A Constructive Evolutionary Approach to the Machine-Part Cell Formation Problem

Geraldo Ribeiro Filho
Universidade de Mogi das Cruzes-CCET
Av. Cândido Xavier Almeida Souza, 200
08780-911 - Mogi das Cruzes – SP - Brazil
geraldo@lac.inpe.br

Luiz Antonio Nogueira Lorena
LAC/INPE- Instituto Nacional de Pesquisas Espaciais
Caixa Postal 515
12201-970 - São José dos Campos – SP - Brazil
lorena@lac.inpe.br

Abstract

This paper presents a constructive evolutionary approach to the machine-part cell formation (MPCF) problem, generally considered in manufacturing cell design, where a zero-one machine-part matrix must have its rows and columns moved to form machines and parts clusters. The Constructive Genetic Algorithm (CGA) was proposed recently to solve clustering problems, and is applied here to the MPCF. The MPCF is modeled as a bi-objective problem that guides the construction of feasible assignments of machines and parts to specify clusters, and provides evaluation of schemata and structures in a common basis. A particularly derived structure and schema representation considers Jaccard distances for binary strings. A variable size population is formed only by schemata, considered as building blocks for feasible solutions construction along the generations. Recombination gives population diversification, and local search mutation is applied to structures. Experimental results are shown for instances specially generated and others taken from the literature.

Key words: genetic algorithms, manufacturing cell design, p-median problem.

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. One of these approaches is the application of the "*group technology*" (Burbidge,1969) to decompose manufacturing systems into manageable sub-systems, or groups, by aggregating similar parts into part families and machines into cells. The ideal cell is independent and completely manufactures its part family(s). Automation and control are simplified through the creation of independent cells. The production flow analysis of Burbidge (1963) is one of the first and well-known methodologies associated with group technology.

There are many methods that works over a machine-part matrix with elements being zeros or ones, indicating which machines are used to produce each part. Given a matrix *A* (*figure 1*), where the rows corresponds to parts and columns to machines and $a_{ij} = 1$, if the part *i* needs the machine *j* to be produced. Basically, the algorithms change rows and columns positions to produce blocks of ones, forming parts families and machine cells simultaneously (*figure 2*).

Chandrasekharan and Rajagopalan (1989) and Venugopal and Narendran (1993) present some analysis over the zero-one matrix to extract properties and to advise for cell formation algorithms. Other algorithms following these lines can be viewed in the papers of McCormick (1972), King (1980, 1982) and Chu and Tsai (1990).

Many other techniques have been proposed in literature. Hierarchical clustering methods (Stanfel, 1985 and McAuley, 1972), non-hierarchical clustering (Chandrasekharan and Rajagopalan, 1986), graph based techniques (Rajagopalan and Batra, 1975), neural networks (Malave and Ramchandran, 1991), fuzzy logic (Xu and Wang, 1989) and *metaheuristics* like Simulated Annealing (Boctor, 1991 and Venugopal, and Narendran, 1992) and Genetic Algorithms (Joines, 1993).

Genetic Algorithms (*GAs*) are very well known, having several applications to general optimization and combinatorial optimization problems (Davis,1991; De Jong, 1975; Goldberg, 1989; Holland, 1975; Michalewicz, 1996). A typical *GA* is based on the controlled evolution of a structured population, recombination operators and the schema formation and propagation over generations.

This paper presents an application of a *Constructive Genetic Algorithm* (*CGA*) to solve the *MPCF* problem. The application is made through an analogy with the *p*-median problem, since both are clustering problems. The search for *p* median vertices on a network (graph) is a classical location problem. The objective is to locate *p* facilities (medians) minimizing the sum of the distances from each demand point to its nearest facility. Very good results were obtained by Lorena and Furtado (1998) using the *CGA*.

The *MPCF* is modeled as a bi-objective p-median problem that is used as a basis to construct feasible assignments of machines and parts to specified clusters, and considers evaluation of schemata and structures in a common basis. A particularly derived structure and schema representation considers *Jaccard* distances for binary strings. A variable size population is formed only by schemata, considered as building blocks for feasible solutions construction along the generations. Recombination gives population diversification, and a kind of local search mutation is applied to the generated structures representing feasible solutions.

