem. The algorithm heuristically solves the distance matrix for machines as a TSP and finds a Hamiltonian path that gives the rearranged rows in the block diagonal structure. The disadvantage of this approach is that actual machine groups are not evident from its solution. Lee and Garcia-Diaz (1993) transformed the cell formation problem into a network flow formulation and used a primal-dual algorithm developed by Bertsekas and Tseng (1988) to determine the machine cells. Other graph approaches include the heuristic graph partitioning approach of Askin and Chiu (1990) and the minimum spanning tree approach of Ng (1993) and (1996).

1.3. Mathematical programming approaches

Mathematical programming methods treat the clustering problem as a mathematical programming optimization problem. Different objective models have been used. Kusiak (1987) suggested the *p*-median model for GT, where it minimizes the total sum of distances between each product/machine pair. Shtub (1989) modeled the grouping problem as a generalized assignment problem. Choobineh (1988) formulated an integer programming problem which first determines product families and then assigns product families to cells with an objective of minimizing costs. Co and Araar (1988) developed a three-stage procedure to form cells and solved an assignment problem to assign jobs to machines. Gunasingh and Lashkari (1989) formulated an integer programming problem to group machines and products for cellular manufacturing systems. Srinivasan, Narendran, and Mahadevan (1990) modeled the problem as an assignment problem to obtain product and machine cells Chen and Heragu (1999) present two stepwise decomposition approaches to solve large-scale industrial problems. Won (2000) presents a two-phase methodology based on an efficient *p*-median approach. Akturk and Turkcan (2000) propose an integrated algorithm that solves the machine/product grouping problem by simultaneously considering the within-cell layout problem.

1.4. Other approaches

Joines, Culbreth, and King (1996) developed an integer program that is solved using a genetic algorithm. Cheng, Gupta, Lee, and Wond (1998) formulate the problem as a traveling salesman problem and solve the model using a genetic algorithm. Plaquin and Pierreval (2000) propose an evolutionary algorithm for cell formation taking into account specific constraints. Zhao and Wu (2000) present a genetic algorithm for cell formation with multiple routes and objectives. Caux, Brauniaux, and Pierreval (2000) address the cell formation problem with multiple process plans and capacity constraints using a simulated annealing approach. Dimopoulos and Mort (2001) present a hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems. Onwubolu and Mutingi (2001) develop a genetic algorithm approach taking into account cell-load variation. Brown and Sumichrast (2001) propose an approach using Grouping Genetic Algorithm (GGA). Uddin and Shanker (2002) address a generalized grouping problem, where each part has more than one process route. The problem is formulated as an integer programming problem and a procedure based on a genetic algorithm is suggested as a solution methodology.

The cell formation problem is a combinatorial optimization problem that is NP-hard and therefore optimization algorithms yield a globally optimal solution in a prohibitive computation time. None of the approaches presented above guarantees optimal solutions. Metaheuristics have emerged to solve combinatorial optimization problems with global or near-global optimal solution in a reasonable computation time.

The objective of this paper is to present a procedure for obtaining product-machine groupings when the manufacturing system is represented by a binary product-machine incidence matrix. The approach combines a genetic algorithm with a local search heuristic. The genetic algorithm is responsible for generating sets of machines cells. The local search heuristic is applied on the set of machines cells with the objective of constructing sets of machine/product groups and improving their quality.

In Section 2, we present the problem in terms of a block diagonalization problem. Some measures of grouping quality are discussed in Section 3. Section 4 presents the local search procedure and the genetic algorithm. The performance of the approach, on a set of 35 GT problems available in the literature, is shown in Section 5. In Section 6, concluding remarks are made.

2. The block diagonalization problem

In this paper, we attempt to solve the machine and product-grouping problem as a zero one block diagonalization problem (BDP), to minimize inter-cellular movement and maximize the utilization of the machines within a cell.

Fig. 1 presents an example of the block diagonalization process of a 15×12 matrix (the zero values were replaced by spaces in order to make the figure more readable). In this case (see Fig. 1(a)), there are 15 products ($p=1,2,\ldots,15$) to be produced in a set of 12 machines ($M=M_1,M_2,\ldots,M_{12}$). The objective of the diagonalization problem is to produce a matrix such as the one in Fig. 1(b).

As can be observed in Fig. 1(b), four product/machine groups were formed see Table 1.

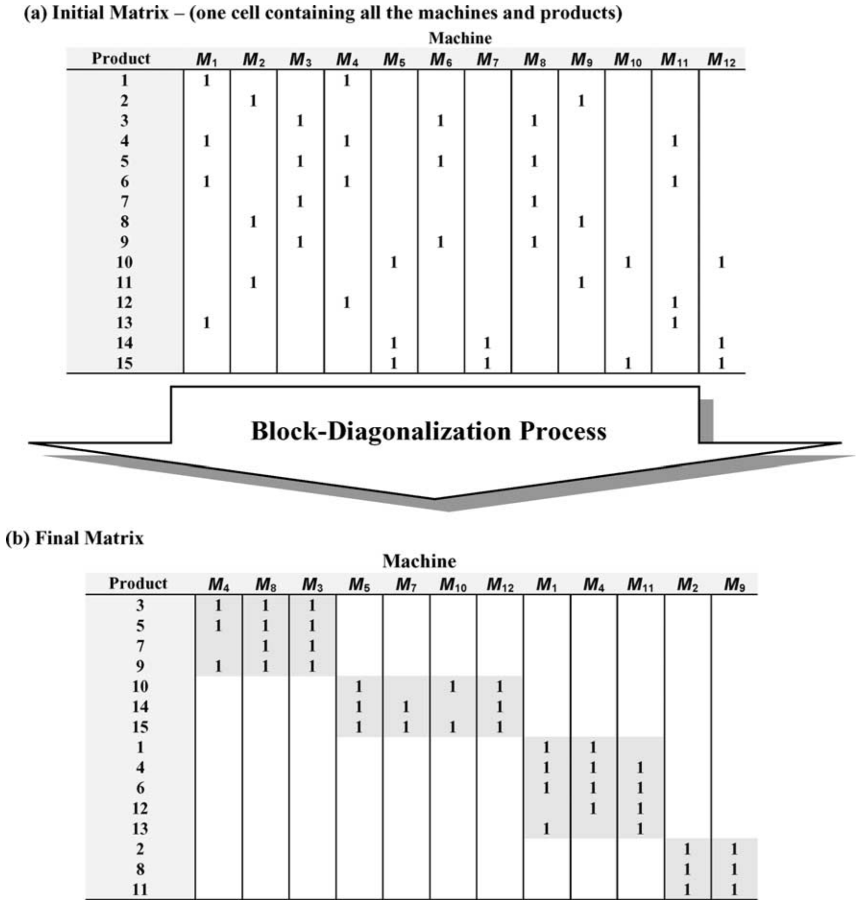
3. Measure of performance

Several measures of goodness of machine-product groups in cellular manufacturing have been proposed. Sarker and Mondal (1999) present a simulation study of the effects of several factors on the efficiency measures. Sarker (2001) introduces a new measure of goodness of machine-product grouping and presents a survey of existing measures. The grouping efficiency and grouping efficacy are two popular grouping measures because they are simple to implement and generate block diagonal matrices. Grouping efficiency was first proposed by Chandrasekharan and Rajagopolan (1989). It incorporates both machine utilization and inter-cell movement and is defined as the weighted sum of two functions η_1 and η_2:

$$\text{Grouping efficiency} = \eta = q\eta_1 + (1-q)\eta_2$$

where

- η_1 ratio of the number of 1's in the diagonal blocks to the total number of elements in the diagonal blocks of the final matrix;
- η_{12} ratio of the number of 0's in the off-diagonal blocks to the total number of elements in the off-diagonal blocks of the final matrix;
- q weight factor.



(a) Initial Matrix – (one cell containing all the machines and products)

Product	Machine M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}
1	1			1								
2		1							1			
3			1			1		1				
4	1			1							1	
5			1			1		1				
6	1			1							1	
7			1					1				
8		1							1			
9			1			1		1				
10					1					1		1
11		1							1			
12				1							1	
13	1										1	
14					1		1					1
15					1		1			1		1

Block-Diagonalization Process

(b) Final Matrix

Product	Machine M_4	M_8	M_3	M_5	M_7	M_{10}	M_{12}	M_1	M_4	M_{11}	M_2	M_9
3	1	1	1									
5	1	1	1									
7		1	1									
9	1	1	1									
10				1		1	1					
14				1	1		1					
15				1	1	1	1					
1								1	1			
4								1	1	1		
6								1	1	1		
12									1	1		
13								1		1		
2											1	1
8											1	1
11											1	1

Fig. 1. Block diagonalization example.

One drawback of grouping efficiency is the low discriminating capability (i.e. the ability to distinguish good quality grouping from bad). For example, a bad solution with many 1's in the off-diagonal blocks often shows efficiency figures around 75%. When the matrix size increases, the effect of 1's in the off-diagonal blocks becomes smaller, and in some cases, the effect of inter-cell moves is not reflected in grouping efficiency. To overcome the low discriminating power of grouping efficiency between well-structured and ill-structured incidence matrices, Kumar and Chandrasekharan (1990) proposed another measure, which they call grouping efficacy. Unlike grouping efficiency, grouping efficacy is not affected by the size of the matrix.

Table 1
Resulting product/machine groups for the example in Fig. 1b

Cells	Machines	Products
1	$\boldsymbol{M}_3$, $\boldsymbol{M}_6$, $\boldsymbol{M}_8$	3, 5, 7, 9
2	$\boldsymbol{M}_5$, $\boldsymbol{M}_7$, $\boldsymbol{M}_{10}$, $\boldsymbol{M}_{12}$	10, 14, 15
3	$\boldsymbol{M}_1$, $\boldsymbol{M}_4$, $\boldsymbol{M}_{11}$	1, 4, 6, 12, 13
4	$\boldsymbol{M}_2$, $\boldsymbol{M}_9$	2, 8, 11

The grouping efficacy can be defined as

$$\text{Grouping efficacy} = \mu = \frac{\boldsymbol{N}_1 - \boldsymbol{N}_1^{\text{Out}}}{\boldsymbol{N}_1 + \boldsymbol{N}_0^{\text{In}}}, \tag{1}$$

where

N_1 total number of 1's in matrix A;
$\boldsymbol{N}_1^{\text{Out}}$ total number of 1's outside the diagonal blocks;
$\boldsymbol{N}_0^{\text{In}}$ total number of 0's inside the diagonal blocks.

The closer the grouping efficacy is to 1, the better will be the grouping. The grouping efficacy for the matrices in Fig. 1(a) (one group containing all the machines and products) and (b) are, respectively,

$$\mu_a = \frac{39 - 0}{39 + 141} = 21.67\%$$

$$\mu_b = \frac{39 - 0}{39 + 6} = 86.67\%$$

As expected, the matrix in Fig. 1(b) has a much higher grouping efficacy than the one in Fig. 1(a).

We chose grouping efficacy as the measure of performance for the hybrid genetic algorithm proposed in this paper for several reasons:

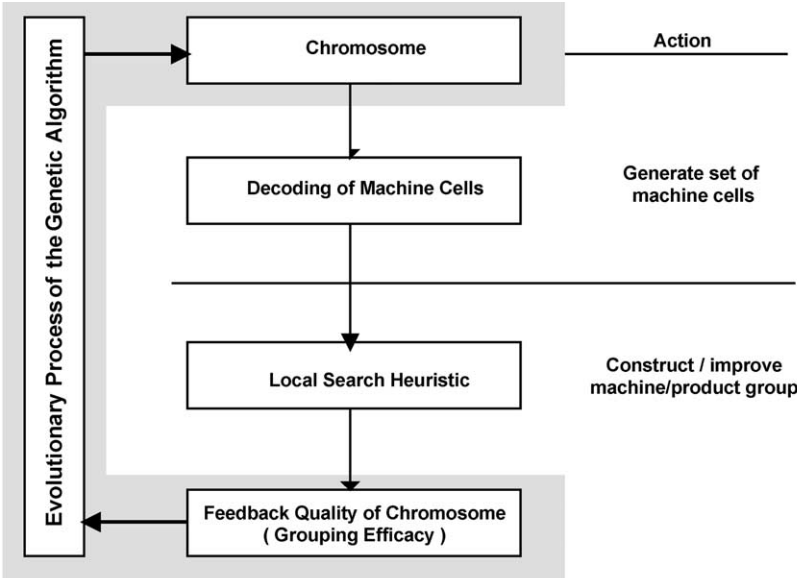
- In the literature it has been considered the standard measure to report the quality of the grouping solutions.
- It is considered a better measure than the grouping efficiency.
- It is able to incorporate both the within-cell machine utilization and the inter-cell movement.
- It has a high capability to differentiate between well-structured and ill-structured matrices (high discriminating power).
- It generates block diagonal matrices which are attractive in practice.
- It does not require a weight factor.

