



Group Technology: Hybrid Genetic Algorithm with Greedy Formation and a Local Search Cluster Technique in the Solution of Manufacturing Cell Formation Problems

Rogério Malta Branco^(✉) and Carlos Rodrigues Rocha

IFRS Campus Rio Grande, Rio Grande, RS, Brazil
rogerio.branco@riogrande.ifrs.edu.br

Abstract. Group Technology (GT) is a manufacturing philosophy that explores similarities in product and process design. Starting from a binary machine-part matrix, the objective is to form clusters, made up of families of parts and machine cells aiming to minimize the number of voids and exceptional elements in the cells. Since this is a combinatorial problem, the proposed hybrid genetic algorithm (GA) finds good solutions with a partial greedy population that uses similarities with machines and with parts. A local search k-means based method can recreate cells with small movements among machine assignments. The proposed framework performance presents good results, most of them overcoming literature classic problems solutions, when considering the group effectiveness indicator. The results are presented and discussed.

Keywords: Genetic algorithm · Local search · Group Technology · K-means · Cell Formation Problems

1 Introduction

The Group Technology (GT) is a manufacturing philosophy based on the principle of identifying and grouping machines and parts by similarity, leading the production systems to obtain advantages throughout all stages of design and manufacturing [1].

Applying the GT in a productive process will result in a binary machine-part incidence matrix. The cellular manufacturing layout consists in to arrange these elements into cells, seeking to minimize extracellular movements and maximize the intracellular ones. The Manufacturing Cell Formation Problems - MCPF's are considered NP-arduous because of their combinatorial nature and the literature related presents a considerable number of methods applied in their solution, where some of the best known and used for quality comparison are: rank order clustering – ROC [2], modified rank order clustering – MODROC [3], GA [4], SA [5], which uses the grouping efficacy index to measure the grouping effectiveness.

In clustering problems, the modified algorithms with local search techniques are inserted in basic heuristics, overcoming them in terms of grouping effectiveness. The proposed framework employs these hybridizations in a GA, in the search for solutions for the MCFPs, being specifically the method k-means the one chosen to improve the

individuals of the population. As objective function, the coefficient of effectiveness, known in the literature as being a good index of clusters of cellular performance, is applied as well as being used in the GT problems used to compare results.

2 Machine-Part Cell Formation Problem

Group Technology gives similar treatment to similar elements, that is, it divides the manufacture into small groups (cells) of machines that will process pieces with greater similarity between themselves (families).

Thus, the parts of that family will always be processed by the same machines and tool changes will have reduced configuration time, also reflecting in the same way in the processing time of the system.

The application of the GT in a production system consists of asserting several steps and, in an inner process of the Production Flow Analysis (AFP), a binary matrix is generated relating existing pieces and the corresponding machines that process them. This matrix is called incidence, and at this stage it is necessary to relate those of greater similarity so that the parts families and machine cells are constructed, known as Manufacturing Cell Formation Problem (MCFP).

With the main objective of assigning machines to cells and parts for families, this arrangement needs a function to guide the performance of these groupings. Some are well known, such as the use of machines [6], clustering efficiency [7], clustering efficacy [8], etc. According to [6], two are the most used: efficiency and effectiveness of grouping.

The grouping efficacy (μ) mentioned below is adopted for two reasons: first because it overcomes the weaker discriminating power of grouping efficiency measure by assigning equal weight for the number of voids and the number of exceptional elements; the second is because the results obtained in the works used as performance comparison apply this group quality indicator. This measure is defined as follows:

$$\mu = \frac{e - e_0}{e + e_v} \quad (1)$$

where e is the total number of operations (1's) in the given matrix, e_v is the number of voids (0's in the diagonal groups), and e_0 is the number of exceptional elements (1's out the diagonal groups).

Also, [9] and [10] justify this index adoption since it: incorporates both the within-cell machine use and the inter-cell movement; generates block diagonal matrices which are interesting in practice; is independent from the number of cells, among others.

