

PERFORMANCE COMPARISON BETWEEN A CLASSIC PARTICLE SWARM OPTIMIZATION AND A GENETIC ALGORITHM IN MANUFACTURING CELL DESIGN

COMPARACIÓN DEL DESEMPEÑO ENTRE UN ALGORITMO CLÁSICO DE OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS Y UN ALGORITMO GENÉTICO EN EL DISEÑO DE CELDAS DE MANUFACTURA

JOHANNA RODRÍGUEZ LEÓN

Ingeniera Industrial Escuela de Estudios Industriales y Empresariales Universidad Industrial de Santander, Colombia. johannita7@hotmail.com

JABID EDUARDO QUIROGA MÉNDEZ

Magister en Ingeniería Mecánica, Profesor Escuela de Ingeniería Mecánica Universidad Industrial de Santander, Colombia, jabib@uis.edu.co

NESTOR RAUL ORTIZ PIMIENTO

Magister en Ingeniería, Profesor Escuela de Estudios Industriales y Empresariales Universidad Industrial de Santander, Colombia, nortiz@uis.edu.co

Received for review March 13th, 2012, accepted December 14th, 2012, final version January, 18th, 2013

ABSTRACT: This article studies the performance of two metaheuristics, the Particle Swarm Optimization (PSO) and the Genetic Algorithm (GA), in the manufacturing cell formation problem of a factory that needs to organize three production cases in an efficient way for four, five and six manufacturing cells to produce 30, 40 and 50 different products to be processed in 10, 10 and 20 type machines, respectively. The procedure for adjusting the particular parameters of each algorithm is implemented through a Design of Experiments which includes their own analysis of variance. Both algorithms are implemented in Matlab[®]. The results obtained by each meta heuristic are compared in terms of the cost of the best solution found and the execution time used to find that solution, so that it is possible to establish which methodology is the most appropriate when solving this optimization problem.

KEYWORDS: Manufacturing cells, Group Technology, Cellular Manufacturing, Meta-heuristic Models, Particle Swarm Optimization, Genetic Algorithm, Intercellular Transfers.

RESUMEN: Este artículo estudia el desempeño de los meta-heurísticos Optimización de Enjambre de Partículas y Algoritmos Genéticos en el problema de formación de celdas de manufactura de una empresa que desea organizar de manera eficiente tres escenarios de producción: cuatro, cinco y seis celdas de manufactura para la fabricación de 30, 40 y 50 productos diferentes a ser procesados en 10, 10 y 20 tipos de máquinas, respectivamente. El proceso de ajuste de los parámetros particulares de cada algoritmo se realiza a través de un diseño de experimentos con su respectivo análisis de varianza. Los algoritmos son implementados en Matlab[®]. Los resultados obtenidos por cada metaheurística son comparados en términos del costo de la mejor solución encontrada y del tiempo de ejecución empleado para llegar a dicha solución, de manera que sea posible establecer cual metodología es la más adecuada a la hora de solucionar este problema de optimización.

PALABRAS CLAVE: Celdas de Fabricación, Tecnología de Grupos, Manufactura Celular, Metodologías Meta-heurísticas, Optimización Enjambre de partículas, Algoritmos Genéticos, Transferencias Intercelulares.

1. INTRODUCTION

Nowadays, production systems must be flexible to respond to market needs in the elaboration of new products or to satisfy the customer requirements. The rapid obsolescence of the products and the sudden and unexpected fluctuations of the demand, has led to new concepts of production systems such as Cellular

Manufacturing (CM). CM is based upon the principles of Group Technology, which seeks to group products to be manufactured in similar characteristics (size, shape, or common processing). In CM systems, machines are grouped together according to families of parts produced. Each production cell must be able to produce any member of the family. The organization of a plant using this structure becomes a challenge because of the

numerous variations in the machines grouping and in the creation of products families.

In this study, a cell formation problem is addressed using PSO and GA comparing each methodology in terms of algorithm efficiency and feasible solution. The objective function involves the product demand, the operating time, the capacity by type of machine and constraints on the cell sizes to be arranged.

The paper is organized as follows: in section 2 a relevant literature review is presented, section 3 introduces briefly the theoretical framework describing each methodology and the mathematical model is formulated. In section 4 a statistical analysis is presented, which was performed to establish the PSO and GA parameters. Then, the specific characteristic of the three problems used in this study are introduced in section 5. Section 6 summarizes the results obtained in each of the proposed approaches. Finally, conclusions drawn from this study are given in section 7.

