A PSO-based Clustering Algorithm for Manufacturing Cell Design

Orlando Durán
Escuela de Ing. Mecánica,
Pontificia Universidad Católica de Valparaiso, Chile
orlando.duran@ucv.cl

Nibaldo Rodriguez
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso, Chile
nibaldo.rodriguez@ucv.cl

Luiz Airton Consalter
FEAR Universidade de Passo Fundo,
Passo Fundo, RS,
Brasil
lac@upf.br

***Abstract*—**Since the last years different metaheuristic methods have been used to solve clustering problems. This paper addresses the problem of manufacturing Cell Formation using a modified particle swarm optimisation (PSO) algorithm. The main modification made to the original PSO algorithm consists on that in this work it is not used the vector of velocities as the standard PSO algorithm does. The proposed algorithm uses the concept of proportional likelihood with modifications, a technique that is used in data mining techniques. Some simulations are presented and compared. The criterion used to group the machines in cells is based on the minimization of inter-cell movements. The computational results show that the PSO algorithm is able to find the optimal solutions on almost all instances.

Keywords: Manufacturing cells, machine grouping, particle swarm optimization

I. INTRODUCTION

This Cellular Manufacturing is an organizational approach based on Group Technology (GT). Cellular manufacturing aims to divide the plant into a certain number of cells. Each cell contains machines that process similar types or families of products. Manufacturing Cells (MC) have been shown to provide considerable cost and productivity benefits to practical manufacturing environments. Other considerable number of benefits to be gained by grouping machines into cells can be found in literature.

The major issue in the design of manufacturing cells is the identification of machine and component groups. This identification process requires an effective approach to form part families so that similarity within a part family can be maximized. According to Selim et al. (1998) clustering analysis is the most frequently used method for MC design. However, and because of cellular formation problem (CFP) is a NP-complete problem, there is still the challenge of creating an efficient clustering method. This paper deals with the use of a discrete particle swarm optimization algorithm in clustering problems. The remaining of the paper is organized as follows. In Section II, we discuss related works concerning machine grouping and particle swarm optimization techniques. In Section III, the statement of the problem is presented. The proposed algorithm is presented in section IV. Experimental results are presented in Section V. Finally, we give some conclusions and lines for future research.

II. CLUSTERING PROBLEM AND MACHINE GROUPING

The attempts to solve clustering or machine grouping problems can be classified in two groups. The first group consists of algorithms that attempt to determine the optimal solution, and the second one, the metaheuristics or approximate methods. In the last years different metaheuristic methods have been used to solve the cell formation problem, Boctor [2] and Chen and Srivastava [3] used Simulated Annealing. Venugopal and Narendran [4] presented the first attempt to solve a cell-formation problem with the help of an evolutionary computation algorithm. Gupta et al. [5] employed the same genetic representation of solutions as Venugopal and Narendran [4] but followed a different multiobjective optimization approach for the simultaneous minimization of the total number of intercell-intracell moves and within-cell load variation. Aljaber et al. [6] and Lozano et al. [7] used Tabu Search. More recently the paper of Andres and Lozano [8] presents the first particle swarm optimization (PSO) algorithm designed to address this problem.

PSO is an evolutionary computation (EC) method inspired by flocking birds (Eberhart and Kennedy [9]) and has been applied to many different areas including manufacturing. PSO is initialised with a population of random solutions and this initial population evolves over generations to find optima. However, in PSO, each particle in population has a velocity, which enables them to fly through the problem space instead of dying and mutations. Therefore, each particle is represented by a position and a velocity. The modification of the position of a particle is performed by using its previous position information and its current velocity. Each particle knows its best position (personal best) so far and the best position achieved in the group (group best) among all personal bests. These principles can be formulated as:

$$v_i^{n+1} = wv_i^n + c_1 r_1^n (p_i^n - x_i^n) + c_2 r_2^n (p_g^n - x_i^n) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{n+1} \quad (2)$$