4. The new approach

The approach presented in this paper combines a genetic algorithm with a local search heuristic. The genetic algorithm is used to generate sets of machine cells. The evolutionary process, embedded in the genetic algorithm, is responsible for improving the grouping quality of the sets of machine cells generated. The local search heuristic is applied to the sets of machines cells generated by the genetic algorithm. The objective of the heuristic is to construct a set of machine/product groups and improve it, if possible. The heuristic feeds back to the genetic algorithm the grouping efficacy of the set of machine/product groups it constructs. Fig. 2a shows the sequence of steps applied to each chromosome generated by the genetic algorithm.

The remainder of this section describes in detail the genetic algorithm and the local search heuristic.

(a)



(b)

Genetic algorithm

{

  Generate initial population P_t

  Evaluate population P_t

  while stopping criteria not satisfied **repeat**

  {

    Select some elements from P_t to put into P_{t+1}

    Crossover some elements of P_t and put them into P_{t+1}

    Mutate some elements of P_t and put them into P_{t+1}

    Evaluate new population P_{t+1}

    $P_t = P_{t+1}$

  }

}

Fig. 2. (a) Architecture of the new approach. (b) Standard genetic algorithm.

4.1. Genetic algorithm

Genetic algorithms (GA) were introduced by Holland (1975) and have been applied in a number of fields, e.g. mathematics, engineering, biology, and social science (Goldberg, 1989). GAs are search algorithms based on the mechanics of natural selection and natural genetics. They combine the concept of survival of the fittest with structured, yet randomized, information exchange to form robust search algorithms.

The concept of genetic algorithms is based on the evolution process that occurs in natural biology. An initial population of possible solutions (referred to as individuals or chromosomes) is generated. Some individuals are selected to be parents to produce offspring via a crossover operator. All the individuals are then evaluated and selected based on Darwin's concept of survival of the fittest. The process of reproduction, evaluation, and selection is repeated until a termination criterion is reached. In addition, a mutation operator with certain probability is applied to the individuals to change their genetic makeup. The objective of this mutation process is to increase the diversity of the population and ensure an extensive search.

Each iteration (also referred to as generation or family of solutions) is made up of chromosomes. Each chromosome is in turn made up of individual genes. These genes are encodings of the design variables that are used to evaluate the function being optimized. In each iteration of the search process, the system has a fixed population of chromosomes that represent the current solutions to the problem. Fig. 2b represents a pseudo-code for a standard genetic algorithm.

The GA calls a subroutine to compute the fitness value (the quality) for each chromosome in the population. This fitness value is the only feedback to the GA.

The genetic algorithm presented in this paper uses a random key alphabet $U(0,1)$ and an evolutionary strategy (see Fig. 5) identical to the one proposed by Bean (1994). An important feature of random keys is that all offspring formed by crossover are feasible solutions. This is accomplished by moving much of the feasibility issue into the fitness evaluation procedure. If any random key vector can be interpreted as a feasible solution, then any crossover is feasible. Through the dynamics of the genetic algorithm, the system learns the relationship between random key vectors and solutions with good objective values.

As mentioned earlier, the fitness function used is the grouping efficacy. The other important aspects of genetic algorithms: chromosomal representation and decoding, parent selection, crossover, and mutation will be discussed next

4.1.1. Chromosomal representation and decoding

A chromosome represents a solution to the problem and is encoded as a vector of random keys (random numbers). Each chromosome is made of $\boldsymbol{M}+1$ genes where $\boldsymbol{M}$ is the number of machines:

Chromosome = $(\text{gene}_1,\ \text{gene}_2, \ldots,\ \text{gene}_M,\ \text{gene}_{M+1})$.

The $M+1$st gene is used to determine the number of machine cells and uses the following decoding expression

$$n\text{Cells} = \lfloor \text{gene}_{M+1} \times M \rfloor,$$

where $(x($ is the smallest integer larger than x. Genes 1 through $\boldsymbol{M}$are used to determine the assignment of machines to machine cells and use the following decoding expression

$$\text{Cell}_i = \lfloor \text{gene}_i \times n\text{Cells} \rfloor \quad i = 1, \ldots, M.$$

Fig. 3 presents an example of the decoding of a chromosome.

4.1.2. Reproduction, crossover, and mutation

Many variants of genetic algorithms are formed by varying the reproduction, crossover, and mutation operators. The reproduction and crossover operators determine which parents will have offspring, and how the genetic material is exchanged between the parents to create those offspring. Mutation allows for random alteration of genetic material. Reproduction and crossover operators tend to increase the quality of the populations and force convergence. Mutation opposes convergence and replaces genetic material lost during reproduction and crossover.

Reproduction is accomplished by copying the best individuals from one generation to the next, in what is often called an elitist strategy (Goldberg, 1989). The advantage of an elitist strategy over traditional probabilistic reproduction is that the best solution is monotonically improving from one

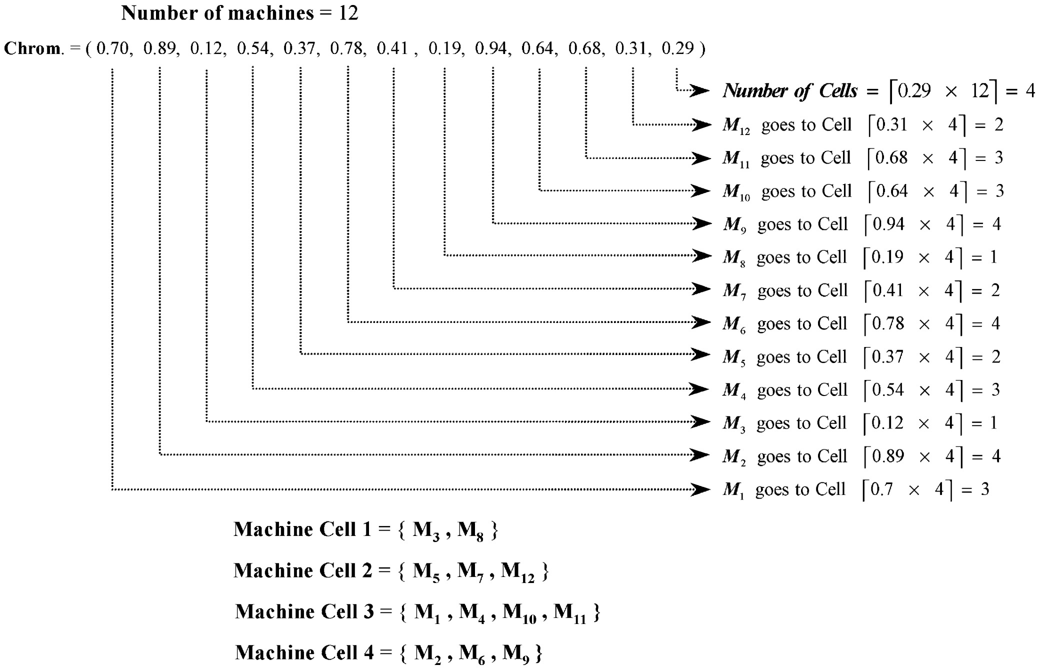


Fig. 3. Example of the decoding of a chromosome.

generation to the next. The potential downside is population convergence. This can however be overcome by high mutation rates described below.

Parameterized uniform crossovers (Spears & DeJong, 1991) are employed in place of the traditional one-point or two-point crossover. After two parents are chosen randomly from the full old population (including chromosomes copied to the next generation in the elitist pass), at each gene a biased coin is tossed to select which parent will contribute the offspring. Fig. 4 presents an example of the crossover operator. It assumes that a coin toss of heads selects the gene from the first parent, a tails chooses the gene from the second parent, and that the probability of tossing a heads is 0.7. Below is one potential crossover outcome:

To prevent premature convergence of the population, at each generation one or more new members of the population are randomly generated from the same distribution as the original population. This process has the same effect as applying at each generation the traditional gene-by-gene mutation with small probability.

All chromosomes of the first generation are randomly generated. Fig. 5 depicts the evolutionary process.

Coin toss	*H*	*H*	*T*	*H*	*T*
Parent 1	*0.57*	*0.93*	*0.36*	*0.12*	*0.78*
Parent 2	*0.46*	*0.35*	*0.59*	*0.89*	*0.23*
Offspring	*0.57*	*0.93*	*0.59*	*0.12*	*0.23*

Fig. 4. Example of uniform crossover.

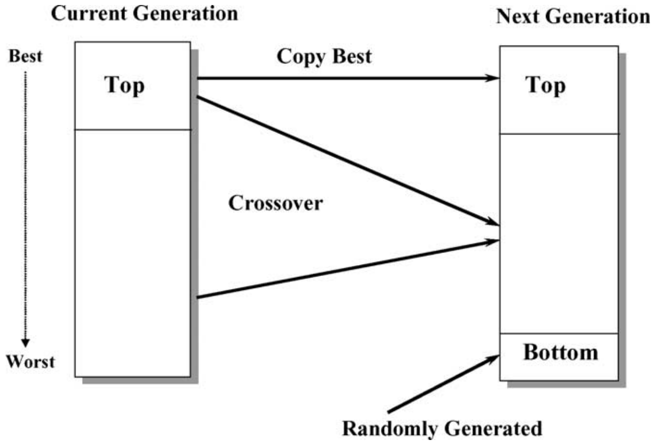


Fig. 5. Evolutionary process.

4.2. Local search heuristic

The local search heuristic is applied to the sets of machine cells generated by the genetic algorithm When the machine cells are known, it is customary to assign a product to the cell where it visits the maximum number of machines. This is optimal to minimize inter-cell movement (because it reduces the exceptional elements). However, it does not guarantee good utilization of the machines within a cell. To overcome this problem, a local search heuristic, which takes into consideration both inter-cell movement and machine utilization was developed. Srinivasan and Narendran (1991) and Adil, Rajamani, and Strong (1997) developed heuristics whose main loop is similar to ours. The main difference between our heuristic and their consists in the rule used to assign products/machines to the machines cells/product groups and in the stopping criteria.

The heuristic consists of an improvement procedure that is repeatedly applied. Each iteration k of the procedure starts with a given initial set of machine cells $\boldsymbol{M}_k^{\text{INITIAL}}$, and produces a set of product families $\boldsymbol{P}_k^{\text{FINAL}}$, and a set of machine cells $\boldsymbol{M}_k^{\text{FINAL}}$. Two block-diagonal matrices can be obtained by combining $\boldsymbol{M}_k^{\text{INITIAL}}$ with $\boldsymbol{P}_k^{\text{FINAL}}$ and $\boldsymbol{M}_k^{\text{FINAL}}$ with $\boldsymbol{P}_k^{\text{FINAL}}$. From these two matrices, the one with the highest grouping efficacy is chosen as the resulting block-diagonal matrix of the iteration k. The procedure stops if $\boldsymbol{M}_k^{\text{FINAL}} = \boldsymbol{M}_k^{\text{INITIAL}}$ 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 resulting from the previous iteration k-1, (for $k > 2$). Otherwise, the procedure sets $\boldsymbol{M}_{k+1}^{\text{INITIAL}} = \boldsymbol{M}_k^{\text{FINAL}}$ and continues to iteration $k+1$.

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

(1) Assignment of products to the initial set of machine cells $\boldsymbol{M}_k^{\text{INITIAL}}$. (Note that the initial the set of machine cells of iteration 1, $\boldsymbol{M}_1^{\text{INITIAL}}$, is supplied by the genetic algorithm). Products are assigned to machine cells one at a time (in any order). A product is assigned to the cell that maximizes an approximation of the grouping efficacy, that is, a product is assigned to the machine cell C^*, given by

$$C^* = \underset{C}{\operatorname{argmax}} \left\{ \frac{\boldsymbol{N}_1 - \boldsymbol{N}_{1,C}^{\text{Out}}}{\boldsymbol{N}_1 + \boldsymbol{N}_{0,C}^{\text{In}}} \right\},$$

where

argmax argument that maximizes expression
N_1 total number of 1's in matrix A;
N_1^{Out} total number of 1's outside the diagonal blocks if the product is assigned to cell C;
N_0^{In} total number of 0's inside the diagonal blocks if the product is assigned to cell C.