2. RELEVANT LITERATURE REVIEW

The cell formation problem is an NP-hard combinatorial optimization problem for which several exact and approximate solution methods have been proposed [1].

Different methodologies for solving the cell forming problem based on mathematical models have been proposed since the 1980s, Kusiak [2] and Shtub [3]. The complexity associated with the model based approach has motivated the use of metaheuristics as an alternative solution methodology.

In the literature, the PSO implementation for solving the cell formation problem is limited. Andrés and Lozano [4] PSO is only used to find the optimum number of transfers of lots of product between cells. Neither the product manufacturing time, nor cell size related constraints are considered. Ming and Ponnambalama [5] proposed a hybrid concept between PSO and GA for minimizing the total cell load variations and the total component traffic. A discrete PSO algorithm is proposed for minimizing the intercell transfers in [6], however, neither production cost nor production time are considered. Mehdizadeh and Tavakkoli [7] proposed an algorithm based on Fuzzy clustering and Particle Swarm Optimization (FPSO) to solve the cell formation problem. Anvari, Mehrabad and

Banzinpour [8] considered the cell formation problem using a hybrid PSO-GA. A new mutation operator is introduced to update the velocity equation to minimize the possibility that the search gets trapped in local minimums.

The GA implementation for solving the cell formation problem began with Venugopal and Narendran [9]. Gupta, Kumar and Sundram [10] proposing a GA to minimize the intracell and intercell movements. In [11] an algorithm with new genetic operators and a new chromosome representation is proposed. In [12] an integer programming model employing GA is proposed. Morad and Zalzal [13] used GA to handle two problems in manufacturing systems: the formation of manufacturing cells in cellular manufacturing and batch scheduling. Dimopoulos and Zalzal [14] examined a CM optimization problem to configure the cells in a facility maximizing the total number of batches processed per year.

Wu, Chu, Wang and Yang [15] presented a hierarchical genetic algorithm (GA) to solve the cell formation and layout decisions of cellular manufacturing considering two highly correlated fitness functions, and proposing a group mutation operator to increase the probability of mutation. In [16] a mathematical model of a nonlinear mixed-integer programming type is presented for designing cellular manufacturing systems. This paper develops and uses genetic algorithms (GAs), simulated annealing (SA) and tabu search (TS) for a Cellular Manufacturing CM model in a dynamic environment setting. In [17] the operational time and the sequence of operations are considered to minimize the total cell load variation.

Tunnukij and Hicks [18] presented a GA variation for which the number of manufacturing cells and the number of parts or machines per cell are not preset. In [19] GA are used to perform a multiobjective optimization. Cells are formed so as to simultaneously minimize three conflicting objectives, namely, the level of the work-in-process, the intercell moves and the total machinery investment. In [20] the product mix, product demand in each period, machine relocation and new equipment is considered in the Cell Formation (CF) problem.

Caprihan, Slomp, Gusaran and Agarwal [21] developed a Quantum PSO (QPSO) procedure to design virtual manufacturing cells for which machines and jobs are assigned to the cells seeking the maximization of

productive output, while simultaneously minimizing the inter-cell movements. This problem is also solved using GA, the results showed that the QPSO implementation outperformed the GA algorithm in running time and results.

Sarayloo and Tavakkoli [22] developed an Imperialistic Competitive Algorithm (ICA), which optimizes inspired by imperialistic competition. ICA is compared with other well-known evolutionary algorithms, i.e. genetic algorithm (GA) and particle swarm optimization (PSO), to show its efficiency. However, in this paper the performance of PSO to GA is not compared.

In [23-24] a comparison between PSO and GA is studied in aerospace and control applications. Similarly, for the CF problem the comparison among different strategies should be compared to the solution obtained by an exact optimization method.

3. THEORETICAL FRAMEWORK

3.1. The PSO Algorithm

PSO is a metaheuristic that optimizes a function by having a population of candidate solutions, and iteratively trying to improve a candidate solution with regard to a *fitness* function. PSO explores the search-space using the position and velocity of each particle [25]. Each particle's movement is influenced by its local best known position and is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions [26].