where w is inertia weight; c1, c2 are two positive constants, called cognitive and social parameter respectively; d=1, 2, …, D; i=1, 2, …, m, and m is the size of the swarm; r1, r2 are random numbers, uniformly distributed in [0,1]; and n=1, 2, …, N, denotes the iteration number, N is the maximum allowable iteration number. The first term on the right hand side of Eq. (1) is the previous velocity of the particle, which enables it to fly in search space. The second and third terms are used to change the velocity of the agent according to pbest and gbest. Generally speaking, the set of rules that govern PSO are: evaluate, compare and imitate. The evaluation phase measures how well each particle (candidate solution) solves the problem at hand. The comparison phase identifies the best particles. The imitation phase produces new particles based on some of the best particles previously found. These three phases are repeated until a given stopping criterion is met. The objective is to find the particle that best solves the target problem.

III. PROBLEM STATEMENT

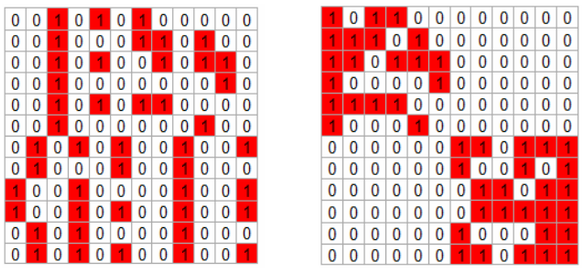
In this work, clustering problem is considered as an optimization one. So, let us introduce the elements of this optimisation model.

Given an incidence matrix A = [a_{ij}], where:

a_{ij} = 1 if j-th component visits i-th machine
a_{ij} = 0 otherwise.

E-FORENSICS 2008, January 21-23, Adelaide, Australia

DOI 10.4108/wkdd.2008.2655



(a) (b)

Figure 1. Initial incidence matrix (a) and a rearranged matrix (b)

When a machine-component incidence matrix is constructed, no machine groups or part families are easily visible. As it was commented before, the main objective in machine grouping is the formation of set of machines and workpieces in groups so that the number of intercell transportations of pieces is minimized. Therefore, the initial matrix (figure 1(a)) has to be transformed into a matrix that has a block diagonal structure (figure 1 (b)). This rearrangement aims at minimization of total intercell moves and minimization of within-cell load variation. A rigorous mathematical formulation of machine-component grouping problem with these objectives is given in [2].

The optimisation model is stated as follows. Let M be the number of machines, P the number of parts, C the number of cells, i the index of machines (i = 1,….,M), j the index of parts (j =1,…..,P), k the index of cells (k=1,…..,C), A = [a_{ij}] the M x P binary machine–part incidence matrix, Mmax the maximum number of machines per cell; We selected as the objective function to be minimized the number of times that a given part requires to be processed by a machine that does not belong to cell that the part has been assigned to.

$$y_{ik} = \begin{cases} 1 & \text{if machine i} \in \text{cell k} \\ 0 & \text{otherwise} \end{cases}$$

$$z_{jk} = \begin{cases} 1 & \text{if part j} \in \text{family k} \\ 0 & \text{otherwise} \end{cases}$$

The problem is represented by the following mathematical model:

Minimize

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

Subject to

$$\sum_{k=1}^{C} y_{ik} = 1 \qquad \forall i,$$

$$\sum_{k=1}^{C} z_{jik} = 1 \qquad \forall j,$$

$$\sum_{i=1}^{M} y_{ik} \le M_{\max} \qquad \forall k$$

IV. Proposed Swarm-Based Clustering Method

A number of adaptations of traditional PSO algorithm have been implemented in order to develop a particle swarm optimisation algorithm in machine grouping. First, traditional PSO algorithm was developed for continuous domains. A discrete binary version of the PSO algorithm was developed by [9]. Correa et al. [10] proposed a Discrete PSO (DPSO) algorithm for attribute selection in data mining applications. The algorithm proposed here is based on the DPSO and the concept of proportional likelihoods. The main difference between traditional PSO algorithm and the algorithm proposed here is that the proposed algorithm does not use a vector of velocities as the standard PSO algorithm does. It works with a mechanism inspired in the proportional likelihoods concept. According Correa et al. [10], the notion of proportional likelihood used in the DPSO algorithm and the notion of velocity used in the standard PSO are somewhat similar.