In the Fig. 1, there is a one-piece binary incidence matrix. Reordered, to form machine cells and parts families, it has voids (in green) and exceptional elements (in red). An empty means that although machine and part have been assigned the respective cell and family, the machine will not process the part. An exceptional element implies intercellular movement, since the part will have some processing in another cell, increasing the processing time, among other complications.

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

Fig. 1. Example of a reordered two cell incidence matrix.

Considering the example in the Fig. 1, the Eq. 2 shows the calculus of grouping efficacy (μ) coefficient, since the e , e_0 and e_v can be easily found:

$$\mu = \frac{18 - 3}{18 + 3} = 0,7143 = 71,43\% \tag{2}$$

3 The Proposed Heuristic for Cell Formation Problems

The framework presented in this paper solves MCFP problems by applying a hybrid genetic algorithm that uses a local search method to refine the solutions.

With an initial population partially formed by a greedy constructor method, the convergency can be reached faster than only using random construction methods. In addition, procedures are used to accelerate these convergences, like the neighborhood research, where an GA solution is used as a starting point for another algorithm: a local search method as the k-means algorithm, applied to a chromosome, trying to improve its cluster efficiency.

3.1 The Genetic Algorithm with Local Search

Combining the survival of individuals with each other and inspired by natural and genetic selection mechanisms. This general theory of systems with robust adaptation finds an excellent practical application in the optimization of mathematical functions.

GAs differ from other heuristics by having distinct characteristics: act on a set of points (population) and not on isolated points; operate in a space of coded solutions and not directly in the search space; they need, as information, only the value of an objective function (function of adaptability or suitability); use probabilistic transitions rather than deterministic rules [11].

Briefly, a GA begins with an initial population and the adaptation of the chromosomes is calculated. Genetic operators are applied to selected individuals (a better adaptation implies a greater chance of selection), based on their suitability, and a new generation of individuals is created. This procedure will be repeated until some final criterion is reached.

In the population, each individual is represented by a chromosome, denoting viable solutions to the problem. Thus, to this framework, each gene is a cell or family and each locus, a machine or part (depending on the portion of the analysed chain), thus having length equal to the number of machines plus the number of pieces presented in the $M \times N$ matrix, as it can be seen in Fig. 2.

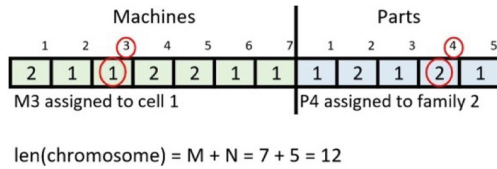


Fig. 2. Example of a chromosome denoting the two-cell formation shown in Fig. 1.

As mentioned, some genetic operators will be used: cloning, crossover and mutation. Cloning consists in the inclusion of the individual in the next generation. In other hand, crossover combines information of chromosomes from selected individuals, generating new ones. Otherwise, with low probabilistic rate, the mutation randomly disrupts the machine-cell designation, trying to avoid local maxima. The selection technique is the Roulette Method [6], with selection proportional to the fitness.

After selection, as can be seen in Fig. 3, and occurring only in the chromosomes’ “machine” portion, the genetic combination draws a cut position where the data origins from one parent and after, from the other one. A constructor algorithm, driven by these cells’ segments, will construct the families’ ones.

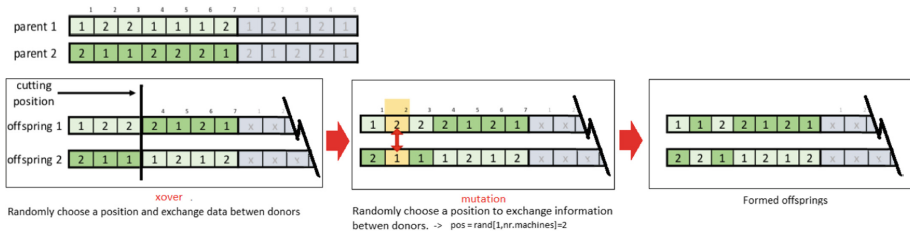


Fig. 3. The genetic operators: crossover and mutation.