A *CGA* review is presented in section two, detailing the schemata representation, the *fg-fitness* and the selection, recombination and mutation processes. Section three presents computational tests considering two instances from the literature and specially generated instances, providing insights about the *CGA* performance on instances of increasing difficulty.

2. **CGA review**

The *CGA* is proposed to address the problem of evaluating schemata and structures in a common basis. While in the other evolutionary algorithms, the evaluation of individuals is based on a single function (the fitness function), in *CGA* this process relies on two functions, mapping the space of structures and schemata onto $\Re_+$.
We resume the *CGA* in this section (for a complete description see the paper of Lorena and Furtado (1998, available on http://www.lac.inpe.br/~lorena/cga/cga_clus.PDF)).

2.1 Representation

For *schema representation*, it was used a string of $n+m$ symbols, where n is the number of parts (or rows in the original matrix) and m is the number of machines (or columns in the matrix). Each of the two portions of the schema evolves independently from each other. This application considers only

one part family for each machine cell. The number k of clusters (part families or machine cells) to be formed must be defined *a priori*.
The schemata have in each position one of the following three possible symbols:

1, to indicate a median part;
0, to indicate a non-median part assigned to its nearest median; and
#, to indicate a non-median part not yet assigned to a median.

Then, both portions of the schemata (parts and machines) must have exactly k positions with the symbol 1 and the rest with 0's or #'s. A schema with no #'s is an *structure* that represents a feasible solution, where every non-median part is assigned to its nearest median, and the same for the columns.

For instance, if we need to form three manufacturing cells, a schema for a problem with 10 parts and 15 machines could be represented by

s_i = (0,1,#,1,0,#,1,0,#,0 / 0,0,1,#,#,0,1,#,1,#,0,0,0,#,0).

Where the first 10 symbols represent parts and the last 15 symbols represent machines. Let $V_1^p(s_i)=\{2,4,7\}$ be the median part set, $V_2^p(s_i)=\{1,5,8,10\}$ the assigned non-median part set, and for machines, $V_1^m(s_i)=\{3,7,9\}$, $V_2^m(s_i)=\{1,2,6,11,12,13,15\}$.

Using the following notation: $V_1^p(s_i)=\{z_1^p,\ldots,z_k^p\}$ and $V_1^m(s_i)=\{z_1^m,\ldots,z_k^m\}$, the part clusters will be formed assigning elements from $V_2^p(s_i)$ to elements from $V_1^p(s_i)$ and the machine clusters assigning elements from $V_2^m(s_i)$ to elements from $V_1^m(s_i)$. The assignment of each element from $V_2^p(s_i)$ to the nearest element from $V_1^p(s_i)$ (each element from $V_2^m(s_i)$ to the nearest from $V_1^m(s_i)$) is made based on the *Jaccard distance* among them (represented here by $m_{z_j q}$).

The *Jaccard* similarity coefficient for two binary strings is defined as the number of positions with value 1 in both sequences divided by number of positions with value 1 in both or one of the sequences. This coefficient is used as a "distance" measure subtracting it from one.

The part clusters $C_1^p(s_i), C_2^p(s_i), \ldots, C_k^p(s_i)$ and machine clusters $C_1^m(s_i), C_2^m(s_i), \ldots, C_k^m(s_i)$ are formed after the assignments.