This algorithm deals with discrete variables (cells) and its population of candidate solutions contains particles of a given size (n=number of machines). Each component of the particle (vector) takes a value between 1 and k and represents the cell to which the machine is assigned. Potential set of solutions to the optimisation problem at hand are represented by a swarm of particles. There are N particles in a swarm. X(i) keeps a record of the best position it has ever attained. This information is stored in a separated particle labelled as B(i). The swarm also keeps a record of the global best position ever attained by any particle in the swarm. This information is also stored in a separated particle labelled G.

The initial population of particles is generated by producing a series of integer random numbers. These numbers are uniformly generated between 1 and n inclusive. Potential solutions (particles) to the target problem are encoded as fixed length discrete strings; i.e., X(i) = (x(i;1); x(i;2);…; x(i;n)), where $x_{i,j} \in \{1,\ldots,k\}$, i =1, 2,..., N and j = 1, 2, ..., n. For example, given the number of possible cells k=3 and N= 4, a swarm could look like this:

X (1) = (1; 2; 1; 2; 3; 3)
X (2) = (1; 1; 2; 1; 2; 3)
X (3) = (1; 2; 3; 2; 1; 3)
X (4) = (1; 2; 2; 3; 1; 3)

In this example, particle X (1) = (1; 2; 1; 2; 3; 3) represents a candidate solution where machines 1 and 3 are located to cell number 1; machine 2 and 4 are located in cell 2 and machine 5 and 6 are assigned to cell number 3. After the initial population of particles is generated, the process of calculation of the fitness function is performed.

As it was commented previously, the algorithm proposed in this paper is based on the notion of proportional likelihoods and is an adaptation of the concept of velocity proposed by Correa et all.[10]. The updating process is based on X(i), B(i) and G and works as follows. In the context of combinatorial optimisation the velocity of a particle must be understood as an ordered set of transformations that operates on a solution. The transformation of a solution is represented with a term that represents the difference between two positions. Hence, in each case (X(i)–B(i)) and (X(i)-G) they represent the necessary movements to change from the position given by the first term, subtracting term to the position given by the second, subtracted term.

For example, consider the following instances of X(i) and B(i)

X (i) = (1; 3; 3; 2; 1;3)
B (i) = (1; 1; 3; 3; 2;3)

The difference between X(i) and B(i) represents the changes that will be needed to move the particle i from X to B. The number μ represents the number of elements different of 0 in the subtraction of B in X. If the difference between a given element of X(i) and B(i) is not null, it means that there is a possibility of change of that position. If the difference is other than 0, that position is susceptible of change through the operations that are described as follows.

A new vector P, is generated that register the positions of the elements equal to zero. A random number is generated and assigned to β. This number β corresponds to the number of changes that will be made to X(i) based on the difference between X(i) and B(i), therefore β is in the interval (0, μ). Then, a set ψ of β binary number is randomly generated. If the binary number is 1 the change is made, in other hand, if the

number is 0, the change is not performed. A similar process is performed to update the particle position in accordance to the best global position (G). In case that several of the applied movements involve the same position (machine), the change caused by the global best position, the second operation in our algorithm, has the priority. For instance, if:

X(i) – B(i) = (1-1 ; 3-1;3-3; 2-3; 1-2; 3-3) = (0, 2,0,-1,-1,0)

The new vector P(i) is generated

P(i) = (2,4,5)

Thus $\mu = 3$.

Suppose that $\beta=2$ and $\psi = (0,1,1)$

That means that positions 4 and 5 will be replaced in X(i) by the elements (in the same positions) of B(i).