The refinement of the individuals is used before a new iteration starts. As mentioned, the procedure is the k-means, which is a non-hierarchical method that aims to produce partitions in a set of objects with prior knowledge of the quantity of these.

The procedure starts with k-centroids defined at random, and, in successive iterations, each element is grouped according to some criterion, such as least squares. Because it is an iterative method, centroids are constantly recalculated until they no longer change.

Just as the calculation of the distance between the points is necessary, since cohesive groups are desired around a common point (centroid), it is also necessary to

calculate the similarity between the machines. This coefficient calculation consists in the coincidences between them, so a XNOR logic (Eq. 3) between the binary vectors is calculated, where a nonzero bit in S_{ij} indicates that both machines are equal in terms of part processing, so the sum of the elements of S_{ij} implies that the higher the result, the greater the similarity between the evaluated elements (vectors R_i and R_j in Eq. 3).

$$S_{ij} = (R_i \text{ XNOR } R_j) \quad (3)$$

With a totally generated chromosome, the k-means algorithm starts from known centroids: the cells already characterized. The objective is to carry out movements, in order to improve these groupings. Because it is costly, this local search has a stop criterion of only 1 iteration.

The centroids calculation promotes exchanges based on the distances between the elements. After reconstruction of the “machines” chromosomes fraction, the constructor procedure creates the “pieces” segment. Recalculating the adaptation, if better than the original, will replace it in the current population.

3.2 The Greedy Constructive Heuristic

Like the genetic operators, also the greedy algorithm of manufacturing cell formation will only act on the chromosomes’ machine segment and so, it begins by calculating the distance matrix between the machines, where each row of the incidence matrix is a n-dimensional point. The Eq. 4 shows the distance equation, where x_i and y_i are the vectors; $i = \{1, \dots, n\}$ is the position index of the vectors.

$$d = \sum_{i=1}^n \|x_i - y_i\| \quad (4)$$

For the initial cells’ formation, 5 pairs of machines are grouped with great distances from each other, where a random choice will compose the first cell.

The other cells will only have a single machine: that one which accumulates the biggest distance between all the already allocated, for all the cells. Once defined all these seeds of cells, the algorithm designates each not yet allocated machine to that cell with the shortest accumulated distance to its members. In cases of tie, a random choice is made. In cases of tie, a random choice is made.

3.3 The Part-Segment Constructor Algorithm

Based on the calculation grouping efficiency’s coefficient [3], the training strategy of [9] that tries to maximize the grouping coefficient considering the association of a part to a family. This iterative procedure, as presented by the authors in Eq. 5, computes the effect of allocating each piece to a family. As such, the one that maximizes the function is chosen and ends when there are no more allocations to be made.

$$F^* = \arg \max \left\{ \frac{N_1 - N_{1,F}^{out}}{N_1 + N_{0,F}^{in}} \right\} \quad (5)$$

where N_j is the total number of operations (1's) in the given matrix, $N_{1,F}^{in}$ is the number of voids (0's in the diagonal groups for the association of a part i to a family f), and $N_{1,F}^{out}$ is the number of exceptional elements (1's out the diagonal groups for the same association of a part i to a family F).

4 Method and Computational Results

The method applied by the framework consists in to form an initial population which has half part of the individuals made by a greedy method that uses similarities of parts and machines to compose the chromosomes. The complement of the population is formed by individuals of random constitution, aiming to counterbalance the homogeneity generated by the other method of formation.

The test-bed of the work consists in the application of the framework on 25 problems provided by [6] and related in Table 1.

Table 1. Problems obtained from the literature for analysis.