In general, each particle p_i is composed of five main elements: a) the current particle position $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$, b) the current best position $pBest_i = (p_{i1}, p_{i2}, \dots, p_{in})$, c) the velocity vector $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$; d) the best solution *fitness* $_x_i$ for the current x_i ; e) the current best solution *fitness* $_pBest_i$. The algorithm begins by randomly choosing initial positions and velocities for each particle. Then the *fitness* $_x_i$ and *fitness* $_pBest_i$ are determined. The iterative process to determine the solution is performed in (1) and (2).

$$v_i^{k+1} = w * v_i^k + c_1 * rand_1 * (pBest_i - x_i^k) + c_2 * rand_2 * (p_{gi} - x_i^k) \tag{1}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{2}$$

where v_i^k is the velocity for the i^{th} particle in the k^{th} iteration; w is the inertia factor; c_1 and c_2 are acceleration constants (cognitive and social); $rand_1$ and $rand_2$ are random real numbers sampled from a uniform distribution between 0 and 1; x_i^k is the current position of the i^{th} particle in the k^{th} iteration; $pBest_i$ is the best position (solution) for the i^{th} particle and p_{gi} represents the particle position for the best *fitness* $_pBest$ of the outline of p_i (*lBest* meaning *localbest*) or the entire swarm (*gBest* meaning *globalbest*).

The equation (1) represents the updated velocity vector for the i^{th} particle in the k^{th} iteration. The cognitive component is modeled by the factor ' $c_1 * rand_1 * (pBest_i - x_i^k)$ ' and represents the distance between the current position and the best known position of this particle, that is, the decision assumed by the own experience in its life. The social component is considered by ' $c_2 * rand_2 * (p_{gi} - x_i^k)$ ' and represents the distance between the current position and the best position in the swarm. Equation (2) represents the movement for the i^{th} particle in the k^{th} iteration.

3.2. The Genetic Algorithm

A GA is a search heuristic that mimics the process of natural evolution. In a GA, a population of strings (called chromosomes), which encode candidate solutions (called individuals) to an optimization problem, which evolves toward better solutions using genetic operators.

A GA consists of the following operations: a) Selection. During each successive generation, a proportion of the existing population is selected to breed a new generation keeping the best sequences of genetic material. b) Reproduction. The next step is to generate a second generation of the population of solutions from those selected in the first operation by using genetic operators: crossover, and/or mutation. For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected to create each new child, and the process continues until a new population of solutions of appropriate size is

generated [27]. The mutation provides a variation of the information contained in the chromosomes, which can lead the search space exploration to new environments avoiding stagnation or the appearance of degenerate populations. The *fitness* function measures the quality of the solution and determines if the genetic material will be transmitted to the subsequent generations.

3.3. The Cell Formation (CF) Problem

The CF problem is expressed as the minimization of the production costs. The mathematical model utilized in this work is based on [28]. In this process the following costs are considered:

Operating cost: The cost of operating machines for the production of parts. This cost depends on the cost of operating each machine type per hour and the number of hours required for each machine type.

Intercell material handling cost: The cost of transferring parts between cells when the parts cannot be produced completely in a single cell. This cost is determined by multiplying the number of batches of each product to be transferred by the cost of transporting a batch of product between any pair of cells

The following decisions must be made during the design process: The assignment of operations of each product to the cells and the determination the number of machines to perform the assigned task. The following considerations must be imposed in the model:

1. There must be sufficient machine capacity to produce each product, to satisfy the specified level of demand in each period.
2. The cell size must be specified. However upper and lower bounds can be used instead of a specific number.
3. The number of cells in the system must be specified.

The objective function based on the previous considerations is established in the following equation:

Minimize

$$\text{COST} = \sum_{c=1}^C \sum_{m=1}^M (AMC_{mc} * CM_m) +$$

$$CTI * \left[\sum_{p=1}^P Dp * \left(\sum_{j=1}^{Op-1} \sum_{c=1}^C |MA_{(j+1)pc} - MA_{jpc}| \right) \right] \quad (3)$$