2.2. The fg-fitness

In general terms, after the formation of clusters $C_1^p(s_i), C_2^p(s_i), \ldots, C_k^p(s_i)$ and $C_1^m(s_i), C_2^m(s_i), \ldots, C_k^m(s_i)$, the function f and the function g are computed as follows:

$$g(s_i) = \sum_{j=1}^{k} \sum_{q \in C_j^p(s_i)} m_{z_j q} + \sum_{j=1}^{k} \sum_{q \in C_j^m(s_i)} m_{z_j q} \text{, and}$$

$$f(s_i) = g(s_i) - \sum_{j=1}^{k} \max_{q \in C_j^p(s_i)} \left\{ m_{z_j q} \right\} - \sum_{j=1}^{k} \max_{q \in C_j^m(s_i)} \left\{ m_{z_j q} \right\}$$

A common upper bound to f and g will be necessary at the evolution process. To compute this upper bound g_{max}, in the very beginning of the process, a structure s_{random} representing a feasible solution (no #'s) is randomly generated and $g(s_{random})$ is taken as the g_{max} value.

2.3. Selection and recombination

The population is kept ordered according to "completeness" of the schema, i.e., the number of labels #s, and the schema fg-fitness. The schemata in population are non-decreasing ordered using the key $\Delta(s_i) = \dfrac{1+d_i}{n(s_i)}$, where *n(s_i)* is the number of labels different from # in s_i, and $d_i = \dfrac{[\mathrm{g(si)} - \mathrm{f(si)}]}{\mathrm{g(si)}}$.

The method used for selection takes the first schema from the best part of the population (*base*) and the second one from the whole population (*guide*). Before recombination, the first schema is complemented to generate a structure representing a feasible solution, i.e., all #'s are replaced by 0's. This complete structure suffers mutation and is compared to the best one found so far. Only the best one is kept along the process. The recombination merges information from both selected schemata, but preserves the number of medians in both portions (parts and machines) of the new generated schema or structure. If it is a new schema then it is inserted into the population, otherwise it suffers mutation and is compared to the best one found so far.

The recombination is best described in the following. The assignment operations must be performed in that order.

RECOMBINATION

$S_{base}(j) = \#$ and $S_{guide}(j)=\#$ then $S_{new}(j)$ ¬ #
$S_{base}(j) = 1$ and $S_{guide}(j)=1$ then $S_{new}(j)$ ¬ 1
$S_{base}(j) = 0$ and $S_{guide}(j)=0$ then $S_{new}(j)$ ¬ 0
$S_{base}(j) = 1$ and $S_{guide}(j)=\#$ then $S_{new}(j)$ ¬ 1
$S_{base}(j) = 0$ and $S_{guide}(j)=\#$ then $S_{new}(j)$ ¬ 0
$S_{base}(j) = \#$ and $S_{guide}(j)=0$ then $S_{new}(j)$ ¬ 0
$S_{base}(j) = \#$ or 0 and $S_{guide}(j)=1$ then
    $S_{new}(j)$ ¬ 1 and $S_{new}(k)$ ¬ 0 for some $S_{new}(k)=1$
$S_{base}(j) = 1$ and $S_{guide}(j)=0$ then
    $S_{new}(j)$ ¬ 0 and $S_{new}(k)$ ¬ 1 for some $S_{new}(k)=0$

The mutation process used was a technique that implements successive changes in the median position inside each cluster followed by cluster reconstruction made by vertex reallocation. The following pseudo-code is more illustrative.

MUTATION

```
Begin:
  S' ¬ S;
  For each part cluster in S'
        Move the median to the cluster vertex which gives the minimum distance sum in the cluster;
  For each machine cluster in S'
        Move the median to the cluster vertex which gives the minimum distance sum in the cluster;
  If  S is better than S'
        return S;
  else
        S'' ¬  S' with non-median vertex realocation;
        If S''  is better than  S'
              S ¬  S'';
```

Goto Begin;

else

return S';

End:

At each generation, after new schemata insertion, the population is scanned to remove all schemata satisfying the condition $a \geq \frac{dg_{\max} - [g(s_i) - f(s_i)]}{d[g_{\max} - g(s_i)]} = d(s_i)$, where d is a real number satisfying $0 < d \leq 1$ The evolution parameter a is initially set to zero and slowly increased at each generation. For all computational tests, an initial population was randomly created with 20% of the rows and columns in each schema with symbols 0 and exactly *k* (number of part families or machine cells) with symbols 1.

