A Hybrid Simulation Based on Multi-Objective Algorithm for Manufacturing Cells Optimization

Carrum-Siller Elías[1], Torres-Treviño Luis[2], Pérez-Villanueva Pedro[1]

[1]Corporación Mexicana de Investigación en Materiales S.A.
Ciencia y Tecnología No 790 Col. Saltillo 400, C.P. 25290 Saltillo, Coahuila. México. Tel: +52 (844) 411-3200
[2]Centro de Innovación, Investigación y Desarrollo en Ingeniería y Tecnología
Universidad Autónoma de Nuevo León Km. 10 de la nueva carretera al Aeropuerto Internacional de Monterrey
PIIT Monterrey, Apodaca, Nuevo León, México
eliascarrum, pperez} @comimsa.com
luis.torres.ciidit@gmail.com

Abstract. This paper presents a hybrid simulation based on multi-objective algorithm for creation and optimization of manufacturing cells; these cells are created by using the principle of group technology with binary matrices. The algorithm used in this paper is the NSGA-II using a seed made by a modified ART neural network, the NSGA-II algorithm is used to maximize the final inventory, minimize the WIP, and minimize the movement time in order to create an optimized cells, after that, the best solution is compared using simulation against the original matrices, the cell formation given by and modify ART neural network and the NSGA-II algorithm without the seed. The solution given by the hybrid NSGA-II algorithm gives superiors solutions when the seed is used.

Keywords: Hybrid systems, optimization based simulation, neuronal networks, adaptive resonance theory.

1 Introduction

One of the most used philosophies in cell formation on manufacturing areas is Group Technology (GT). This philosophy consists in create groups of similar parts and machines into families to minimize the movement of parts between machines in the same cluster. Also GT has been applied to minimize material handling and relocation costs. GT can be associated with other benefits like [1]:

- Reduction of storage material in work areas
- Reduction of bottlenecks
- Reduction of transportation among operations

Over the past years the classic techniques like ranking order algorithm, P-median model, bond energy, etc. have been replaced by using a soft computing approach where most used are the evolutionary algorithms and the neuronal networks (NN). A few applications of GA's and NN's will be presented subsequently.

Genetic algorithms (GA) are the most used evolutionary algorithm. GA was developed by Holland in 1975 and has grown as the most used paradigm to solve optimization problems [2]. There are several variants of the GA; nevertheless, all have four general procedures: evaluation of the individuals, selection of the best individuals, crossover, and mutation of individuals [3]. Every individual is a solution represented as a binary vector and a set of solutions represents a population of potential solutions or individuals making analogy to natural processes. Since the introduction of the GA made by Holland has been widely used to solve group technology (GT) problems [4], some of them seek to reduce intracellular and intercellular movements [5], also has been applied to minimize the grouping efficiency [6], not only simple GA have been applied to solve GT also these type of problems are been solved with multi-objective algorithms that seek to minimize the intercellular flows and the cell load variation [4].

The use of neural networks to solve GT problems is extensively used due to the unique capabilities on the recognition of patters from experience and generates new knowledge [7]. In literature can be found various examples of neural network used to solve GT problems, like malavé [8] use a neural network based on a competitive learning rule to group the machine part incidence matrix, another neural network applied to solve GT problems is the Kohonen self organized featuring maps [9]. El-Kebbe [7] made a comparison among three networks the Kohonen network the ART1 network and the Fuzzy ART network in he is research he found that the three networks obtain similar results.

2 Group Technology Simulator

The simulator used in the NSGAII to obtain the evaluation is based in the GT principle where will be determinate the total amount of pieces obtained, the work inventory process (WIP), and the transportation time between machines, this simulator use 3 different matrix to work , the machine-process matrix, the distance between machines matrix and the process distribution matrix.

The machine-process matrix is formed by obtaining the total amount of machines and process to evaluate; this will create a matrix of size MxN. The machines will be represented by the lines and the process by the columns, the matrix will be filled with 1 if the machine can made the process and with 0 or empty otherwise. After the creation of the machine-process matrix it is necessary a recollection of information about the material movement, this creates a new matrix of size MxM where will contain the distance of movement for the part in each process, the distance will be collected based on the first matrix using the sequence of the process.

The distribution matrix will be of size MxN where each element of the matrix will be represent the statistical distribution for each process in case the process cannot be elaborated in one machine this will be 0. For the present paper the simulator was tested using random distance matrices, one for each problem. The process distribution matrix was created using a normal distribution between 0 and 2 minutes each problem.

3 Modify ART

The adaptive resonance theory neuronal network was created by Grossber & Carpenter [10] and applied for GT formations in the recognition of categories by Dagli [11]. A modification of the ART1 [10] neural network is made. This modification consist in eliminate the vigilance parameter ρ, with this is eliminated the problem of setup a viable vigilance parameter. The modification is presented as following:

Step 1. Initialize the weight matrix β_{ij}; where m are the machines and n are the processes this is given by the size of the entry matrix.

$$\beta_{ij} = \frac{1}{(m+1)} \tag{1}$$

$$t_{ij} = 1 \tag{2}$$

$$i = 1,2, \ldots \ldots . m$$

$$j = 1,2, \ldots \ldots . . n$$

Step 2. Introduce input x.