Subject to:

$$\sum_{c=1}^C MA_{jp} = 1, \text{ for all } j, p \quad (4)$$

$$\sum_{p=1}^P \sum_{j=1}^{Op} (Dp * Top_{jp} * M_{jp} * MA_{jp}) \leq CAP_m * AMC_{mc}, \text{ for all } m, c \quad (5)$$

$$\sum_{m=1}^M AMC_{mc} \geq TMin, \text{ for all } c \quad (6)$$

$$\sum_{m=1}^M AMC_{mc} \leq TMax, \text{ for all } c \quad (7)$$

$$AMC_{mc} \geq 0 \text{ (Integer), for all } m, c \quad (8)$$

$$MA_{jpc} \geq 0 \text{ (Binary), for all } j, p, c \quad (9)$$

Where c is the cell index ($c=1,2,\dots,C$); m is the index for the different types of machines ($m=1,2,\dots,M$); p is the index for the different types of product ($p=1,2,\dots,P$); j is the index for the different types of operations required by part ($j=1,\dots,Op$); D_p is the demand for product p in the period; O_p is the number of operations required to manufacture a part or product p ; M_{jp} is the incidence matrix part-machine. In this matrix M represents the type of machine required for each product; Top_{jp} is the time required to perform operation j for part type p ; CAP_m is the capacity of each machine of type m in each period; CM_m is the operating cost per period of machine type m ; CTI is the intercell material handling cost per batch; $TMin$ is the lower bound cell size; $TMax$ upper bound cell size; AMC_{mc} is an $M \times C$ matrix that contains the number of machines of type m to be assigned to the cell c in each period and MA_{jpc} is the matrix of size $Op \times P$ with: 1 if the operation j of part type p is assigned to the cell c ; 0 otherwise.

In equation (3) the first term is the operating machine total costs, the second term is the total intercell material handling cost, the equation (4) limits the assignation of each product operation to the respective cell; the

equation (5) limits machine and cells capacity for satisfying the demand and the equations (6-7) limits the number of machines for the different cells.

4. STATISTICAL ANALYSIS

A statistical analysis was performed to establish the PSO and GA parameters. Then, the set of instances to be included in the computational experiments was determined. Specific configuration of each problem included four, five and six cells for producing 30, 40 and 50 types of products using 10, 10, and 20 types of machines respectively.

In the case of PSO, an experimental design was used to determine the values of c_1 , c_2 , w , and the size of the population. The setting of these parameters will be performed through a completely randomized design. Table 1 provides the experimental design, indicating the established values for each parameter to be adjusted.

Initially, the algorithm was run for four values of each factor, moving one at a time. The experiment was repeated five times and the average values were taken. An Analysis of Variance was performed to these data in order to study the influence of the chosen parameters in the performance of the algorithm. In table 2 are shown the results of the ANOVA.

Table 1. Experimental values used in the Analysis of Variance of PSO parameters

FACTOR (PARAMETER)	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
Acceleration Coefficient C1	2	4	6	8
Acceleration Coefficient C2	2	4	6	8
Inertial Weight W	1.0	0.9	0.8	0.7
Population	300	500	800	1200

Table 2. Results of PSO Analysis of Variance

PARAMETER	Ho	VALUE
Acceleration Coefficient C1	Non-acceptance	4
Acceleration Coefficient C2	Acceptance	2
Inertia Weight W	Acceptance	0.9
Population	Non-acceptance	1200

In the case of the GA, a factorial experiment for adjusting the GA parameters was run [28]. Table 3 provides the selected factors with their respective levels.

Table 3. Experimental values used in the Analysis of Variance of GA parameters

FACTOR (PARAMETER)	HIGH	LOW
Crossover	Multipoint	One point
Crossover scheme	Alternative	Conventional
Mutation	Gen to gen	One gen
Crossover probability	0.98	0.90
Mutation probability	0.05	0.01
Population size	500	100
Number of generations	500	100

Table 4 presented the results of the ANOVA analysis for the GA parameters [28].

Table 4. Results of AG parameter setting

PARAMETER	LEVEL	VALUE
Type of crossover operator	-1	Simple
Type of crossover scheme	1	Alternative
Type of mutation operator	1	Gen to gen
Crossover probability	-1	0.90
Mutation probability	-1	0.01
Population size	1	500
Number of generations	1	500

5. PROBLEM CONFIGURATION

The performance of the PSO and the GA implemented was evaluated using three configuration problems. The input data for each problem were generated randomly, under the following conditions:

The demand of each product was generated using a discrete uniform distribution between 10 and 25 lots; the production sequence of each product was

generated randomly using the probability distribution shown in table 5; the specific machine employed in each operation was selected using a discrete uniform distribution between 1 and M; the production time for a lot of any product per operation was sampled from a discrete uniform distribution between 1 and 10 minutes. The machine available time was 8 hours, 5 days per week during a three months programming period.