In this step, the heuristic generates a set of product families $\boldsymbol{P}_k^{\mathrm{FINAL}}$. Let μ_κ^1 be the efficacy of the block-diagonal matrix defined by $\boldsymbol{M}_k^{\mathrm{INITIAL}}$ and $\boldsymbol{P}_k^{\mathrm{FINAL}}$.

(2) Assignment of machines to the set of product families $\boldsymbol{P}_k^{\mathrm{FINAL}}$obtained in step (1). Machines are assigned to product families, one at a time (in any order). A machine is assigned to the product family that maximizes an approximation of the grouping efficacy, that is, a machine is assigned to the product family F^*, given by

$$F^* = \underset{F}{\operatorname{argmax}} \left\{ \frac{\boldsymbol{N}_1 - \boldsymbol{N}_{1,F}^{\mathrm{Out}}}{\boldsymbol{N}_1 + \boldsymbol{N}_{0,F}^{\mathrm{In}}} \right\},$$

where

argmax argument that maximizes expression
N_1 total number of 1's in matrix A;
N_1^{Out} total number of 1's outside the diagonal blocks if the product is assigned to cell F;
N_0^{In} total number of 0's inside the diagonal blocks if the product is assigned to cell F.

In this step, the local search heuristic generates a new set of machine cells $\boldsymbol{M}_k^{\mathrm{FINAL}}$. Let μ_κ^2 be the efficacy of the block-diagonal matrix defined by $\boldsymbol{M}_k^{\mathrm{FINAL}}$ and $\boldsymbol{P}_k^{\mathrm{FINAL}}$.

The block-diagonal matrix resulting from the iteration has a grouping efficacy given by $\mu_\kappa = \max(\mu_\kappa^1, \mu_\kappa^2)$. If $\boldsymbol{M}_k^{\mathrm{FINAL}} = \boldsymbol{M}_k^{\mathrm{INITIAL}}$ or $\mu_\kappa \le \mu_{\kappa-}(k \ge 2)$, then the iterative process stops and the block-diagonal matrix of iteration k-1 is the result. Otherwise, the procedure sets $\boldsymbol{M}_{k+1}^{\mathrm{INITIAL}} = \boldsymbol{M}_k^{\mathrm{FINAL}}$ and continues to step (1) of iteration $k+1$.

4.2.1. An example

Suppose we start with the initial set of machine cells given by the genetic algorithm and shown in Table 2:

Thus,

$$\boldsymbol{M}_1^{\mathrm{INITIAL}} = \{(\boldsymbol{M}_3, \boldsymbol{M}_8), (\boldsymbol{M}_5, \boldsymbol{M}_7, \boldsymbol{M}_{12}), (\boldsymbol{M}_1, \boldsymbol{M}_4, \boldsymbol{M}_{10}, \boldsymbol{M}_{11}), (\boldsymbol{M}_2, \boldsymbol{M}_6, \boldsymbol{M}_9)\}$$

'Table 2
Initial set of machine cells

Cells	Machines
1	$\boldsymbol{M}_3$, $\boldsymbol{M}_8$
2	$\boldsymbol{M}_5$, $\boldsymbol{M}_7$, $\boldsymbol{M}_{12}$
3	$\boldsymbol{M}_1$, $\boldsymbol{M}_4$, $\boldsymbol{M}_{10}$, $\boldsymbol{M}_{11}$
4	$\boldsymbol{M}_2$, $\boldsymbol{M}_6$, $\boldsymbol{M}_9$

Table 3
Computations for step 1 of the local search heuristic

Products	Product machines	Machine cells			
		(M_3, M_8)	(M_5, M_7, M_{12})	(M_1, M_4, M_{10}, M_{11})	(M_2, M_6, M_9)
		μ_C	μ_C	μ_C	μ_C
1	M_1,M_4	(39−2)/(39+2)=90.2%	(39−2)/(39+3)=88.1%	**(39−0)/(39+2)=95.1%**	(39−2)/(39+3)=88.1%
2	M_2,M_9	(39−2)/(39+2)=90.2%	(39−2)/(39+3)=88.1%	(39−2)/(39+4)=86.0%	**(39−0)/(39+1)=97.5%**
3	M_3,M_6,M_8	**(39−1)/(39+0)=97.4%**	(39−3)/(39+3)=85.7%	(39−3)/(39+4)=83.7%	(39−2)/(39+2)=90.2%
4	M_1,M_4,M_{11}	(39−3)/(39+2)=87.8%	(39−3)/(39+3)=85.7%	**(39−0)/(39+1)=97.5%**	(39−3)/(39+3)=85.7%
5	M_3,M_6,M_8	**(39−1)/(39+0)=97.4%**	(39−3)/(39+3)=85.7%	(39−3)/(39+4)=83.7%	(39−2)/(39+2)=90.2%
6	M_1,M_4,M_{11}	(39−3)/(39+2)=87.8%	(39−3)/(39+3)=85.7%	**(39−0)/(39+1)=97.5%**	(39−3)/(39+3)=85.7%
7	M_3, M_8	**(39−0)/(39+0)=100.0%**	(39−2)/(39+3)=88.1%	(39−2)/(39+4)=86.0%	(39−2)/(39+3)=88.1%
8	M_2,M_9	(39−2)/(39+2)=90.2%	(39−2)/(39+3)=88.1%	(39−2)/(39+4)=86.0%	**(39−0)/(39+1)=97.5%**
9	M_3,M_6,M_8	**(39−1)/(39+0)=97.4%**	(39−3)/(39+3)=85.7%	(39−3)/(39+4)=83.7%	(39−2)/(39+2)=90.2%
10	M_5,M_{10},M_{12}	(39−3)/(39+2)=87.8%	**(39−1)/(39+1)=95.0%**	(39−2)/(39+3)=88.1%	(39−3)/(39+3)=85.7%
11	M_2,M_9	(39−2)/(39+2)=90.2%	(39−2)/(39+3)=88.1%	(39−2)/(39+4)=86.0%	**(39−0)/(39+1)=97.5%**
12	M_4,M_{11}	(39−2)/(39+2)=90.2%	(39−2)/(39+3)=88.1%	**(39−0)/(39+2)=95.1%**	(39−2)/(39+3)=88.1%
13	M_1,M_{11}	(39−2)/(39+2)=90.2%	(39−2)/(39+3)=88.1%	**(39−0)/(39+2)=95.1%**	(39−2)/(39+3)=88.1%
14	M_5,M_7,M_{12}	(39−3)/(39+2)=87.8%	**(39−0)/(39+0)=100.0%**	(39−3)/(39+4)=83.7%	(39−3)/(39+3)=85.7%
15	M_5,M_7, M_{10},M_{12}	(39−4)/(39+2)=85.4%	**(39−1)/(39+0)=97.4%**	(39−3)/(39+3)=85.7%	(39−4)/(39+3)=83.3%

Step 1 Determining a set of product families

Table 3 presents the value of $\mu_C = \left\{\frac{N_1 - N_{1,C}^{\mathrm{Out}}}{N_1 + N_{0,C}^{\mathrm{In}}}\right\}$, for each product and each machine cell. A product is assigned to the cell with the highest value of μ_C (the cells in bold in Table 3).

Thus,

$$P_1^{\mathrm{FINAL}} = \{(3,5,7,9),(10,14,15),(1,4,6,12,13),(2,8,11)\}$$

.The resulting grouping combining $\boldsymbol{M}_1^{\mathrm{INITIAL}}$ and $\boldsymbol{P}_1^{\mathrm{FINAL}}$ is given in Table 4 and the corresponding block-diagonal matrix is given in Fig. 6.

The grouping efficacy after step 1 is

$$\mu_1^1 = \frac{39-5}{39+12} = 66.67\%$$

Step 2 Determining a set of machine cells

Table 5 presents the value of the grouping efficacy, $\mu_F = \left\{\frac{N_1 - N_{1,F}^{\mathrm{Out}}}{N_1 + N_{0,F}^{\mathrm{In}}}\right\}$, for each product and each machine cell. A machine is assigned to the product family with the highest value of μ_F (the cells in bold in Table 5).

Table 4
Set of machine/product groups obtained after step 1.

Group	Machines	Products
1	M_3, M_8	3, 5, 7, 9
2	M_5, M_7, M_{12}	10, 14, 15
3	M_1, M_4, M_{10}, M_{11}	1, 4, 6, 12, 13
4	M_2, M_6, M_9	2, 8, 11

	Machine											
Product	M_8	M_3	M_5	M_7	M_{12}	M_1	M_4	M_{10}	M_{11}	M_2	M_6	M_9
3	1	1									1	
5	1	1									1	
7	1	1										
9	1	1									1	
10			1		1			1				
14			1	1	1							
15			1	1	1			1				
1						1	1					
4						1	1		1			
6						1	1		1			
12							1		1			
13						1			1			
2										1		1
8										1		1
11										1		1

Fig. 6. Block diagonal matrix corresponding to the product/machine cells in Table 4.

Thus,

$$\boldsymbol{M}_1^{\text{FINAL}} = \{(\boldsymbol{M}_3, \boldsymbol{M}_6, \boldsymbol{M}_8), (\boldsymbol{M}_5, \boldsymbol{M}_7, \boldsymbol{M}_{10}, \boldsymbol{M}_{12}), (\boldsymbol{M}_1, \boldsymbol{M}_4, \boldsymbol{M}_{11}), (\boldsymbol{M}_2, \boldsymbol{M}_9)\}.$$

The resulting grouping combining $\boldsymbol{P}_1^{\text{FINAL}}$ and $\boldsymbol{M}_1^{\text{FINAL}}$ is given in Table 6.

The corresponding block-diagonal matrix is given in Fig. 7.

The grouping efficacy after step 2 is

$$\mu_1^2 = \frac{39 - 0}{39 + 6} = 86.67\%.$$

The resulting block-diagonal matrix obtained at the end of step 2 has a grouping efficacy of $\mu_1 = \max(\mu_1^1, \mu_1^2) = \max(66.67\%, 86.67\%) = 86.67\%$. Since the set of machine cells obtained at the end of this step is different from the initial set of machine cells and has greater grouping efficacy we set

Table 5
Computations for step 2 of the local search heuristic

Machines	Machine products	Product families			
		(3, 5, 7, 9)	(10, 14, 15)	(1, 4, 6, 12, 13)	(2, 8, 11)
		μ_F	μ_F	μ_F	μ_F
$\boldsymbol{M}_1$	1, 4, 6, 13	(39−4)/(39+4)=81.4%	(39−4)/(39+3)=83.3%	**(39−0)/(39+1)=97.5%**	(39−4)/(39+3)=83.3%
$\boldsymbol{M}_2$	2, 8, 11	(39−3)/(39+4)=83.7%	(39−3)/(39+3)=85.7%	(39−3)/(39+5)=81.8%	**(39−0)/(39+0)=100.0%**
$\boldsymbol{M}_3$	3, 5, 7, 9	**(39−0)/(39+0)=100.0%**	(39−4)/(39+3)=83.3%	(39−4)/(39+5)=79.5%	(39−4)/(39+3)=83.3%
$\boldsymbol{M}_4$	1, 4, 6, 12	(39−4)/(39+4)=81.4%	(39−4)/(39+3)=83.3%	**(39−0)/(39+1)=97.5%**	(39−4)/(39+3)=83.3%
$\boldsymbol{M}_5$	10, 14, 15	(39−3)/(39+4)=83.7%	**(39−0)/(39+0)=100.0%**	(39−3)/(39+5)=81.8%	(39−3)/(39+3)=85.7%
$\boldsymbol{M}_6$	3, 5, 9	**(39−0)/(39+1)=97.5%**	(39−3)/(39+3)=85.7%	(39−3)/(39+5)=81.8%	(39−3)/(39+3)=85.7%
$\boldsymbol{M}_7$	14, 15	(39−2)/(39+3)=88.1%	**(39−0)/(39+1)=97.5%**	(39−2)/(39+5)=84.1%	(39−2)/(39+3)=88.1%
$\boldsymbol{M}_8$	3, 5, 7, 9	**(39−0)/(39+0)=100.0%**	(39−4)/(39+3)=83.3%	(39−4)/(39+5)=79.5%	(39−4)/(39+3)=83.3%
$\boldsymbol{M}_9$	2, 8, 11	(39−3)/(39+4)=83.7%	(39−3)/(39+3)=85.7%	(39−3)/(39+5)=81.8%	**(39−0)/(39+0)=100.0%**
$\boldsymbol{M}_{10}$	10, 15	(39−2)/(39+4)=86.0%	**(39−0)/(39+1)=97.5%**	(39−2)/(39+5)=84.1%	(39−2)/(39+3)=88.1%
$\boldsymbol{M}_{11}$	4, 6, 12, 13	(39−4)/(39+4)=81.4%	(39−4)/(39+3)=83.3%	**(39−0)/(39+1)=97.5%**	(39−4)/(39+3)=83.3%
$\boldsymbol{M}_{12}$	10, 14, 15	(39−3)/(39+4)=83.7%	**(39−0)/(39+0)=100.0%**	(39−3)/(39+5)=81.8%	(39−3)/(39+3)=85.7%