2.4. The algorithm

The Constructive Genetic Algorithm can be summed up by the pseudo-code (see http://www.lac.inpe.br/~lorena/cga/cga_clus.PDF for details):

CGA

Given g_{max} and d ;
$\alpha := 0$;
$\varepsilon := 0.05$; *{ time interval }*
Initialize P_α ; *{ initial population }*
Evaluate P_α ; *{ fg-fitness }*
For all $s_i \hat{I} P_a$ *compute* $d(s_i)$ *{ rank computation }*
end_for
While (not stop condition) **do**
  For all $s_i \hat{I} P_a$ satisfying $\alpha < d(s_i)$ **do** *{ evolution test }*
    $\alpha := \alpha + \varepsilon$;
    Select P_α from $P_{\alpha-\varepsilon}$; *{ reproduction operator }*
    Recombine P_α ; *{ recombination operators }*
    Evaluate P_α ; *{ fg-fitness }*
  end_for
  For all new $s_i \hat{I} P_a$ *compute* $d(s_i)$ *{ rank computation }*
  end_for
end_while

---

3. Experimental Results

Most of the problem instances were randomly generated for the computational tests. From the literature were taken two instances, one of them with a 20x35-part/machine matrix (Burbidge, 1969) and the other one with a 40x100 matrix (Chandrasekharan and Rajagopalan, 1989).

For *performance measure* was considered 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:

$$Coef = \frac{e - e_1}{e + e_0}$$

where: e = number of 1's in the matrix
e_0 = number of 0's inside the clusters
e_1 = number of 1's outside the clusters

The ideal coefficient value is 1 (no zeros inside and no ones outside the clusters), and better clustering has greater coefficient value.

To generate the instances, the number of parts, number of machines, within-cellular density (*WCD*) and inter-cellular density (*ICD*) were specified. The *WCD* is the ratio between the number of 1's inside the cluster and its size. The *ICD* is the ratio between the number of 1's outside the clusters and the number of matrix elements outside any cluster. Initially the matrix is generated with the clusters as specified, and then the matrix is perturbed randomly changing the rows and columns positions. The final form of the matrix can be used for the algorithm test.

Table 1 shows the results obtained with the instances taken from the literature, with three runs for each instance. The performance coefficient values obtained were the same best values found in the literature (Joines, 1993). *Table 2* shows the results obtained on randomly generated instances that differ from each other by the inter-cellular density. The purpose was basically verifying the algorithm sensibility under different inter-cellular densities. The performance coefficient values obtained can be compared to the values for the original cluster formation, computed previously by the instance generation program. *Table 3* shows the results obtained with randomly generated instances with different within-cellular density. Also, the purpose was verifying the algorithm sensibility. In both cases, it seems to indicate that *ICD* and *WCD* has no effect over *CGA* performance.

All the tests were made using ε=0.01 as the α increment, and d = 0.1 as the overall proportional deviation from $g_{\max}$.

Instance	*Part/ Machine*	*Cells*	*Coef Literature*	*Coef CGA*
Burbidge	20/35	4	0.7571	0.7571
				0.7571
				0.7571
Chandra	40/100	10	0.8403	0.8403
				0.8403
				0.8403

Table 1: Tests using instances from the literature

Instance	*Part/ Machine*	*Cells*	*WCD*	*ICD*	*Coef Original*	*Coef CGA*
W80i02	*20/35*	*4*	*0.8*	*0.02*	*0.7527*	*0.7527*
						0.7527
						0.7527
W80i03	*20/35*	*4*	*0.8*	*0.03*	*0.7330*	*0. 7330*
						0. 7330
						0. 7330
W80i05	*20/35*	*4*	*0.8*	*0.05*	*0.6931*	*0. 6931*
						0. 6931
						0. 6931
W80i10	*20/35*	*4*	*0.8*	*0.10*	*0.6140*	*0. 6140*
						0. 6140
						0. 6140