Problems	Autor (source)	Size
p01	King and Nakornchai [7]	5×7
p02	Waghodekar and Sahu [8]	5×7
p03	Seifoddini [9]	5×18
p04	Kusiak and Cho [10]	6×8
p05	Kusiak and Chow [11]	7×11
p06, p07	Chandrasekharan and Rajagopalan [12]	8×20
p08	Mccormick et al. [13]	16×24
p09	Srinivasan et al. [14]	16×30
p10	King [2]	16×43
p11	Carrie [1]	18×24
p12	Mosier and Taube [15]	20×20
p13	Kumar et al. [16]	20×23
p14	Carrie [1]	20×35
p15, p16, p17, p18, p19	Chandrasekharan and Rajagopalan [12]	24×40
p20	Paydar [17]	27×27
p21	Carrie [1]	28×46
p22	Kumar and Vannelli [18]	30×41
p23	Stanfel [9]	30×50
p24	Paydar [17]	53×37
p25	Chandrasekharan and Rajagopalan [12]	100×40

The authors also analyze the results from GA [4], Zodiac [19], Grafics [20], MST [21], GATSP [22]. Their own proposal, here denominated G&R is also compared with this framework results, all related in Table 2. The formation of a population is 5% of the best individuals (clones), 50% of selected individuals for crossing, and the rest, with 80% random and 20% greedy formation. The population size is 150 individuals, regardless of the size of the problem, as well as the mutation rate, equals to 2%. For each problem, 20 rounds of the GA are performed, statistics are generated and comparative data between the framework results and the literature are also related.

Table 2. Comparison between cluster efficiency indexes obtained from the literature, by different methods and the presented framework.

	C ^a	Literature best efficiency grouping index						Best	Diff (%)	Generations ^c			Time (s)		
		Zodiac	Grafics	MST	GATsp	GA	G&R			Ftns ^b	Min	Avg	Max	Min	Avg
p1	2	73,68	73,68				73,68	81,25	10,27	0	0,0	0	0,01	0,01	0,02
p2	2	56,52	60,87			62,50	62,5	69,57	11,31	0	0,0	0	0,01	0,01	0,02
p3	2	77,36			77,36	77,36	79,59	79,59	0,00	0	0,0	0	0,02	0,03	0,04
p4	2	76,92			76,92	76,92	76,92	76,92	0,00	0	0,0	0	0,01	0,02	0,02
p5	3	39,13	53,12		46,88	50,00	53,13	56,25	5,87	0	1,0	2	0,02	0,06	0,09
p6	3	85,24	85,24	85,24	85,24	85,25	85,25	85,25	0,00	0	0,0	0	0,03	0,03	0,05
p7	2	58,33	58,13	58,72	58,33	55,91	58,72	58,72	0,00	0	0,5	1	0,03	0,05	0,07
p8	6	32,09	45,52	48,70			52,58	51,96	-1,18	1	14,5	41	0,19	1,36	3,64
p9	4	67,83	67,83	67,83			67,83	67,83	0,00	1	1,0	1	0,14	0,14	0,15
p10	5	53,76	54,39	54,44	53,89		54,86	54,86	0,00	2	3,3	5	0,32	0,47	0,76
p11	6	41,84	48,91	44,20			54,46	54,95	0,90	2	29,5	98	0,25	2,73	8,75
p12	5	21,63	38,26		37,12	34,16	42,94	43,36	0,98	6	6,0	6	0,59	0,59	0,59
p13	5	38,66	49,36	43,01	46,62	39,02	49,65	49,25	-0,81	2	3,2	6	0,24	0,35	0,61
p14	4	75,14	75,14	75,14	75,28	66,30	76,22	76,14	-0,10	1	1,0	1	0,16	0,17	0,20
p15	7	85,11	85,11	85,11	85,11		85,11	85,11	0,00	1	1,0	1	0,25	0,26	0,29
p16	7	73,51	73,51	73,51	73,03	73,03	73,51	73,51	0,00	1	1,4	2	0,25	0,32	0,42
p17	9	20,42	20,42	51,81	49,37	37,62	51,97	52,83	1,65	2	5,5	15	0,48	1,15	2,86
p18	9	18,23	44,51	44,72	44,67	34,76	47,06	47,5	0,93	16	72,8	129	3,02	12,98	22,86
p19	9	17,61	41,67	44,17	42,50	34,06	44,87	45,45	1,29	25	72,8	129	4,64	13,10	23,05
p20	4	52,14	41,37	51,00			54,27	54,31	0,07	3	13,1	32	0,41	1,50	3,55
p21	9	33,01	32,86	40,00			44,62	46,91	5,13	70	109,	148	14,93	22,95	30,97
p22	11	33,46	55,43	55,29	53,8	40,96	58,48	61,27	4,77	3	6,8	15	0,88	1,82	3,79
p23	11	21,11	47,96	46,3	45,93	37,55	50,51	58,82	16,45	6	50,1	139	1,84	13,78	38,77
p24	2	52,21	52,21				56,42	59,16	4,86	10	77,6	145	1,89	13,39	24,65
p25	10	83,66	83,92	83,92	84,03	83,90	84,03	84,03	0,00	1	1,9	2	0,78	1,19	1,32