Table 6
Set of machine/product groups obtained after 2

Group	Machines	Products
1	$\boldsymbol{M}_3$, $\boldsymbol{M}_6$, $\boldsymbol{M}_8$	3, 5, 7, 9
2	$\boldsymbol{M}_5$, $\boldsymbol{M}_7$, $\boldsymbol{M}_{10}$, $\boldsymbol{M}_{12}$	10, 14, 15
3	$\boldsymbol{M}_1$, $\boldsymbol{M}_4$, $\boldsymbol{M}_{11}$	1, 4, 6, 12, 13
4	$\boldsymbol{M}_2$, $\boldsymbol{M}_9$	2, 8, 11

$\boldsymbol{M}_2^{\text{INITIAL}} = \boldsymbol{M}_1^{\text{FINAL}}$, i.e.

$$\boldsymbol{M}_2^{\text{INITIAL}} = \{(\boldsymbol{M}_3, \boldsymbol{M}_6, \boldsymbol{M}_8), (\boldsymbol{M}_5, \boldsymbol{M}_7, \boldsymbol{M}_{10}, \boldsymbol{M}_{12}), (\boldsymbol{M}_1, \boldsymbol{M}_4, \boldsymbol{M}_{11}), (\boldsymbol{M}_2, \boldsymbol{M}_9)\}$$

and proceed to iteration 2 to repeat steps 1 and 2. At the end of step 2 of the second iteration, we obtain a set of machine cells that is equal to the initial set (i.e. $\boldsymbol{M}_2^{\text{INITIAL}} = \boldsymbol{M}_2^{\text{FINAL}}$), and so we stop. The final block-diagonal matrix is the one shown in Fig. 7 and has a grouping efficacy of 86.67%.

5. Computational results

To demonstrate the performance of the proposed algorithm, we tested the hybrid genetic algorithm on 35 GT instances collected from the literature. The selected matrices range from dimension 5×7–40×100 and comprise well-structured, as well as unstructured matrices. The matrix sizes and their sources are presented in Table 7. The smallest dimension of each matrix was considered to be the number of rows.

We compare the grouping efficacy obtained by our algorithm with the grouping efficacies obtained by the following six approaches:

- ZODIAC (Chandrasekharan & Rajagopalan, 1987);
- GRAFICS (Srinivasan & Narendran, 1991);
- MST—Clustering algorithm (Srinivasan, 1994);
- GATSP—Genetic algorithm (Cheng et al., 1998);

	Machine											
Product	M_6	M_8	M_3	M_5	M_7	M_{10}	M_{12}	M_1	M_4	M_{11}	M_2	M_9
3	1	1	1									
5	1	1	1									
7		1	1									
9	1	1	1									
10				1		1	1					
14				1	1		1					
15				1	1	1	1					
1								1	1			
4								1	1	1		
6								1	1	1		
12									1	1		
13								1		1		
2											1	1
8											1	1
11											1	1

Fig. 7. Block diagonal matrix corresponding to the product/machine groups in Table 6.

Table 7
Experimental results

Problems			Other approaches						Our approach									
			Grouping efficacy						Grouping efficacy			N° Gen.			Improvement		Cpu Time (s)	
Prob. N°	Source	Size	ZODIAC	GRAFICS	MST	GATSP	GP	GA	Min	Avg	Max	Min	Avg	Max	Min (%)	Avg (%)	Max (%)	Avg
1	King and Nakornchai (1982)	5×7	73.68	73.68					73.68	73.68	73.68	1	1	1	0.00	0.00	0.00	0.53
2	Waghodekar and Sahu (1984)	5×7	56.52	60.87				62.50	62.50	62.50	62.50	1	2	3	0.00	0.00	0.00	0.47
3	Seifoddini (1989)	5×18	77.36			77.36		77.36	79.59	79.59	79.59	1	1	1	2.88	2.88	2.88	0.85
4	Kusiak (1992)	6×8	76.92			76.92		76.92	76.92	76.92	76.92	1	1	1	0.00	0.00	0.00	0.66
5	Kusiak and Chow (1987)	7×11	39.13	53.12		46.88		50.00	53.13	53.13	53.13	1	4	10	0.02	0.02	0.02	1.09
6	Boctor (1991)	7×11	70.37			70.37		70.37	70.37	70.37	70.37	1	1	2	0.00	0.00	0.00	1.35
7	Seifoddini and Wolfe (1986)	8×12	68.30	68.30					68.3	68.3	68.3	1	1	1	0.0	0.0	0.0	1.44
8	Chandrasekharan and Rajagopalan	8×20	85.24	85.24	85.24	85.24	85.24	85.25	85.25	85.25	85.25	1	1	1	0.00	0.00	0.00	1.68
9	Chandrasekharan and Rajagopalan (1989a,b)	8×20	58.33	58.13	58.72	58.33	58.72	55.91	58.72	58.72	58.72	1	4	15	0.00	0.00	0.00	1.68
10	Mosier and Taube (1985a)	10×10	70.59	70.59	70.59	70.59			70.59	70.59	70.59	1	1	2	0.00	0.00	0.00	1.82
11	Chan and Milner (1982)	10×15	92.00	92.00	92.00	92.00			92.00	92.00	92.00	1	1	1	0.00	0.00	0.00	2.19
12	Askin and Subramanian (1987)	14×24	64.36	64.36	64.36				69.86	69.86	69.86	1	9	16	8.55	8.55	8.55	6.05
13	Stanfel (1985)	14×24	65.55	65.55		67.44		63.48	69.33	69.33	69.33	1	3	12	2.80	2.80	2.80	6.24
14	McCormick et al. (1972)	16×24	32.09	45.52	48.70				52.58	52.58	52.58	1	21	68	7.97	7.97	7.97	7.85
15	Srinivasan et al. (1990)	16×30	67.83	67.83	67.83				67.83	67.83	67.83	1	1	2	0.00	0.00	0.00	10.24
16	King (1980)	16×43	53.76	54.39	54.44	53.89			54.86	54.86	54.86	1	1	2	0.77	0.77	0.77	13.92
17	Carrie (1973)	18×24	41.84	48.91	44.20				54.46	54.46	54.46	6	45	128	11.35	11.35	11.35	11.34
18	Mosier and Taube (1985b)	20×20	21.63	38.26		37.12		34.16	42.86	42.94	42.96	6	24	51	12.02	12.23	12.28	10.89
19	Kumar et al. (1986)	20×23	38.66	49.36	43.01	46.62	49.00	39.02	49.65	49.65	49.65	7	53	119	0.59	0.59	0.59	12.03
20	Carrie (1973)	20×35	75.14	75.14	75.14	75.28		66.30	76.22	76.22	76.22	4	45	107	1.25	1.25	1.25	15.61
21	Boe and Cheng (1991)	20×35	51.13			55.14		44.44	58.07	58.07	58.07	1	3	6	5.31	5.31	5.31	16.38
22	Chandrasekharan and Rajagopalan (1989a,b)	24×40	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	1	10	35	0.00	0.00	0.00	19.26
23	Chandrasekaran and Rajagopalan	24×40	85.11	85.11	85.11	85.11	85.11		85.11	85.11	85.11	1	1	3	0.00	0.00	0.00	25.28
24	Chandrasekharan and Rajagopalan (1989a,b)	24×40	73.51	73.51	73.51	73.03	73.51	73.03	73.51	73.51	73.51	1	1	1	0.00	0.00	0.00	26.82
25	Chandrasekharan and Rajagopalan (1989a,b)	24×40	20.42	43.27	51.81	49.37		37.62	51.85	51.88	51.97	28	92	132	0.08	0.14	0.31	26.48

(*continued on next page*)

Table 7 (*continued*)

Problems			Other approaches						Our approach									
			Grouping efficacy						Grouping efficacy			N° Gen.			Improvement		Cpu Time (s)	
Prob. N°	Source	Size	ZODIAC	GRAFICS	MST	GATSP	GP	GA	Min	Avg	Max	Min	Avg	Max	Min (%)	Avg (%)	Max (%)	Avg
26	Chandrasekharan and Rajagopalan (1989a,b)	24×40	18.23	44.51	44.72	44.67		34.76	45.78	46.69	47.06	33	86	147	2.37	4.41	5.23	25.97
27	Chandrasekharan and Rajagopalan (1989a,b)	24×40	17.61	41.67	44.17	42.50		34.06	44.51	44.75	44.87	24	81	128	0.77	1.31	1.58	26.06
28	McCormick et al. (1972)	27×27	52.14	41.37	51.00				54.27	54.27	54.27	3	6	23	4.09	4.09	4.09	25.90
29	Carrie (1973)	28×46	33.01	32.86	40.00				44.10	44.37	44.62	11	76	137	10.25	10.93	11.55	43.78
30	Kumar and Vannelli (1987)	30×41	33.46	55.43	55.29	53.80		40.96	57.30	58.11	58.48	13	105	146	3.37	4.83	5.50	43.00
31	Stanfel (1985)	30×50	46.06	56.32	58.70	56.61		48.28	58.82	59.21	59.66	12	80	145	0.20	0.87	1.64	52.45
32	Stanfel (1985)	30×50	21.11	47.96	46.30	45.93		37.55	50.25	50.48	50.51	8	32	81	4.77	5.25	5.32	48.97
33	King and Nakornchai (1982)	36×90	32.73	39.41	40.05				41.48	42.12	42.64	46	80	110	3.57	5.17	6.47	81.46
34	McCormick et al. (1972)	37×53	52.21	52.21					56.42	56.42	56.42	1	2	4	8.06	8.06	8.06	87.66
35	Chandrasekharan and Rajagopalan (1989a,b)	40×100	83.66	83.92	83.92	84.03	84.03	83.90	84.03	84.03	84.03	1	5	12	0.00	0.00	0.00	152.13

Cpu Time (s), CPU time in seconds for 150 generations; **Improvement,** improvement of our algorithm against the best of the other approaches; **N° Gen.,** Generation number where best grouping efficacy was obtained.

- GA—Genetic algorithm (Onwubolu & Mutingi, 2001);
- GP—Genetic programming (Dimopoulos & Mort, 2001).

These six approaches provide the best results, found in the literature, for the 35 problems used for comparison.

The grouping efficacy resulting from the application of ZODIAC to above problems is reported in Srinivasan and Narendran (1991), in Srinivasan (1994), and in Cheng et al. (1998). The grouping efficacy resulting from the application of GRAPHICS, MST, GATSP, GA and GP is the one reported by their authors in their papers. The paper by Srinivasan and Narendran (1991) includes five problems not included in Table 7. Two of these problems were from an unpublished master's thesis, and were unavailable. One of the instances was from Seifoddini and Wolfe (1986), but the referenced source does not have any problem of the same dimension 12×12 and with the same number of 1's. The remaining two other problems, with matrices of sizes 12×10 and 10×20 were from McAuley (1972) and Badarinarayna (1987). However, even after much effort to obtain these instances, we were unable to find them. The paper by Dimopoulos and Mort (2001) includes 11 problems not included because none of the other approaches uses them.

ZODIAC, GRAPHICS and MST do not allow singletons (cells having less than two products or two machines) and this constraint degrades the performance of the algorithms in comparison with others methodologies. In order to make the comparisons fair and meaningful we have not included in the comparison the problems values presented by Cheng et al. (1998), Dimopoulos and Mort (2001), and Onwubolu and Mutingi (2001) whose solution allows the existence of singletons clusters. Additionally, we have not included the values obtained by Dimopoulos and Mort (2001) for problems 10 and 16 because the value reported for problem 10 is not possible and the value reported for problem 16 is not consistent with the value reported for corresponding the grouping efficiency of the problem.