Table 2: Tests for ICD sensibility

Instance	*Part/ Machine*	*Cells*	*WCD*	*ICD*	*Coef Original*	*Coef CGA*
W70i02	*20/35*	*4*	*0.7*	*0.02*	*0.6398*	*0. 6398*
						0. 6398
						0. 6398
W80i02	*20/35*	*4*	*0.8*	*0.02*	*0.7527*	*0. 7527*
						0. 7527
						0. 7527
W90i02	*20/35*	*4*	*0.9*	*0.02*	*0.8280*	*0. 8280*
						0. 8280
						0. 8280

Table 3: Tests for WCD sensibility

4. Final considerations

This work describes an application of the *Constructive Genetic Algorithm - CGA* proposed by Lorena and Lopes (1996) to the clustering formation of parts and machines in manufacturing cells. The *CGA* provides the following new features to genetic algorithms, such as the direct evaluation of schemata, population dynamic in size and formed only by schemata and the new *fg-fitness* process.

The computational results obtained were very good; presenting performance measure values as good as those listed in the literature. The algorithm seems to be unaffected by within-cellular density or inter-cellular density variation.

Acknowledgments:
The second author acknowledges Conselho Nacional de Desenvolvimento Científico e Tecnológico -CNPq (proc. 350034/91-5, 520844/96-3, 680082/95-6) and Fundação para o Amparo a Pesquisa no Estado de S. Paulo - FAPESP (proc. 95/9522-0 e 96/04585-6) for partial financial support

References

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

Burbidge, J.L. Production flow analysis. **Production Engineer**, 42, 1963, p.742-752.

Burbidge, J.L. **An introduction of group technology**. In: Seminar on group Technology,Turin, 1969.

Chandrasekharan, M.P. and Rajagopalan, R. An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. **International Journal Production Research**, 24(2), 1986, p. 451-464.

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

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

Davis, L.D. **Handbook of Genetic Algorithms**. Van Nostrand Reinhold, New York, 1991.

De Jong, K. **An analysis of the behavior of a class of genetic adaptive systems**. Ph.D. thesis, University of Michigan, Ann Arbor, MI, 1975.

Goldberg, D.E. **Genetic algorithms in search, optimization and machine learning**. Addison-Wesley, Reading, MA, 1989.

Holland, J.H. **Adaptation in natural and artificial systems**. MIT Press, New York, 1975.

Joines, J.A. **Manufacturing cell design using genetic algorithms**. M.S. thesis, North Carolina State University, Raleigh, NC, 1993.

King, J.R. Machine-component grouping formation in group technology. **International Journal of Management Science**, 8(2), 1980, p.193-199.

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

Lorena, L. A N. and Furtado, J. C. **Constructive Genetic Algorithms for Clustering Problems. Evolutionary Computation** – submitted – Available at http://www.lac.inpe.br/~lorena/cga/cga_clus.PDF, 1998.

Lorena, L.A.N. and Lopes, F.B. **A dynamic list heuristic for 2D-cutting.** In: System Modelling and Organization, ed. J. Dolezal and J. Fidler, Chapman & Hall, London, p.481-488, 1996.

Malave, C. O. and Ramachandran, A neural network based design of cellular manufacturing system. **Journal of Intelligent Manufacturing**, 2, 1991, p.305-314.

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

McCormick Jr., W.T. ; Schweitzer, P.J. and White, T.W. Problem decomposition and data reorganization by a cluster technique. **Operations Research**, 20(5), 1972, p. 993-1009.

Michalewicz, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. Springer-Verlag, Berlin, 1996.

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

Stanfel, L. E. Machine clustering for economic production. **Engineering Costs and Production Economics**, 9, 1985, p.73-81.

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

Venugopal, V. and Narendran, T. T. Design of cellular manufacturing systems based on asymptotic forms of a Boolean matrix. **European Journal of Operational Research**, 67, 1993, p.405-417.

Xu, H. and Wang, H. P. Part family formation for gt applications based on fuzzy mathematics. **International Journal Production Research**, 27(9), 1989, p.1637-1651.