^aNumber of cells;

^bThe best results obtained during the application of the framework on the problems in the several consecutive rounds;

^cNumber of generations to obtain de better efficiency index (*).

The implementation of the framework is done in Python language on a i7-7700HQ notebook, with 2.8 GHz and 16 GBytes of RAM. Although not being the scope of the work, the implementation of algorithms in Python is feasible for building the necessary programming codes, being free of easy access to documentation, as well as good libraries of functions already created, such as Numpy, for example.

Also, operating on some of the problems studied, [23] obtained similar values to those found here. Although with a different strategy, they also considered the Euclidean distances between machines and, subsequently, cells.

5 Conclusion

In this study, a hybrid genetic algorithm is proposed with the objective of maximizing clustering efficiency in cell manufacturing problems, of NP-hard combinatorial nature. To obtain good solutions in reasonable computational time, local search techniques associated with genetic algorithms are applied. The present framework also proposes strategies of combined greedy and random formation for the initial populations, as well as constructive procedures of individuals managed by biased rules of pre-optimization, obtaining very satisfactory results in the questions of grouping of efficiency and computational time, pointing very favorable perspectives of using this structure in solving this type of problem.

The maximization of clustering efficiency performed by this framework is compared with 6 other proposals. Briefly, it is assumed that all the average values obtained exceeded the other applied methods mean values. Moreover, for 88% of the problems, the efficiency indicators were equal to or better than the best known in the literature (52% already in the first performed generation). Still, for more than half of the problems, the results surpassed those then known values of the literature, as show in Table 2.

In the formation of the initial population, it is observed that the greedy constructive algorithm consumes more processing time than the random one, besides taking the set of individuals to have a greater homogeneity. Furthermore, greater uniformity leads to worse results from construction, which reinforces the decision to define a randomness rate of 80% and 20% of greedy formation. In addition, in this composition, the best results already appear in the first iterations (initial generation and first iteration), without further improvements in the following iterations, of each test.

The k-means local search algorithm did not imply significant differences in the results when applied with rates higher than 75%, on the individuals generated in the current population. Thus, since its interference in computational cost is directly linked to the increase of this rate, it was limited to this value.

The Python language is a bit slower than compiled languages, but it is free, easy to develop, with ample support material, and the time taken to get the results is satisfactory. Moreover, the local research combined with a good construction procedure, even being computationally expensive, allow an accelerated convergence in the search for good solutions of these types of problems.