The test was run on a personal computer having a MS Windows Me PC with an AMD Thunderbird 1.333 GHz processor. The algorithm was coded in Visual Objects 2.0b-1 from CA-Computer Associates.

The present state-of-the-art practice on genetic algorithms does not provide information on how to configure them. Therefore, a small pilot study was conducted in order to obtain a reasonable configuration. The algorithm was configured as follows and the configuration was held constant for all problems. The number of chromosomes in the population equals three times the number of rows in the problem. The probability of tossing heads during crossover was made equal to 0.7. The elitist strategy copies to the next generation the top (the best) 20% of the previous population chromosomes. Mutation substitutes with randomly generated chromosomes the bottom (the worst) 30% of the population chromosomes. The genetic algorithm stops after 150 generations. The algorithm was replicated 10 times using different initial seeds for the pseudo-random number generator incorporated.

The local search procedure presented in Section 4.2 can produce singletons (cells having less than two products or two machines). We address these cases by penalizing their grouping efficacy, i.e. we consider them to have a grouping efficacy of zero. By doing this we make sure that the evolutionary process of the genetic algorithm will remove the corresponding chromosomes from the population since the chromosomes with the lowest quality are not copied into the next generation.

The test results are presented in Table 7. In Appendix A, we present the block-diagonal matrices, found in the first run of the proposed algorithm, for each of the 35 problems mentioned in Table 7.

As can be seen in Table 7, the algorithm proposed in this paper obtained machine/product groupings, which have a grouping efficacy that is never smaller than any of the best reported results. More specifically, the algorithm obtains for 14 (40%) problems values of the grouping efficacy that are equal to the best ones

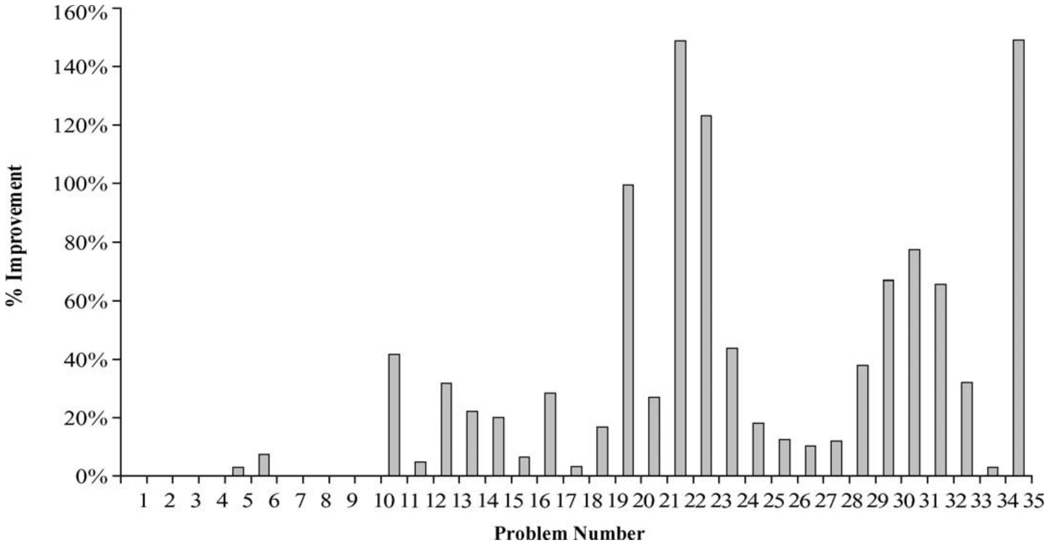


Fig. 8. % Improvement of the Local Search Heuristic w.r.t. the Customary Allocation Rule.

found in the literature and improves the values of the grouping efficacy for 21 (60%) problems. In 11 (31%) problems, the percentage improvement is higher than 5%. For 12 (34%) problems, the solution was obtained in the first generation, showing the good quality and power of the local search heuristic.

To further evaluate the performance of the local search heuristic, another test was run. In this test, the proposed algorithm was run with the local search heuristic replaced by the customary allocation rule, i.e. products are allocated to the cell where it visits the maximum number of machines (since the machine cells are known). Fig. 8 shows a graph with the percentage improvement of the grouping efficacy obtained by local heuristic over the customary allocation rule.

6. Conclusion

A new approach for obtaining machine cells and product families has been presented. The approach combines a local search heuristic with a genetic algorithm. The genetic algorithm uses a random keys alphabet, an elitist selection strategy, and a parameterized uniform crossover. Computational experience with the algorithm, on a set of 35 GT problems from the literature, has shown that it performs remarkably well. The algorithm obtained solutions that are at least as good as the ones found the literature. For 57% of the problems, the algorithm improved the previous solutions, in some cases by as much as 12%.

Acknowledgements

The authors would like thank (in alphabetical order): A. S. Carrie, C. H. Cheng, Y. Gupta, B. Mahadevan, G. Srinivasan, and L. Stanfel, for providing and helping to collect the problem data sets.

Appendix A

```
    235 14
   --------
 1 |XXX|  |
 3 |XX |  |
 7 | XX|  |
   --------
 2 |   |XX|
 4 |   |XX|
 5 |  X|X |
 6 | X |XX|
   --------
```

King and Nakornchai (1982) – 5 x 7

```
    14 235
   --------
 1 |XX|   |
 7 |X |   |
   --------
 2 | X|X X|
 3 | X|XX |
 4 | X|XXX|
 5 |X |XXX|
 6 |X | XX|
   --------
```

Waghodekar and Sahu (1984) - 5 x 7

```
    235 14
   --------
 4 |XXX|  |
 7 |XX |  |
 9 |  X|  |
10 |XXX|  |
15 |XXX|  |
18 |XXX|  |
   --------
 1 |X  |XX|
 2 |   |XX|
 3 |X  |XX|
 5 |   |XX|
 6 |X  |XX|
 8 |X  |XX|
11 |X  |XX|
12 |X  |XX|
13 |X  |XX|
14 |   |XX|
16 |   |XX|
17 |   |XX|
   --------
```

Seiffodini (1989) – 5x18

```
    146 235
   ---------
 4 |XXX|   |
 7 |XXX|X  |
   ---------
 1 |   |X X|
 2 |X  |X  |
 3 |   |XXX|
 5 |   |X X|
 6 |   |XXX|
 8 |   |XXX|
   ---------
```

Kusiak (1992) – 6x81

```
    23 56 147
   -----------
 1 |X | X| X |
 5 |X |X |   |
10 | X| X|   |
11 |XX|  |   |
   -----------
 4 |  | X|  X|
 8 |  |XX|   |
 9 |  | X|  X|
   -----------
 2 |  |  |X  |
 3 |  |  |XXX|
 6 |  |  | XX|
 7 |  |  |X X|
   -----------
```

Kusiak and Chow (1987) – 7x11

```
    67 12 345
   -----------
 4 |X |  |  X|
 5 |XX|  |   |
 8 | X|  |   |
10 |XX|  |   |
   -----------
 1 |  |X |X  |
 2 |  |XX|   |
 6 |  |XX|   |
 9 |  | X|   |
   -----------
 3 |  |  |XXX|
 7 |  |  |XX |
11 |  |  |X X|
   -----------
```

Boctor (1991) – 7x11

```
    78 3456 12
   ------------
11 |XX|   X|  |
12 |XX|    |  |
   ------------
 6 |  |XX  | X|
 7 |  |XXXX| X|
 8 |  |XXXX|  |
 9 |  |XXXX|  |
10 |  | XX | X|
   ------------
 1 |  |    |XX|
 2 |  |    |X |
 3 |  |X   |XX|
 4 |  |X   |XX|
 5 |  |X   | X|
   ------------
```

Seiffodini and Wolfe (1986) – 8x12

```
    56 2478 13
   ------------
 1 |XX|    |  |
 5 |XX|    |  |
10 |XX| X  |  |
12 |XX|  X |  |
15 |XX|    |  |
   ------------
 3 |  |XXXX|X |
 4 |  |XXXX|  |
 6 |X |XXXX|  |
 7 |  |XXXX|  |
18 |  |XXXX|  |
20 | X|XXXX|  |
   ------------
 2 |  |    |XX|
 8 |  |    |XX|
 9 | X|    |XX|
11 |  |  X |XX|
13 |  |    |XX|
14 |  |X   |XX|
16 |  |    |XX|
17 |X |    |XX|
19 |  |    |XX|
   ------------
```

Chandrasekaran and Rajagopalan (1986a) - 8x20

```
    357 12468
   -----------
 5 |X X|   XX|
 6 |XXX| X   |
11 |X X| X X |
12 |XXX|   X |
13 |XXX|  X X|
16 |XXX|X  XX|
17 |XXX|  X X|
19 |XXX|X XX |
20 |XXX| XX  |
   -----------
 1 | X |X  XX|
 2 | X | X XX|
 3 | X |XXX X|
 4 |   |XXX X|
 7 |X X| XXX |
 8 |X X|  XXX|
 9 |   |XXX X|
10 | X |X XXX|
14 |   |X XXX|
15 | X |X XX |
18 |   |XXXX |
   -----------
```

Chandrasekaran and Rajagopalan (1986b) - 8x20

```
                1
    1456 38 2790
   --------------
 1 |XX X|  |    |
 7 |  XX|  |    |
10 |X XX|  |    |
   --------------
 5 |    |X |    |
 6 |    |XX|    |
 9 |    | X|    |
   --------------
 2 |    |  |  XX|
 3 |    |  |XXXX|
 4 |    |  |X XX|
 8 |    |  |XXX |
   --------------
```

Mosier and Taube (1985a) – 10x10

```
               1
    258 3469 170
   --------------
 3 |XXX|    |   |
 5 |XXX|    |   |
 8 |XXX|    |   |
13 |XXX|    |   |
15 |XXX|    |   |
   --------------
 1 |   |XXX |   |
 4 |   | XXX|   |
 6 |   |X XX|   |
 9 |   |XXXX|   |
14 |   |XXXX|   |
   --------------
 2 |   |    |XXX|
 7 |   |    | XX|
10 |   |    |XXX|
11 |   |    |XXX|
12 |   |    |XXX|
   --------------
```

Chan and Milner (1982) – 10x15

```
     11 11       1
    123 04 457 231 689
   --------------------
 7 |X X|  |   |   |   |
 8 |XXX|  |  X|   |   |
 9 | XX|  |   |   |   |
18 |  X|  |   |   |   |
   --------------------
11 |   | X|   |   |X  |
13 |   | X|   |   |  X|
23 |   |X |   |   |   |
   --------------------
 2 |   |  |XXX|   |   |
 3 |   |  |XXX|   |   |
17 |   |  |XXX|   |   |
19 |   |  |X  |   |   |
20 |   |  |XXX|   |   |
22 |  X|  |XX |   |   |
   --------------------
 4 |   |X |   |XXX|   |
 5 |   |  |   |XXX|   |
21 |   |  |   | XX|   |
   --------------------
 1 |   |  |   |   |XX |
 6 |   |  |   |   | XX|
10 |   | X|   |   |XXX|
12 |   |  |   |   |XXX|
14 |   |  |   |   |XX |
15 |   | X|   |   |XXX|
16 |   |  |   |   |XX |
   --------------------
```

Askin and Subramanian (1987) - 14x24

```
         1   11     11
    457 94 2301 68 123
   --------------------
 1 |XXX|  |    |  |   |
 2 |XXX|  |    |  |   |
17 |XXX|  |    |  |   |
19 |X  |  |    |  |   |
20 |XXX|  |    |  |   |
23 |XX |  |    |  |  X|
   --------------------
 5 |   |X |    | X|   |
 9 |   |XX|    |XX|   |
11 |   | X|    |X |   |
13 |   |XX|    |  |   |
15 |   |XX|    |XX|   |
   --------------------
 3 |   |  |XXXX|  |   |
 4 |   |  |XX X|  |   |
21 |   |  | X X|  |   |
24 |   |  |  XX|  |   |
   --------------------
10 |   |  |    |XX|   |
12 |   |X |    |XX|   |
14 |   |  |    |XX|   |
16 |   |  |    |XX|   |
22 |   |  |    |XX|   |
   --------------------
 6 |   |  |    |  |X X|
 7 |  X|  |    |  |XXX|
 8 |   |  |    |  | XX|
18 |   |  |    |  |  X|
   --------------------
```