Step 3. Multiply the entry vector x by the weight matrix β :

$$\sigma = \beta * x_i \tag{3}$$

Step 4. Determine the winning neurons, in this step the neural network will provide the cell where it belongs to each entry.

$$y = max\{\sigma_j\} \tag{4}$$

$$j = 1,2, \ldots . . m$$

Step 5. Select the matrix of weights t in the winning column j of step 4.

$$t_y \tag{5}$$

$$j = 1,2, \ldots . . m$$

Step 6. Make estimation between step 5 and entry x as shown:

$$\alpha = t_{yj} \& x_i{}' \tag{6}$$

Where & is given by the *AND* operator.

Step 7. The weight matrix β and t are updated in accordance with the winning neuron y , the 0.5 constant is usually associated with the parameter λ in the ART1 algorithm this can be set up between 0 and 1, for this modification the parameter will be remain as a constant:

$$\beta(y) = \frac{\sigma}{0.5(\sum_{i=1}^{n} \sigma)} \tag{7}$$

$$t(y) = x_i \tag{8}$$

$$j = 1,2, \dots \dots . m$$

Step 8. Back to step 2 until all the inputs are finished.

4 NSGA-II

The fast elitist non-dominated sorting genetic algorithm II (NSGA-II) was proposed by Deb *et al* [12]. This is an improved version of the original non dominated sorting genetic algorithm (NSGA) [13]. This algorithm is used to found the optimal solution for manufacturing cells trying to maximize the final production, minimize the work inventory process and minimize the material handling time that gives a determinate cell formation. In order to improve the computational performance of the algorithm a modification of the same is made. This alteration consist in the creation of a data base included in the algorithm to eliminate the simulations already made, this modifications is necessary due the simulation time is elevated for big matrices, for example, a matrix of size 40 x 100 the simulation time is around 10 to 15 minutes, since the algorithm is running using 100 individuals and 100 generations the computational time to found a result was too high. The program algorithm is illustrated subsequently:

Variables:

```
NTI= Number population.
MPM= Machine-Part Matrix.
Pc= Crossover probability.
Pm= Mutation probability.
NTGen= Total number of generations.
Fe1= Final production.
Fe2= WIP.
Fe3= Time of movement.
BDM= Data base.
FM3= Non-Dominated population.
MS= Machine selection.
PS= Process selection.
HM= Variation machine population.
HP= Variation process population.
```

First the population is created using permutations, two matrices will be created one for the machines and one for the process these matrices will be evaluated in corresponding each other.

```
[Machines Process]←create permutation (MPM, NTI);
```

A creation of manufacturing cells is created using the modify ART1 algorithm, then a 10 % of the population is replaced by the solution given by the neural network.

```
[Machine2 Process2]← ART1 (MPM);
```

A creation of an empty variable is need to generate the data base for the simulations, this data base is called BDM which contains the machine and process already evaluated and the result of the simulation. If the machine- part matrix already exists in the date base obtain the results of the previous simulation avoiding the repetition of it.

```
[Fe1 Fe2 Fe3 BDM]←Evaluation (MPM, Machines,Process, BDM);
```

Then the normal procedure of the NSGA-II algorithm is elaborate; short the non-dominated of the population, a selection is made using the tournament procedure, then the variation process which uses the PMX crossover procedure and the union of the previous selected population with the new population. Then make the same until all generations are finished.

```
Fm3← Short Non-Dominates (Fe1, Fe2, Fe3);
[MS PS]←Selection (Fm3, Machines, Process);
[HM HP]←Variation (MS,PS,Pc,Pm);
Machines← union (Machines,HM);
Process← union (process,HP);
For i=1:NTGen
     [Fe1 Fe2 Fe3 BDM]←Evaluation (MPM, Machines, Process,   BDM);
    Fm3← Short Non-Dominates (Fe1, Fe2, Fe3);
    [MS PS]←Selection (Fm3, Machines, Process);
    [HM HP]←Variation (MS,PS, Pc, Pm);
    Machines← union (Machines, HM);
    Process← union (process, HP);
End
```

5 Experimentation

For the experimentation where used 2 different matrices this matrices have dissimilar number of machines and process, the first one of size 4 x 5 and the second one 6 x 1.

Each matrix where solved using the ART1 neural network, the NSGA-II and the NSGA-II using seed. The solutions take form the NSGA II algorithms where chosen by the user helped by the Pareto front where the user take the most fitted solution to his problem which is showing in figure 1. Each solution where compared using the same simulator software using a media of 25 simulation for each example against the formation given by the 3 techniques and the original form of the matrix.

5.1 Example 1 Matrix 4x5

Table 1 Initial matrix size 4x5

	1	2	3	4	5
1		1		1	1
2	1		1		
3		1		1	
4	1		1		

Table 2 Distance Table

0	8	9	15
8	0	15	9
9	15	0	10
15	9	10	0

As it shows on the table 5 the hybrid NSGA-II has a better performance in the total amount of pieces produced, however the ART1 have a less inventory process.