The Intercell material handling cost for each lot was assumed to be one monetary unit; the Operating cost was generated randomly using a discrete uniform distribution between 100 and 2000 monetary units.

Table 5. Probability distribution for the production sequence

NUMBER OF OPERATIONS	PROBABILITY
2	30%
3	30%
4	20%
5	10%
6	10%

The particular configuration for the three problems studied using the two metaheuristics is summarized in table 6.

Table 6. Studied problems configuration

ATTRIBUTE	P1	P2	P3
Products	30	40	50
Type of machines	10	10	20
Number of cells	4	5	6
Minimum Size Cell	2	2	2
Maximum Size Cell	10	10	20

6. RESULTS

Table 7 show the best solutions found using PSO after 2000 iterations, GA after 500 iterations, and the solutions found by CPLEX software after 2 hours.

Table 7. Best solutions in terms of monetary cost

PROBLEM	COST		
	PSO	GA	CPLEX
1	21581	21544	21349
2	36115	35917	33678
3	46689	48025	46084

Table 8 present the running time analysis for the metaheuristics studied, in this table can be observed a noticeable difference in running time for the three problems studied.

Table 8. Best solution in terms of running time in seconds (s)

PROBLEM	TIME		
	PSO	GA	CPLEX
1	1626	728.1	7200
2	2186.7	679.3	7200
3	5781.2	1400.8	7200

In the three configuration problems the AG is far superior in efficiency to PSO.

7. CONCLUSIONS

In this paper a Particle Swarm Optimization (PSO) and a Genetic Algorithm (GA) were evaluated using a Cell Formation problem. The experimental results showed that there is a slight difference in the performance to find the minimum cost. In two of the three scenarios GA reported a better performance, although the algorithms present variations in response, in all cases studied the differences are below 3%. In general, the solution provided by each algorithm is close to the solution obtained by the CPLEX software (analytical solution).

The sensitivity analysis performed on the PSO parameters show a great influence of the size of the population. Additionally, an increase in the number of cells produces a better performance of the PSO algorithm.

As far as computational time is concerned, the PSO approach took longer to converge in all the cases studied. The difference was greater than 200%.

REFERENCES

[1] Papaioannou, G. and Wilson, J. M., The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research, *European Journal of Operational Research*, 206, pp. 509–521, 2010.

[2] Kusiak, A., The generalized group technology concept, *International Journal of Production Research*, 25, pp. 561–569, 1987.

