



Ingeniare. Revista Chilena de Ingeniería

ISSN: 0718-3291

facing@uta.cl

Universidad de Tarapacá

Chile

Castrillón, Omar Danilo; Sarache, William Ariel; Giraldo, Jaime Alberto  
DESIGN OF A HYPERHEURISTIC FOR PRODUCTION SCHEDULING IN JOB SHOP  
ENVIRONMENTS

Ingeniare. Revista Chilena de Ingeniería, vol. 18, núm. 2, agosto, 2010, pp. 203-214

Universidad de Tarapacá

Arica, Chile

Available in: <http://www.redalyc.org/articulo.oa?id=77216407007>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System

Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal

Non-profit academic project, developed under the open access initiative

## DESIGN OF A HYPERHEURISTIC FOR PRODUCTION SCHEDULING IN JOB SHOP ENVIRONMENTS

### DISEÑO DE UNA HIPERHEURÍSTICA PARA LA PROGRAMACIÓN DE LA PRODUCCIÓN EN AMBIENTES JOB SHOP

Omar Danilo Castrillón<sup>1</sup> William Ariel Sarache<sup>1</sup> Jaime Alberto Giraldo<sup>1</sup>

Recibido 19 de agosto de 2008, aceptado 26 de julio de 2010

*Received: August 19, 2008 Accepted: July 26, 2010*

#### RESUMEN

El objetivo del presente trabajo es disminuir el tiempo de proceso (Makespan) e incrementar el tiempo de trabajo de las máquinas, disminuyendo el tiempo de ocio en ambientes de job shop, a través del diseño de una hiperheurística basada en colonia de hormigas y algoritmos genéticos. Este trabajo se desarrolla en dos etapas: en la primera se realiza la definición e identificación de una hiperheurística para la secuenciación de procesos en ambientes job shop. En la segunda etapa es mostrada la efectividad del sistema en la programación de la producción. En el proyecto de investigación se seleccionó una empresa del sector metalmecánico, donde por medio de una combinación de colonia de hormigas y algoritmos genéticos se programa la ruta óptima para un pedido, logrando la optimización o suboptimización de su respectivo tiempo total de proceso en un porcentaje superior al 95%.

Palabras clave: Taller de trabajos, programación, heurísticas, simulación, función evaluación, tiempo total de proceso, tiempo ocioso.

#### ABSTRACT

*The objective of the present work is to diminish the total process time (Makespan) and to increase the machine process time, by diminishing the idle time in a job-shop environment. Through the design of a hyperheuristic based on an ant colony and genetic algorithms. This work is developed in two phases: in the first phase, a hyperheuristic identification and definition is carried out for sequencing processes in job shop environments. In the second phase, the system effectiveness in the traditional production programming is shown. In the investigation project, an enterprise from the metal mechanic sector was chosen, where by means of a combination of an ant colony and genetic algorithms, the optimal route for an order is scheduled, achieving the optimization or suboptimization of its respective total process time in an upper percentage of 95%.*

*Keywords: Job shop, scheduling, heuristics, simulation, fitness, evaluation function, makespan, idle time.*

#### INTRODUCTION

The different uses of heuristics [1-3] can be found in artificial intelligence, specifically in operational research. In a generic sense, this intelligence refers to all methods, techniques or intelligent procedures for carrying out a task, that are not the result of a rigorous formal analysis but of an expert's knowledge. The heuristics analysis leads to meta heuristics (beyond heuristics) which is a set of intelligent strategies to improve very general heuristic procedures with a high performance. In a slightly different sense, the term hyperheuristic can be defined as a heuristic, chosen between heuristics in order to solve a given optimization problem. Currently, the use of hyperheuristics in the processes of production

scheduling and programming in Job Shop environments has not been widely spread [4]. However, there are some hyperheuristics based on methodologies such as heuristics [5], algorithms [6-10], genetic algorithms [11-14], intelligent particles [15-19], ant colonies [20], immune systems [21], and others [22-28].

Some of the following problems in this engineering area might be solved: resources distribution, inefficient machines assignments, inadequate arrangement and sequence of I lots in each one of the J machines, not fulfilling the delivery terms, inappropriate demand evaluation, difficulty in handling the purchase orders, deficient inventory control, frequently pushing work activities, imbalance in the work centers capacity and dissatisfaction in quality.

<sup>1</sup> Departamento de Ingeniería Industrial. Universidad Nacional de Colombia. Campus la Nubia. Manizales, Colombia.  
E-mail: odcastrillong@unal.edu.co; jaiagiraldog@unal.edu.co; wasarachec@unal.edu.co

There might be multiple heuristics for trying to solve a great part of the above problems, but these techniques are static or of a short reach and present problems when the number of lots  $I$  and machines  $J$ , changes considerably. Therefore it could be said that there are no techniques for a general solution; and even the informed simulation techniques are difficult to apply due to the very high number of possible solutions to the problem  $I!J$ .

Currently, the great dynamism of the artificial intelligence techniques makes these arise as an alternative solution to this problem. As they establish newer and better solutions, starting from existing solutions, great versatility in the solution of this sort of problems is allowed.

As previous affirmations, genetic algorithms, algorithms based on ant colonies, algorithms based on intelligent particles, expert systems, neuron nets, diffused logic, heuristics, taboo search and its variants; are different artificial intelligence techniques [29].