X`(i) = (1 3 3 3 2 3)

The process is repeated with the new position X'(i) and G obtaining the new position of X(i+1).

V. NUMERICAL ILLUSTRATION

The purpose of this section is to show, through a numerical example how the proposed formulation can be used in design a cellular manufacturing system. To test the performance of the proposed model it was generated randomly a set of 5 problems with 12 parts and it was assumed that the parts require at most 6 machines and at least 3 machines selected from 12 machines. For selecting the algorithm's parameters a set of nine experiments were carried out. The swarm size was set either to 10, 30 or 60 particles. Additionally, the number of iterations was set to 30, 60 and 80 iterations. For each experiment, the minimum value, the maximum value and the medium value, along with the standard deviation of the obtained values in each test, were computed. Also, the percentage of times that the best value was attained by the algorithm is included. With these three measures, it is possible to visualize the best configuration in terms of stability, variability and repeatability of the algorithm as a function of its parameters. These performance measures are intended to track the impact of the swarm size and the number of iterations in the success of the optimisation process. Figure 2 shows the relation between the percentages of times that the best value was attained by the algorithm. Figure 3 shows the standard deviation of the results obtained by the experiments. Figure 4 shows the best values obtained by each one of the performed experiments. Table I shows the number that identifies each one of the experiments carried out.

TABLE I. NUMBER OF THE 9 EXPERIMENTS PERFORMED

Exp.Number	Swarm Size	Iterations
1	60	30
2	60	60
3	60	80
4	30	30
5	30	60
6	30	80
7	10	30
8	10	60
9	10	80

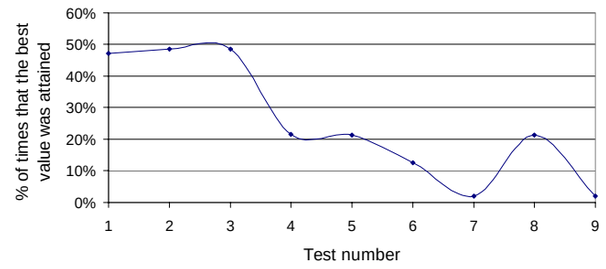


Figure 2. % of times that the best value is attained in each test.

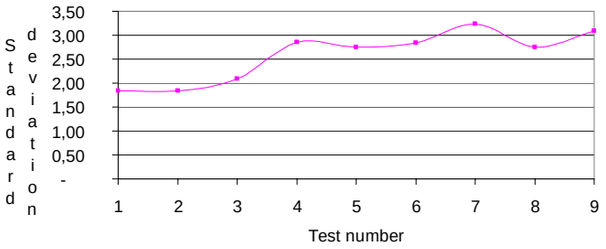


Figure 3. Standard deviation of the results in each experiment

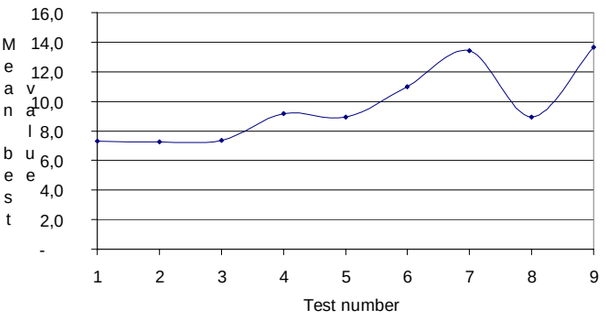
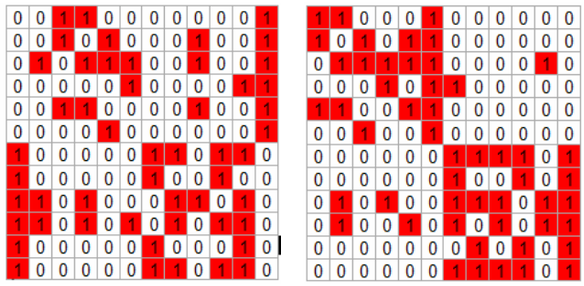


Figure 4. Mean best value attained in each experiment

Considering the figures 2,3 and 4, it can be observed that the swarm size of 60 individuals presented significant best results. No significant differences were detected between the different number of iterations of the algorithm.