Stanfel (85) – 14x24

```
    11  1 1111
    45 93 0126 56 128 347
   -----------------------
 4 |X |  |    |  |X  |X  |
 5 | X|  |    |  |   |   |
 6 |XX|  |    |  |X  |X  |
12 |X |  |    |  |   |   |
   -----------------------
 7 |  |XX| X  |  |X X|   |
 8 |  | X|    |  |X  |   |
   -----------------------
 2 |  | X|X X | X|   |   |
 9 | X|  |XXXX|  |   |   |
11 |  | X|  XX|  |X  |   |
14 |X |  |XXXX|  |   |   |
17 |  |  |X  X| X| X |X  |
20 | X|  |XXXX|  |   |   |
24 |X |X |XXXX|  |   |  X|
   -----------------------
13 |  |  |    | X|  X|   |
15 |  |  |    | X|X  |   |
18 |  |  |    | X|X  |   |
21 |  |  |    |XX|   |   |
   -----------------------
 1 |X |  |    |  |X X|X X|
 3 |  |  |  X | X|X X|   |
10 |  |  |X   |  |XXX|   |
22 |  |  |X   |  |XXX|   |
23 |  |  |X   |  |XXX|   |
   -----------------------
16 |  |  |    |  |   |XX |
19 |  |  |    |  |X  |XXX|
   -----------------------
```

McCormick et al. (1972) - 24x16

```
     111    1     11  1
    5046 3695 147812 23
   ---------------------
 6 |XXXX|    |      |  |
 8 |XXXX|    |      |  |
11 |XXX |    |    X |  |
14 |XXXX|    |      |  |
15 |XXXX|    |      |  |
17 |  X |    |      |  |
21 |XXX |    |      |  |
24 |XX X|    |      |  |
26 |XX X|   X|      |  |
   ---------------------
 5 |    |  XX|X     |  |
19 |    |  X |     X|  |
23 |    |XXXX|      |  |
25 |    |XXXX|  X   |  |
27 |  XX|XXXX|      |  |
28 |    |X X |   X  |  |
29 |    |XX X|      | X|
   ---------------------
 2 |    |    | XXXXX|  |
 4 |    |    |XXXX X|X |
 7 |   X|    |XXXXXX|  |
 9 |    |    |XX XX |  |
12 |    |    |X XXXX|  |
18 | X  |    |XXXXXX|X |
22 |    |    |XXXX X|  |
30 |    |   X|XXX X | X|
   ---------------------
 1 |    |X   |      |X |
 3 |X   |    |      | X|
10 |    |    |      |XX|
13 |    |X   |      | X|
16 |    |    |      |X |
20 |    |    |     X|XX|
   ---------------------
```

Srinivasan et al. (1990) -16x30

```
       1    1   1 111  1
    1296 4585 364 123 70
   ----------------------
 2 | XXX|  X | XX|   |  |
 4 |  X |    |   |   |  |
10 | XXX|    |   |   |  |
18 |  XX|    |   |   |  |
28 | XX |  X |   |   |  |
32 | XXX|    | X |   |  |
37 |XXXX|  X | X |   |  |
38 | XXX|  X |   |   |  |
40 | XX |    | X |   |  |
42 |XXXX|    | X |   |  |
   ----------------------
 5 |    |XX X|   |   |  |
 8 |    | XX | X |   |  |
 9 |    |XXX |   |X  |  |
14 |    |XX X| X |   |  |
15 |    | XX |   |   |  |
16 |    | X  |   |   |  |
19 |    |XXXX| X |   |  |
21 |    |XXXX|   |   |  |
23 |    |XXX | X |   |  |
29 |    |XX  |   |   |  |
33 |    | X X| X |   |  |
41 |    | XXX|   |   |  |
43 |    | XXX| X |   |  |
   ----------------------
 6 |    |    | XX|   |  |
 7 |   X|    |XX |   |  |
17 |    |    |XXX|   |  |
34 |    |    |XX |   |  |
35 |    |    |X X|   |  |
36 |    |    |X  |   |  |
   ----------------------
 3 |    |  X |   |X X|  |
11 |    |  X |   | X |  |
20 |    |  X |   |X  |  |
22 |    |    |   | X |  |
24 |    |  X |   |XXX|  |
27 |    |  X |   |XX |  |
30 |    |    |   |XX |  |
   ----------------------
 1 |    |  X | X |   |XX|
12 |    |  X | X |   | X|
13 |    |    | X |   |XX|
25 |    |    |   |   |XX|
26 |    |    |   |   | X|
31 |    |  X |   |   | X|
39 |    |    | X |   | X|
   ----------------------
```

King (1980) - 16x43

```
       1111          111 11
    12 1245 89 34567 678 03
   -------------------------
10 |X |    |X |  XX |   |  |
23 |XX|    |X |  XX |   |  |
   -------------------------
 7 |  |XX X|  |     |   |  |
13 |X | X X|  |     |   |  |
14 |  |XX  |  |     |   | X|
18 |  |XXXX|  |     |   |  |
21 |  |XXX |  |     |   |  |
   -------------------------
 1 |  |   X|X |     |   |  |
 3 |  |    |XX|   X |   |  |
20 |  |  X |X |     |   |  |
24 |  |    |XX|  XX |   |  |
   -------------------------
 2 |  |    |  |XXX X|   |X |
 5 |  |    |  |XXXXX|   |  |
 6 |  | X  |  | X  X|   |  |
 8 |  |    |  |XXXX |   | X|
 9 |  |    |  |XXXX |   | X|
12 |  |    |  |XXX X|   |X |
15 |  |   X|  |XXXX |   |  |
17 |  |   X|  |XXXX |   |  |
   -------------------------
 4 |  |    |X |  X  |XX |  |
11 |  |    |  |  X  |  X|  |
16 |  |    |  |  X  |  X|  |
   -------------------------
19 |  | X  |  |     |   |XX|
22 |  | X  |  | X   |   | X|
   -------------------------
```

Carrie (1973)- 18x24

```
    11   1112  1      1  111
    23 235670 78 146899 5014
   --------------------------
 2 | X|X     |  |      |    |
11 |XX| X X  |  |X     | X  |
12 |X |      |  |      |    |
17 |XX|X     |  |      |   X|
18 |XX| X    |X |   X  |  XX|
   --------------------------
 3 |  |XXX  X|  |      |X   |
 4 |X | X  XX|  |   X  |  X |
 5 |  |X XXX |  |      |  X |
 8 |  |XXX  X| X|      |    |
16 | X| XXXX |XX|X   X |    |
   --------------------------
 1 |  |     X|XX|X     |    |
 7 |  |X   X |XX|  X   |    |
20 |  |X     |X |   X  |    |
   --------------------------
 6 |  |      |X |XXXXX |    |
 9 |  |  X   |X |XX XX |    |
10 | X|X     |  | X X X| X  |
14 |X | X XX |X |  XXXX|  XX|
   --------------------------
13 | X|XX    |  |X     |XXXX|
15 |  |X   X |  |   X  |XXX |
19 |  |     X|X |    X | XXX|
   --------------------------
```

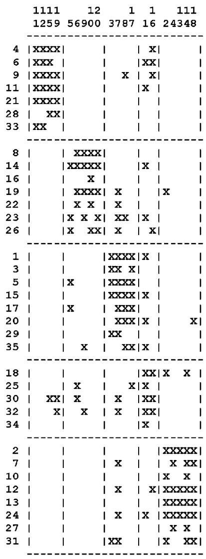
Mosier and Taube (85b) – 20x20

```
        111  11  111  1   12
    1356238 957 2169 70 4840
   --------------------------
 1 |XXXXXXX| X |   X|  |XXX |
 2 | X XX  |X  |    | X|    |
 4 |  X  X |   |    |  |    |
10 |X XXXXX|X  |    |  |    |
11 |XXX X  |   |    |  |    |
15 |XXXXXXX|  X|X   |  | XXX|
20 |    XX |   |  X |  |    |
   --------------------------
13 |       |  X|X   |  | X  |
18 | X  XX |XXX|  X |  |    |
19 |       |XXX|    |  |    |
21 |X      |XX |    |  |X   |
23 |X      |XX |    |  |X   |
   --------------------------
 5 |       |X  |XXXX|  |    |
14 |       |   |X X |  |    |
   --------------------------
 6 |       |  X|    | X|    |
 7 | X     |X X| XX |XX|    |
 8 |       |   |    |XX|    |
 9 |       |   |    |XX|    |
17 |       |   |    | X|X   |
   --------------------------
 3 |X     X|   |   X|  |XX X|
12 |       | X |    |  |XXXX|
16 |       |   |    | X| XXX|
22 |XX X   |XX |    | X|XXXX|
   --------------------------
```

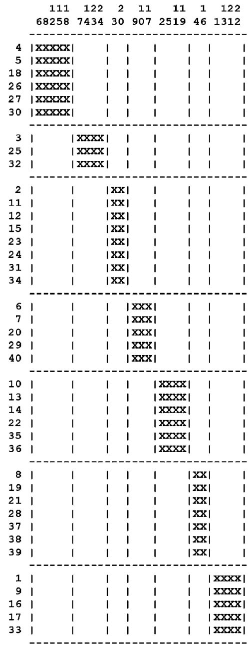
Kumar et al. (1986) - 20x23

```
    11111   111    12     1
    12569 24348 56900 13787
   -------------------------
 4 |XXXXX|     |     |     |
 6 |XXXX |     |     |     |
 9 |XXXXX|     |     |     |
11 |XXXXX|     |     |     |
21 |XXXXX|     |     |     |
28 |X XXX|     |     |     |
30 |X XXX|     |     |     |
32 |X  XX|     |     |     |
33 | X   |     |     |     |
35 |X    |     |     |     |
   -------------------------
 2 |     |XXXXX|     |     |
 7 |     |XX X |     |     |
10 |     |X  XX|     |     |
12 |     |XXXXX|     |     |
13 |     |XXXXX|     |     |
18 |     |X  XX|     |     |
24 |     |XXXXX|     |     |
27 |     |XX X |     |     |
31 |     |X  XX|     | X   |
   -------------------------
 8 |     |     |XXXXX|     |
14 |     |     |XXXXX|     |
16 |     |     |XX X |     |
19 |     |     | XXXX|     |
22 |     |     | XXX |     |
26 |     |     | XXXX|     |
34 |     |     |XX   |     |
   -------------------------
 1 |     |     |     |XXXXX|
 3 |     |     |     |XXXXX|
 5 |     |     |     | XXXX|
15 |     |     |     | XXXX|
17 |     |     |     | XXXX|
20 |     |     |     |X XX |
23 |     |     |     |X XXX|
25 |     |     |     |X  XX|
29 |     |     |     |XX XX|
   -------------------------
```

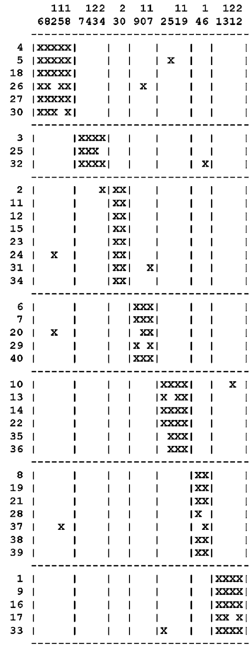
Carrie (1973) - 20x35



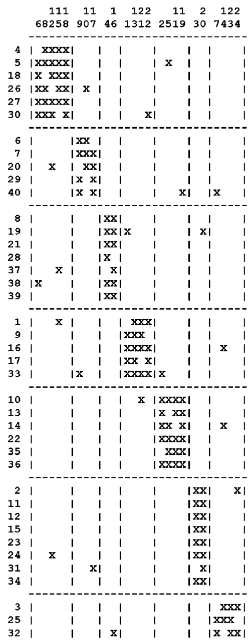
Boe and Cheng (1991) – 20X35



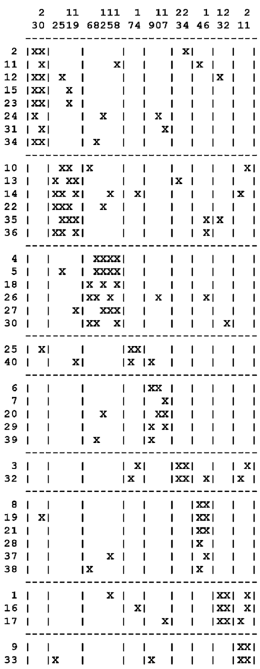
Chandrasekaran and Rajagopalan (1989)
Matrix1 – 24X40