- [3] Shtub, A., Modelling group technology cell formation as a generalized assignment problem, *International Journal of Production Research*, 27, pp. 775–782, 1987.
- [4] Andres, C. and Lozano, S., A particle swarm optimization algorithm for part machine grouping, *Robotics and Computer-Integrated Manufacturing*, 22, pp. 468–474, 2006.
- [5] Ming, L. C. and Ponnambalama, S. G., A hybrid GA/PSO for the concurrent design of cellular manufacturing system. *IEEE International Conference on Systems, Man and Cybernetics*, Singapore, 1855-1860, pp. 12-15 October 2008.
- [6] Duran, O., Rodriguez, N. and Consalter, L. A., A PSO-Based Clustering Algorithm for Manufacturing Cell Design, *First International Workshop on Knowledge Discovery and Data Mining*, Adelaide, 72 – 75, pp. 23-24 January 2008.
- [7] Mehdizadeh, E. and Tavakkoli-Moghaddam, R., A fuzzy particle swarm optimization algorithm for a cell formation problem. *Proceedings of IFSA-EUSFLAT*, Available: http://www.eusflat.org/proceedings/IFSA-EUSFLAT_2009/pdf/tema_1768.pdf [cited 02-12-2011].
- [8] Anvari, M., Mehrabad, M. S. and Banzinpour, F., Machine-part cell formation using a hybrid particle swarm optimization, *International Journal of Advanced Manufacturing Technology*, 47, 745-754, 2008.
- [9] Venugopal, V., Narendran, T.T., A genetic algorithm approach to the machinecomponent grouping problem with multiple objectives, *Computers and Industrial Engineering*, 22 (4), pp. 469–480, 1992.
- [10] Gupta, Y., Gupta, M., Kumar, A. and Sundram, CH., Minimizing total intercell and intracell moves in cellular manufacturing: a genetic algorithm approach. *International Journal of Computer Integrated Manufacturing*, 8 (2), pp. 92-101, 1995.
- [11] Falkenauer, E., A New Representation and Operators for Genetic Algorithms Applied to Grouping Problems. *Evolutionary computation*, 2 (2), pp. 123-144, 1994.
- [12] Joines, J., Culbreth, T. and King, R., Manufacturing Cell Design: An Integer Programming Model Employing Genetic Algorithms, *IIE Transactions*, 28 (1), pp. 69-85, 1996.
- [13] Morad, N. and Zalzal, A., Formulations for cellular manufacturing and batch scheduling using genetic algorithms, *UKACC International Conference on Control*, pp. 473-478, 2-5 September 1996.
- [14] Dimopoulos, C. and Zalzal, A., Optimization of Cell Configuration and Comparisons using Evolutionary Computation Approaches, *IEEE World Congress on Computational Intelligence*, Anchorage, 148-153, pp. 4-9 May 1998.
- [15] Wu, X., Chu, C., Wang, Y. and Yang, W., A Genetic Algorithm for Integrated Cell Formation and Layout Decisions, *Congress on Evolutionary Computation*, Honolulu, pp. 1866-1872, 12-17 May 2002.
- [16] Pervaiz, A., Tavakkoli-Moghaddam, R. and Safaei, N., A Comparison of Heuristic Methods for Solving a Cellular Manufacturing Model in a Dynamic Environment, Available: http://www.wlv.ac.uk/pdf/uwbs_04%20wp007-04%20ahmed%20et%20al.pdf, [cited 02-12-2011].
- [17] Mahapatra, S. and Sudhakara, P., Genetic cell formation using ratio level data in cellular manufacturing systems, *International Journal of Advanced Manufacturing Technology*, 38, pp. 630-640, 2008.
- [18] Tunnukij, T., and Hicks, C., Cell formation in group technology: a combinatorial search approach, *International Journal of Production Research*, 35, 2025-2043, 2009.
- [19] Neto, A. and Filho, E., A simulation-based evolutionary multiobjective approach to manufacturing cell formation. *Computers & Industrial Engineering*, 59, pp. 64-74, 2010.
- [20] Deljoo, V., AL-E-Hashem, S. J., Deljoo, F. and Aryanezhad, M. B., Using genetic algorithm to solve dynamic cell formation problem, *Applied Mathematical Modelling*, 34, pp. 1078-1092, 2010.
- [21] Caprihan, R., Slomp, J., Gusaran. and Agarwal, K., A quantum particle swarm optimization approach for the design of virtual manufacturing cells, *IEEE International Conference on Industrial Engineering and Engineering Management*, Hong Kong, 125 – 129, pp. 8-11 December 2009.
- [22] Fatemeh, S. and Tavakkoli-Moghaddam, R., Imperialistic Competitive Algorithm for Solving a Dynamic Cell Formation Problem with Production Planning. *Advanced Intelligent Computing Theories and Applications. Lecture Notes in Computer Science*. 6215, pp. 266-276, 2010.
- [23] Hassan, R., Cohanin, B. and De Weck, O., A Comparison of Particle Swarm Optimization and The Genetic Algorithm, Available: http://web.mit.edu/deweck/www/PDF_archive/3%20Refereed%20Conference/3_50_AIAA-2005-1897.pdf [cited 02-12-2011].

[24] Panda, S. and Padhy, N., Comparison of Particle Swarm Optimization and Genetic Algorithm for TCSC-based Controller Design, *International Journal of Electrical and Electronics Engineering*, 1 (1), pp. 41-49, 2007.

[25] Correa, R., Begambre, O. y Carrillo, J., Validación de un Algoritmo Híbrido del PSO con el Método Simplex y de Topología de Evolución Paramétrica, *Dyna*, 165, pp. 255-265, 2011.

[26] García, J., NEO: Networking and Emerging Optimization.

Available: http://neo.lcc.uma.es/staff/jmgn/doc/Memoria_PFC_JMGN.pdf 09 14, 2006. [cited 02-12-2011]

[27] Cadavid, J., Rivera, J. y Ceballos, Y., Agrupamiento Homogéneo de Elementos don Múltiples Atributos mediante Algoritmos Genéticos, *Dyna*, 165, pp. 246-254, 2011.

[28] Ortiz, N. R. y Gómez, S., Desarrollo de un Algoritmo Genético para el Diseño de Sistemas de Manufactura Celular a partir de una nueva Función de Aptitud, *UIS Ingenierías*, 6 (2), pp. 71 – 82, 2007.