After comparisons tests, the swarm size was set to 60 individuals. The number of iteration was set to 80. Each experiment was executed and repeated 100 times and the results were compared. Table 1 shows the mean value, absolute deviation, minimum value and maximum value. Figure 5(b) shows the result of applying the proposed PSO algorithm to the example of figure 5(a). The rearranged matrix minimizes the number of exceptional elements with a 2 cells partition.

From the table 1 it can be observed that the algorithm is satisfactory and consistent in founding the best solution. In all the generated cases, the algorithm found the best solution in almost 50% of the experiments. The variability is considered to be within acceptable and normal limits, which guarantees that the algorithm may be considered as a reliable technique in forming machine cells. However further development are still necessary to increase the performance, even in situations with a higher degree of complexity.



(a) (b)

Figure 5. Example of utilization of the proposed PSO algorithm

TABLE II. RESULTS OF THE 5 EXPERIMENTS PERFORMED

Instance	Mean	Max	Min	St. Dev	% of min
1	7,34	18	6	2,1	49
2	7,77	16	6	2,4	39
3	7,02	12	6	1,6	57
4	7,46	16	6	2,3	47
5	7,29	15	6	2,1	51

CONCLUSIONS

In this paper, a new approach based on Particle Swarm Optimisation algorithm has been proposed to clustering problem. A novel discrete PSO algorithm is proposed and applied for the computation of the velocity vector as in the traditional PSO algorithm. The proposed method works with a new notion of the velocity component. The algorithm is based on an approach called the Proportional Likelihoods which is used in data mining problems. The algorithm and its theoretical concepts are explained and illustrated for a cell formation problem with five instances of initial incidence matrix, 12 machines and 12 components. A set of experiments was performed that show that the algorithm is stable and presents low variability. The preliminary results obtained in this research are auspicious. In future steps the proposed algorithm will be applied to a variety of different test problems. In addition, parameter optimisation and hybrid approaches are also topics for future research.

REFERENCES

[1] 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.

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

[3] Chen W-H, Srivastava B.: Simulated annealing procedures for forming machine cells in group technology. European Journal of Operational Research 75:100-11 (1994).

[4] Venugopal, V., T.T. Narendran.: 'A genetic algorithm approach to the machine component grouping problem with multiple objectives. Computers & Industrial Engineering 22.4: 469-480 (1992).

[5] Gupta, Y., Gupta, M., A. Kumar, Sundaram,C.: A genetic algorithm-based approach to cell composition and layout design problems, International Journal of Production Research, vol.34, no.2, pp.447-482, (1996).

[6] Aljaber N., Baek , W., Chen, C-L.: A tabu search approach to the cell formation problem Computers and Industrial Engineering 32 , 1, 169 - 185 (1997).

[7] Lozano, S., Adenso B., Salinas,I., Giménez L.: A One-Step Tabu Search Algorithm for Manufacturing Cell Design. Journal of the Operational Research Society . Vol. 50. Núm. 5. 509-516 (1999).

[8] Andres, C, Lozano, S.:A particle swarm optimization algorithm for part-machine grouping ROBOT CIM-INT MANUF 22 (5-6): 468-474 OCT-DEC (2006)

[9] Kennedy, J. & Eberhart, R. C., 1995, “Particle Swarm Optimization.” In Proceeding of the IEEE International Conference on Neural Networks, Perth, Australia, IEEE Service Center, 12-13.

[10] Correa, E.S., Freitas, A., Johnson, C.G.: . A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set. In M. K. et al., editor, Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2006, pages 35–42, Seattle, WA, USA, July 2006. ACM Press.