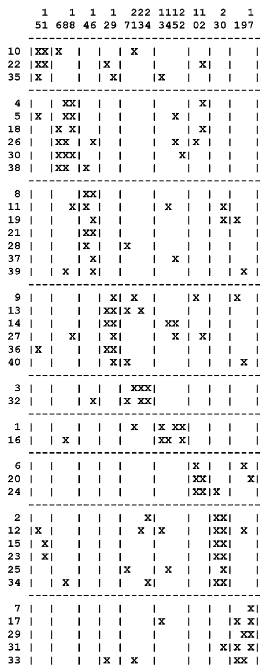
Chandrasekaran and Rajagopalan (1989)
Matrix2 – 24X40



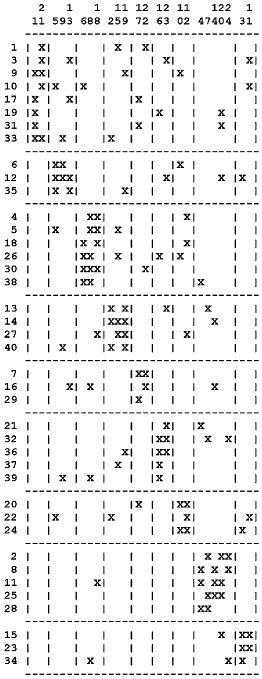
Chandrasekaran and Rajagopalan (1989)
Matrix3 or Matrix4 – 24X40



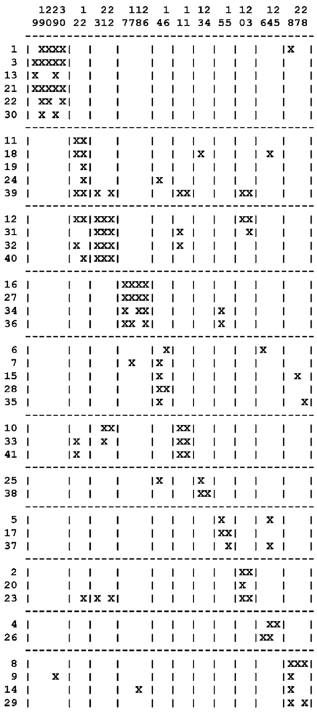
Chandrasekaran and Rajagopalan (1989)
Matrix5 – 24X40



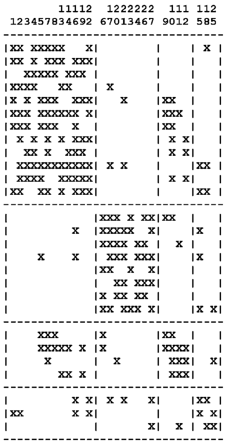
Chandrasekaran and Rajagopalan (1989)
Matrix6 – 24X40



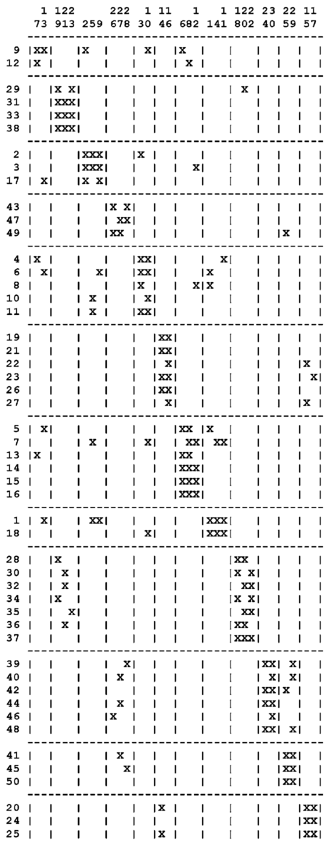
Chandrasekaran and Rajagopalan (1989) Matrix7 – 24 X 40



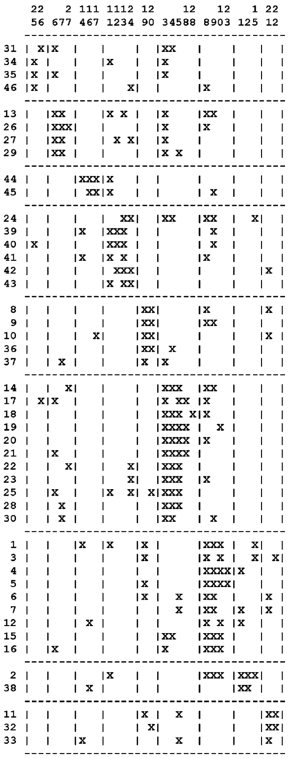
Kumar and Vannelli (1987) - 30 X 41



McCormick et al. (1972) - 27 X 27



Stanfel (1985), (fig. 5) – 30 X 50



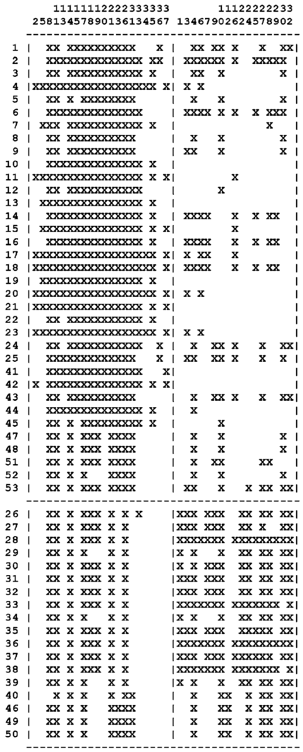
Carrie (1973) – 28 X 46

```
     1   11 12 12  11      2 12223 222 12  1
    191 779 80 58 224 46 385 61360 247 09 53
    ------------------------------------------
 8 |XXX|   |  |  |   |  |   |     |   |  | X|
 9 |X X|   |  |  |   |  |   |     |   |  | X|
21 |XX |  X|  |  |   |  |   |X X  |   |  |  |
36 | XX|   |  |  |   |  |  X|     |   |  |  |
41 | XX|X  |  |  |   |  |   |     | X |  |  |
49 | X |   |  |  |   |  |X  |     |   |  |  |
    ------------------------------------------
28 |   |XXX|  |  |   |  |   |     |   |  |  |
30 |   |XXX|  |  |   |  |   |     |   |  |  |
50 |   | XX|  |X |   |  |   |     |   |  |  |
    ------------------------------------------
 5 |   | X |XX|  |   |  |   | X   |   |  |  |
 6 |   |   |XX|  |   |  |   |  X  |   |  |  |
 7 |   | X | X|  |   |  |   | X   |   |  |  |
12 |   | X | X|X |   |  |   |     |   |  |  |
26 |   |  X|XX|  |   |  |   |     |   |  |  |
27 |   |   |XX|  |   |  |   |     |   |  |  |
29 |   |X  | X|  |   |  | X |     |   |  |  |
    ------------------------------------------
10 |   |   |  |XX|   |  |   |   X |   |  |  |
14 |   |   |  | X|   |  |   | X   |   |  |  |
18 |   |   |  |XX|   |  |   |     |   |  |  |
40 |   |   |  |XX|   |  |  X|     |   |  | X|
46 |   |  X|  |X |X  |  |   |  X  |   |X |  |
    ------------------------------------------
19 |   |X  |  |  |XXX|  |   |     |   |  |  |
20 |   |   |  |  |XXX|  |   |   X |   |  |  |
38 |   |   |  |  |X  |  |   |     |   |  |  |
44 |   | X |  |  | XX|  |   |     |   |  | X|
    ------------------------------------------
17 |   |   |  |  |X  |XX|   |   X |   | X|  |
22 | X |   |  |  |   |XX|   |     |   |  |  |
32 |   |   |  |  |  X|X |   |X    |   |  |  |
43 |X  |  X|  |  |   |X |   |     |  X|  |  |
48 |   |   |  |  |  X| X|   |     |   |  |  |
    ------------------------------------------
 1 |   |  X|  |  |   |  |XXX|     |   |  |  |
 4 |   |   |  |  |   |  |XXX|     |   |  |  |
16 |   |   |  |  |   |  |X X|    X|   |  |  |
33 |   |   |  |  |   |X | XX|     |   |  |  |
42 |   |   |  |  |   |  | X |     |   |  |  |
45 |   |   |  | X|   |  | XX| X   |   |  |  |
    ------------------------------------------
31 |   |   |  |  |  X|  |   |X XX |   |  |  |
35 |   |   |  |  |   |  |   |XX XX|   |  |  |
39 |   |   |  |  |   | X|X  |XXX X|   |X |  |
    ------------------------------------------
13 |   |   |  |  |   |  |   |   X |XXX|  |  |
15 |   |   |  |  |   |  |   |     |XXX|  |  |
34 |X  |   |  |  |   |X |X  |  XX | XX|  |  |
    ------------------------------------------
23 |  X|   |  |  | X |  |   |     |   |XX|  |
24 |   |   |  |  | X |  |   |     |   |XX|  |
25 |  X|   |  |  | X |  |   |     |   | X|  |
37 |   | X |  |  |   |  |   |     |   |X |  |
47 |   |  X|  |  |   |  |   |     |   |X |  |
    ------------------------------------------
 2 |   |   |  |  |   | X|   |     |   |X |XX|
 3 |   |   |  |  |   |  |   |     |   |X |XX|
11 |   |   |  |  |   |  |   |     |   |  |XX|
    ------------------------------------------
```