Table 3 compared solutions

Matrix 4 x 5			
	Final Production	WIP	Handle Time
Original	164	7	254
ART1	200	5	158
NSGAII	170	10	154
Hybrid NSGAII	166	7	176

5.2 Example 2 Matrix 6x11

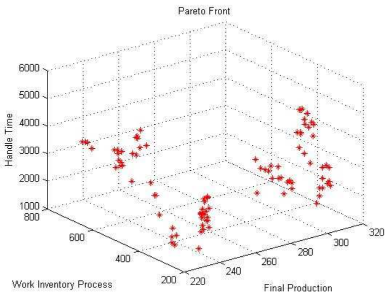
Table 4 Initial Matrix

	1	2	3	4	5	6	7	8	9	10	11
1		1		1		1				1	
2		1	1		1		1				
3			1					1	1	1	1
4			1	1		1			1		1
5	1				1					1	
6	1			1			1				

Table 5 Compared Solutions

Matrix 6 x 11			
	Final Production	WIP	Handle Time
Original	66	8	392.71
ART1	67	10	408.66
NSGA-II	132	8	316.95
Hybrid NSGA-II	132	2	334

For the example 5.2 was selected an exceptional matrix which means that this cannot be separable in any of his members in this example the NSGA-II algorithm have an outstanding result against the ART1. When the seed is used the final production shows no improvement however the WIP is reduced.



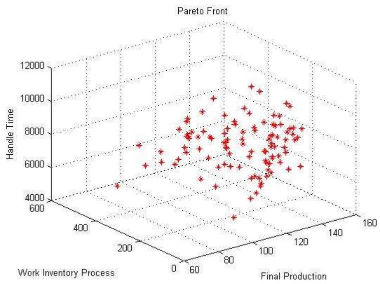


Fig. 1 Pareto Front of example 1 and example 2

6 Conclusions

For the two experiments the hybrid NSGA-II shows a better solution, optimizing the final production, reducing the WIP and reducing the handle time, also was presented a complex problem where any of the parts can be separated called exceptional matrices in this problem the NSGA-II algorithm shows an outstanding performance optimizing the final production in more than 100 %, also when the hybrid NSGA-II was used the final production was the same but the WIP was reduced. Given as a conclusion that the use of the hybrid NSGA-II algorithm using simulation instead of regular evaluations functions is a very powerful tool for solve GT problems.

7 Future Work

For future work is planned to compare another Multi-Objective algorithms using simulation against the NSGA-II to evaluate the behavior of the algorithms and determine the best algorithm for solve GT problems.

References

1. Groover P. Mikell, , Automation, production systems and computer integrated manufacturing, Chapter 15, 420-459, (2001)
2. Mitsuo Gen and Runwei Chen, “Genetic Algorithms & Engineering Optimization”. John Wiley & sons Press (2000)
3. David E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Oxford University Press, New York. (1996)
4. Yasuda K., Huy L. and Yinz Y. A grouping genetic algorithm for the multi-objective cell formation problem, International Journal of Production Research, Vol. 43, No. 4, 829–853(2005)
5. Chan, F.T.S., Lau, K.W., Chan, P.L.Y. and Choy, K.L. Two-stage approach for machine-part grouping and cell layout problems, International Journal of Robotics and Computer Integrated Manufacturing, Vol. 22, Issue 3, pp. 217-238. (2006)
6. Meents Ingo, A Genetic Algorithm for the Group Technology Problem, E.J.W. Boers et al. (Eds.) EvoWorkshop, LNCS 2037, pp. 90{99, 2001. (2001)
7. El-Kebbe, Dania Adnan; Danne, Christoph, On Adapting Neural Networks to Cellular manufacturing. In: Proceedings of Complex Systems and Self-organization Modelling (CoSSoM 2006), Hawai, USA, 1. Jan. (2006)
8. Malavé C.O. and Ramachandran, S., Neural network-based design of cellular manufacturing systems. Journal of intelligent Manufacturing, volume 2, pages 305–314. (1991).
9. Kiang M.Y., Kulkarni U.R. & Tamb K.Y., Self organizing map network as an interactive clustering tool -An application to group technology, *Decision Support Systems*, *15*(4), 1995, 351-374. (1995)
10. Grossber & Carpenter, a massively parallel architecture for a self-organized neuronal pattern recognition machine, Computer Vision, Vol. 37, 54-115 (1987)
11. Dagli, C., Huggahalli, R., Machine-part family formation with the adaptive resonance theory paradigm. International Journal of Production Research 33, 893–913. (1995).
12. Deb, K. Pratap, A. Agarwal, S. Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA-II, Page 1. 182 IEEE transactions on evolutionary computation, Vol. 6, NO. 2, (2002)
13. Srinivas, N. and K. Deb., Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Journal of Evolutionary Computation*, **2**(3), 221-248. (1994)