References

1. Carrie, A.S.: Numerical taxonomy applied to group technology and plant layout. *Int. J. Prod. Res.* **11**(4), 399–416 (1973)

2. King, J.R.: Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. *Int. J. Prod. Res.* **18**(2), 213–232 (1980)
3. Chandrasekharan, M.P., Rajagopalan, R.: MODROC: an extension of rank order clustering for group technology. *Int. J. Prod. Res.* **24**(5), 1221–1233 (1986)
4. Onwubolu, G.C., Mutingi, M.: A genetic algorithm approach to cellular manufacturing systems. *Comput. Ind. Eng.* **39**(1–2), 125–144 (2001)
5. Rao, P.K.: A multi stage heuristic for manufacturing cell formation. *Int. J. Res. Eng. Technol.* **1163**(45), 2308–2321 (2014)
6. Goldbarg, M.C., Luna, H.P.L.: *Otimização Combinatória e Programação Linear: Modelos e Algoritmos*. Elsevier, Rio de Janeiro (2005)
7. King, J.R., Nakornchai, V.: Machine-component group formation in group technology: review and extension. *Int. J. Prod. Res.* **20**(2), 117–133 (1982)
8. Waghodekar, P.H., Sahu, S.: Machine-component cell formation in group technology: MACE. *Int. J. Prod. Res.* **22**(6), 937–948 (1984)
9. Seifoddini, H.: Single linkage versus average linkage clustering in machine cells formation applications. *Comput. Ind. Eng.* **16**(3), 419–426 (1989)
10. Kusiak, A., Cho, M.: Similarity coefficient algorithms for solving the group technology problem. *Int. J. Prod. Res.* **30**(11), 2633–2646 (1992)
11. Kusiak, A., Chow, W.S.: Efficient solving of the group technology problem. *J. Manuf. Syst.* **6**(2), 117–124 (1987)
12. Chandrasekharan, M.P., Rajagopalan, R.: GROUPABILITY: an analysis of the properties of binary data matrices for group technology. *Int. J. Prod. Res.* **27**(6), 1035–1052 (1989)
13. McCormick, W.T., Schweitzer, P.J., White, T.W.: Problem decomposition and data reorganization by a clustering technique. *Oper. Res.* **20**(5), 993–1009 (1972)
14. Srinivasan, G., Narendran, T.T., Mahadevan, B.: An assignment model for the part-families problem in group technology. *Int. J. Prod. Res.* **281**(1), 145–152 (1990)
15. Mosier, C., Taube, L.: Weighted similarity measure heuristics for the group technology machine clustering problem. *Omega* **13**(6), 577–579 (1985)
16. Ravi Kumar, K., Kusiak, A., Vannelli, A.: Grouping of parts and components in flexible manufacturing systems. *Eur. J. Oper. Res.* **24**(3), 387–397 (1986)
17. Paydar, M.M., Saidi-Mehrabad, M.: A hybrid genetic-variable neighborhood search algorithm for the cell formation problem based on grouping efficacy. *Comput. Oper. Res.* **40**(4), 980–990 (2013)
18. Kumar, K.R., Vannelli, A.: Strategic subcontracting for efficient disaggregated manufacturing. *Int. J. Prod. Res.* **25**(12), 1715–1728 (1987)
19. Chandrasekharan, M.P., Rajagopalan, R.: Zodiac – an algorithm for concurrent formation of part-families and machine-cells. *Int. J. Prod. Res.* **25**(6), 835–850 (1987)
20. Srinivasan, G., Narendran, T.T.: GRAFICS – a nonhierarchical clustering algorithm for group technology. *Int. J. Prod. Res.* **29**(3), 463–478 (1991)
21. Srinivasan, G.: A clustering algorithm for machine cell formation in group technology using minimum spanning trees. *Int. J. Prod. Res.* **32**(9), 2149–2158 (1994)
22. Cheng, C.H., Gupta, Y.P., Lee, W.H., Wong, K.F.: A TSP-based heuristic for forming machine groups and part families. *Int. J. Prod. Res.* **36**(5), 1325–1337 (1991)
23. Laha, D., Hazarika, M.: A heuristic approach based on Euclidean distance matrix for the machine-part cell formation problem. *Mater. Today. Proc.* **4**(2), 1442–1451 (2017)