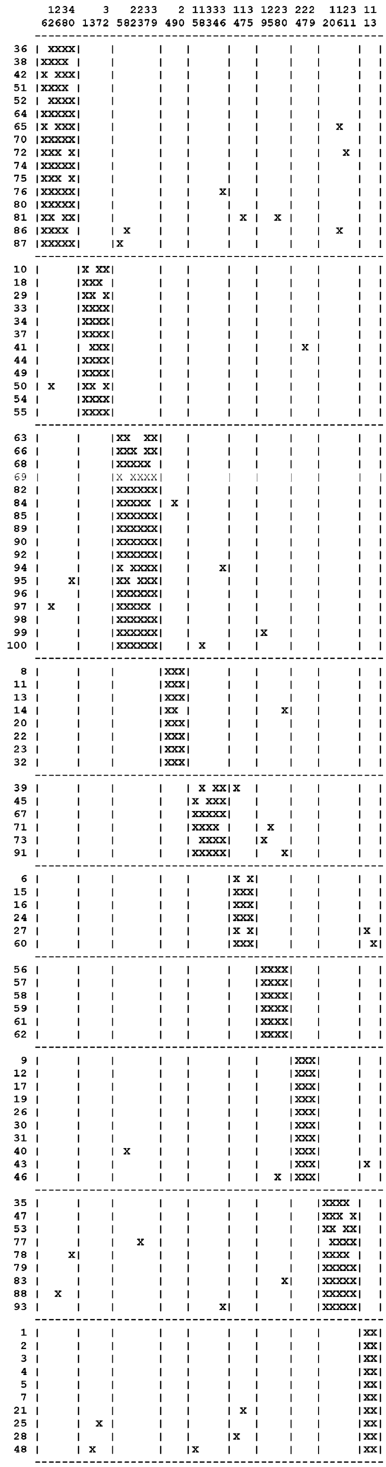
Stanfel (1985), (fig. 6) – 30 x 50

```
     1   2  11 222  11   22 12  11122  123
    42 260 346 368 108 7857 39 917914 5520
   ----------------------------------------
 4 |X |   |   | X |   |    |  |      |    |
23 |X |   |   |   |   |    |  |      |    |
38 |X |   |   |   |   |    |  |      |    |
48 |X |  X|   |   |   |    |  |      |   X|
52 |X |   |   |   |   |    |  |    X |   X|
74 |X |   |   |  X|   |    |  |      |  X |
86 |XX|   |  X|  X|   |    |  |    X | X  |
   ----------------------------------------
17 |  |  X|   |   |   |    |  |      |    |
35 |  | XX|   |   |   |    |  |      |    |
90 |  |X X| X | X |   |    |  |  XXXX|    |
   ----------------------------------------
24 |  |   |X  |   |   |    |  |      |    |
37 |  |   |X  | X |   |    |  |      |    |
44 |  |   |X  |   |   |    |  |      |    |
58 |  |   |X  | X |   |    |  |      |    |
59 |  |   |XXX|   |   |    |  |      |    |
65 |  |   |XXX|  X|   |    |  |   X X|    |
67 |  |   |XX |   |   |    |  |  X   |   X|
70 |  |   | X |  X|   |    |  |      |   X|
71 |  |   |XX |  X|   |    |  |      |   X|
83 |X |   |X X|   |   |  X |  |      | XX |
87 |  |  X|XX |   |   |    |  |  X X |    |
89 |  |   |XXX|   |   |    |  |   XXX|    |
   ----------------------------------------
 1 |  |   |   |  X|   |    |  |      |    |
 2 |  |  X|   | X |   |    |  |      |    |
 3 |  |   |   | X |   |    |  |      |    |
 5 |  |   |   | X |   |   X|  |      |    |
 6 |  |   |   | X |   |    |  |      |    |
 7 |  |   |   | X |   |    |  |      |    |
 9 |  |   |X  | XX|   |    |  |   X  |  XX|
13 |  |   |   | XX|   |   X|  |      |    |
16 |  |   |   |  X|X  |    |  |      |    |
19 |  |   |   | X |   |  X |  |      |    |
25 |  |   |   | XX|   |    |  |      |    |
27 |  |   |   |XXX|   |    | X|      |    |
29 |X |   |   | XX|   |    |  |      |    |
31 |  |   |   | XX|   |   X|  |      |    |
33 |  |   |   |  X|   |    |  |      |  X |
42 |  |  X|   | X |   |    |  |      |    |
43 |X |   |   | XX|   |    |  |      |    |
46 |  |   |   | X |   |   X|  |      |    |
50 |  |   |   | X |X  |    |  |      |    |
54 |  |   |   | XX|   |   X|  |      |    |
60 |  |   |   |XXX|   |    |  |      |    |
69 |X |   |   | XX|   |    |  |X   X |    |
72 |  |   |   | XX|   |   X|  |      |    |
75 |  |   |   | X |   | X  |  |      |    |
78 |  |  X|   | X |   |    |  |    X | X  |
79 | X|  X|   | XX|   |    |  |  X XX|   X|
84 |  | XX|  X| XX|   |   X|  |      |    |
   ----------------------------------------
18 |  |   |   |   | X |    |  |      |    |
26 |  |   |   |   |  X|    |  |      |    |
34 |  |   |   |   |X  |    |  |      |    |
   ----------------------------------------
10 |  |   |   |   |   |  X |  |      |    |
11 |  |   |   | XX|   |X XX|  |      |    |
12 |  |   |   |   |   |   X|  |      |    |
28 |  |   |   |   |   |X XX|  |      |    |
36 |  |   |   | X |   |XXXX|  |      |    |
39 |  |   |   |   |   |X   |  |      |    |
41 |  |   |   | XX|X  |X XX|  |      |    |
49 |X |   |   | X |   |  XX|  |      |    |
61 |  |   |   |   |   |  XX|  |      |    |
   ----------------------------------------
14 |  |   |X  |   |   |    | X|      |    |
32 |  |   |   |   |   |    | X|      |    |
47 |  |   |X  |   |   |    | X|      |    |
53 |  |   |   |   |   |    | X|      |    |
55 |  |  X| X | X |   |  X | X|      |    |
57 |  |   |   |  X|   |    | X|      |    |
66 |  |   |X  | X |   |    | X|      |    |
80 |  |   |   |   |   |    |XX|     X|    |
82 |  |   |   |   |  200||    |XX|     X|   X|
   ----------------------------------------
 8 |  |   |   |   |   |    |  | X XXX|  XX|
15 |  |   |   |   |   |    |  | XXX X|  X |
21 |  |   |   |   |   |    |  | XXXXX|  XX|
22 |  |   |   |   |   |    |  | XXXXX|  XX|
40 |  |   |X  |   |   |    |  |XXXXXX|    |
51 |  |   |   |   |   |    |  | XXX  |    |
56 |  |   |X  |   |   |    |  |XXX X |    |
62 |  |   |  X|   |   |    |  | X X  |    |
63 |  |  X|   | X |   |    |  | XX   |    |
64 |  |   |X  |   |   |    |  |XXX X |    |
68 |  |   |   |   |   |    |  | XXX  |    |
73 |  |   |   |   | X |    |  | XXXX |    |
76 |  |   |   |   |   |    |  |XXXXXX|    |
77 |  |  X| X |   |X  |    |  |X XX X|    |
88 |  |  X| X |   |   |    |  |  XXX | X  |
   ----------------------------------------
20 |  |   |   |   |   |    |  |      |  X |
30 |  |   |   |   |   |    |  | X    |  X |
45 |  |   |   |   |   |    |  |      |   X|
81 |  |   |X  |   |   |  X |  | X    |XXXX|
85 |  |   |X  |   |   |  X |  | X    |XXXX|
   ----------------------------------------
```

King and Nakornchai (1982) - 36 x 90



McCormick et al. (1982) – 37 x 53



Chandrasekaran and Rajagopalan (1987) – 40 x 100

References

Adil, G. K., Rajamani, D., & Strong, D. (1997). Assignment allocation and simulated annealing algorithms for cell formation. *IIE Transactions*, *29*, 53–67.

Akturk, M. S., & Turkcan, A. (2000). Cellular manufacturing system design using a holonistic approach. *International Journal of Production Research*, *38*(1), 2327–2347.

Askin, R. G., & Chiu, K. S. (1990). A graph partitioning procedure for machine assignment and cell formation in group technology. *International Journal of Production Research*, *28*(8), 1555–1572.

Askin, R. G., Cresswell, S. H., Goldberg, J. B., & Vakharia, A. J. (1991). Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing. *International Journal of Production Research*, *29*, 1081–1100.

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

Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, *6*(2), 154–160.

Bertsekas, D., & Tseng, P. (1988). Relaxation methods for minimum cost ordinary and generalized network flow problems. *Operations Research*, *36*, 93–114.

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, C. H. (1991). A close neighbor algorithm for designing cellular manufacturing systems. *International Journal of Production Research*, *29*(10), 2097–2116.

Brown, E. C., & Sumichrast, R. C. (2001). CF-GGA: a grouping genetic algorithm for the cell formation problem. *International Journal of Production Research*, *36*(16), 3651–3669.

Burbidge, J. L. (1979). *Group Technology in Engineering Industry*. London: Mechanical Engineering Publications.

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

Caux, C., Bruniaux, R., & Pierreval, H.:. (2000). Cell formation with alternative process plans and machine capacity constraints: A new combined approach. *International Journal of Production Economics*, *64*(1-3), 279–284.

Chan, H. M., & Milner, D. A. (1982). Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing System*, *1*, 65–75.

Chandrashekharan, M. P., & Rajagopalan, R. (1986a). An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research*, *24*(2), 451–464.

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

Chandrasekharan, M. P., & Rajagopalan, R. (1987). ZODIAC—An algorithm for concurrent formation of part families and machine cells. *International Journal of Production Research*, *25*(6), 835–850.

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

Cheng, C. H., Gupta, Y. P., Lee, W. H., & Wong, K. F. (1998). A TSP-based heuristic for forming machine groups and part families. *International Journal of Production Research*, *36*(5), 1325–1337.

Chen, J., & Heragu, S. S. (1999). Stepwise decomposition approaches for large scale cell formation problems. *European Journal of Operational Research*, *113*(1), 64–79.

Choobineh, F. (1988). A framework for the design of cellular manufacturing systems. *International Journal of Production Research*, *26*(7), 1161–1172.

Chu, C. H., & Tsai, M. (1990). A comparison of three array-based clustering techniques for manufacturing cell formation. *International Journal of Production Research*, *28*(8), 1417–1433.

Co, H. C., & Araar, A. (1988). Configuring cellular manufacturing systems. *International Journal of Production Research*, *26*(9), 1511–1522.

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), 1–19.

Goldberg, D. E. (1989). *Genetic Algorithms in Search Optimization, and Machine Learning*. Reading: Addison-Wesley.

Gunasingh, K. R., & Lashkari, R. S. (1989). Machine grouping problem in cellular manufacturing systems—an integer programming approach. *International Journal of Production Research*, *27*(9), 1465–1473.

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.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, Michigan: The University of Michigan Press.

Joines, J. A., Culbreth, C. T., & King, R. E. (1996). Manufacturing cell design: An integer programming model employing genetic algorithms. *IIE Transactions*, *28*, 69–85.

Khan, M., Islam, S., & Sarker, B. (2000). A similarity coefficient measure and machine-parts grouping in cellular manufacturing systems. *International Journal of Production Research*, *38*(3), 699–720.

Khator, S. K., & Irani, S. K. (1987). Cell formation in group technology: A new approach. *Computers in Industrial Engineering*, *12*(2), 131–142.

King, J. R. (1980). Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm. *International Journal of Production Research*, *18*(2), 213–232.

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

Kumar, K. R., & Chandrasekharan, M. P. (1990). Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research*, *28*(2), 233–243.

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

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

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

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

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

Lee, H., & Garcia-Diaz, A. (1993). A network flow approach to solve clustering problems in group technology. *International Journal of Production Research*, *31*(3), 603–612.

McAuley, J. (1972). Machine grouping for efficient production. *Production Engineer*, *51*(2), 53–57.

McCormick, W. T., Schweitzer, P. J. & White, T. W. (1972). Problem decomposition and data reorganization by a clustering technique. *Operations Research, 20*, 993-1009.

Miltenburg, J. & Zhang, W. (1991). A comparative evaluation of nine well-known algorithms for solving the cell formation in group technology. *Journal of Operations Management*, *10*(1), 44–72.

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

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

Ng, S. (1993). Worst-case analysis of an algorithm for cellular manufacturing. *European Journal of Operational Research*, *69*(3), 384–398.

Ng, S. (1996). On the characterization and measure of machine cells in group technology. *Operations Research*, *44*(5), 735–744.

Onwubolu, G. C., & Mutingi, M. (2001). A genetic algorithm approach to cellular manufacturing systems. *Computers and Industrial Engineering*, *39*(1-2), 125–144.

Plaquin, M., & Pierreval, H. (2000). Cell formation using evolutionary algorithms with certain constraints. *International Journal Of Production Economics*, *64*(1–3), 267–278.

Rajagopalan, R., & Batra, J. L. (1975). Design of cellular production systems: a graph-theoretic approach. *International Journal of Production Research*, *13*(6), 567–579.

Sarker, B. R. (2001). Measures of grouping efficiency in cellular manufacturing systems. *European Journal of Operational Research*, *130*(3), 588–611.

Sarker, B. R., & Mondal, S. (1999). Grouping efficiency measures in cellular manufacturing: A survey and critical review. *International Journal of Production Research*, *37*(2), 285–314.

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, P. M. (1986). Application of the similarity coefficient method in group technology. *IIE Transactions, 18*, 3, 266-270.

Selim, H. M., Askin, R. G., & Vakharia, A. J. (1998). Cell formation in group technology: Evaluation and directions for future research. *Computers and Industrial Engineering*, *34*(1), 3–20.

Shtub, A. (1989). Modelling group technology cell formation as a generalized assignment problem. *International Journal of Production Research*, *27*(5), 775–782.

Spears, W. M., & DeJong, K. A. (1991). *On the virtues of parameterized uniform crossover in Proceedings of the Fourth International Conference on Genetic Algorithms* pp. 230–236.

Srinivasan, G., & Narendran, T. T. (1991). GRAFICS-A nonhierarchical clustering-algorithm for group technology. *International Journal of Production Research*, *29*(3), 463–478.

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.

Uddin, M. K., & Shanker, K. (2002). Grouping of parts and machines in presence of alternative process routes by genetic algorithm. *International Journal of Production Economics*, *76*(3), 219–228.

Vannelli, A., & Kumar, K. R. (1986). A method for finding minimal bottleneck cells for grouping part-machine families. *International Journal of Production Research*, *24*(2), 387–400.

Vohra, T., Chen, D., Chang, J., & Chen, H. (1990). A network approach to cell formation in cellular manufacturing. *International Journal of Production Research*, *28*(11), 2075–2084.

Waghodekar, P. H., & Sahu, S. (1984). Machine-component cell formation in group technology MACE. *lnternational Journal of Production Research 22,* 937-948.

Wemmerlov, U., & Hyer, N. L. (1989). Cellular manufacturing in the US industry: a survey of users. *International Journal of Production Research*, *27*(9), 1511–1530.

Won, Y. (2000). Two-phase approach to GT cell formation using efficient p-median formulations. *International Journal of Production Research*, *38*(7), 1601–1613.

Yasuda, K., & Yin, Y. (2001). A dissimilarity measure for solving the cell formation problem in cellular manufacturing. *Computers and Industrial Engineering*, *39*(1-2), 1–17.

Zhao, C., & Wu, Z. (2000). A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives. *International Journal of Production Research*, *38*(2), 385–395.