The algorithms based on ant colonies, are related to the literature [30] for solving hard NP problems with variable characteristics in time, combining optimization, routing of communication networks, multiobjective optimization and production scheduling problems.

Likewise, genetic algorithms have arisen as a new alternative for solving problems with production scheduling. Their main objective is finding the priority sequence rules that will allow the optimization of a desired objective. Initially, through a priority rule, an initial sequence is generated for programming the orders in each one of the job centers. The order of this sequence is consecutively modified by means of a genetic algorithm. The new population effectiveness (new order of the sequence) is evaluated by means of an evaluation function named fitness and previously defined on the basis of the total makespan. Other strategies are based on the use of genetic algorithms, the sequence of rules that allow the optimization of the desired objective.

In the industrial field, the artificial intelligence has permitted a great variety of solutions to the problems, such as: the optimization of any calculable function, independent of being analytic or digital. The solution of the classic traveler's problem, where the objective is to visit  $m$  different cities, that communicate among them, choosing the shortest route in a closed circuit, where the initial city should be the same as the final one; planning class schedules; planning public transportation; planning airplanes landing; job shops loading, queues analyses in

dynamic programming problems, new heuristics based on the taboo search applied to Job shop Scheduling problems JSSP [31-34].

Though the use of evolutionary algorithms, based on ant colonies and genetic algorithms has been of great interest in the diverse industrial fields. The use of these techniques does not always lead to find an appropriate solution to the problem [30]. Occasionally, it is important to design strategies that help with good solutions for finding the respective problem. This last aspect constitutes this article central problem, in which the combination of two evolutionary techniques is analyzed in the solution of production sequencing problems. Finally, the results of this analysis let us show how its combination facilitates the makespan optimization or suboptimization in cycles of reasonable computations.

In this paper we shall propose a methodology based on artificial intelligence techniques. These techniques, as are illustrated by some authors [35], adequate in processes simultaneously involve dynamic-stochastic, static-stochastic, static-determinist and dynamic-determinist aspects.

## METHODOLOGY

For a *JSSP* solution, several utilizations related to artificial intelligence have been described. Nevertheless, in this section, and as outlined in the introduction, a new methodology is proposed, based on ant colonies and genetic algorithms, with the purpose of optimizing or suboptimizing the solution for these types of problems [36].

**Step 1. Representation.** Formally, the Job Shop Scheduling Problem (*JSSP*) can be represented by a graph [37], where there is a node for each operation  $i \in V$ . (A collection of all the tasks) where,  $\theta$  and  $f$  are two nodes that represent the beginning and the end of all the tasks. For every two consecutive operations in the same job  $(i, j) \in A$  (collection of pairs of all the operations whose precedence relation is determined by each job scheduling technique), there is a directed arch and the operations  $\theta$  and  $f$  are the first and last operations of all the jobs, respectively. For each pair of operations that use the same machine  $\{i, j\} \in E_k$  (collection of operations that are executed in machine  $k$ ) there are two arches  $(i, j)$  and  $(j, i)$  in opposite directions that indicate which operation should be executed before. Each node  $i$  has an associated weight  $p_i$  that indicates the operation execution time  $i$ . Figure 1 represents the disjointed graph associated to an  $n$  machines and  $m$  jobs problem ( $n \times m$ ).

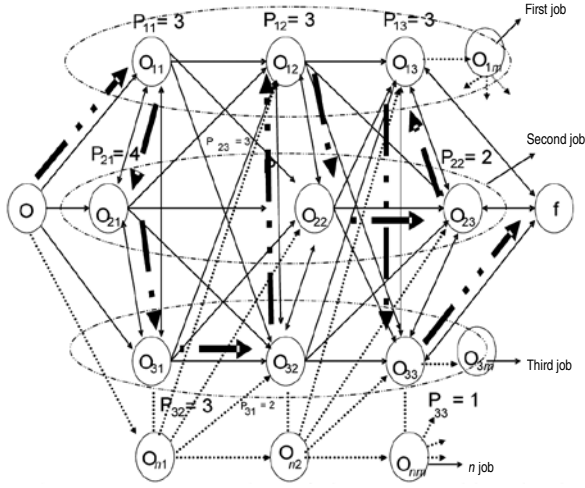


Figure. 1 Representation of the JSSP problem in the form of a graph [37].

Where  $O_{ij}$ , represents job  $i$  in the machine  $j$  and  $P_{ij}$  represents the makespan for the job  $i$  in the machine  $j$ .  
 $m$  represents max number of machines.  
 $n$  represents max number of jobs.

The solution for a job shop scheduling problem (JSSP) consists of the selection of the order in which the operation should be carried out in each machine, which means the selection of one of each pair of arches in opposite directions. Therefore, the resulting graph will not be cyclic and the total length of the longest route between node  $O$  and node  $f$  will be the minimum. If an orientation of arches gives place to a cyclic graph, the corresponding orientation or solution is denominated unfeasible [38].

For example, in Figure 1, the dotted arrows route represents a possible solution for the problem proposed. In the formulated solution, indicated by the nodes sequence  $O, O_{11}, O_{21}, O_{31}, O_{32}, O_{12}, O_{22}, O_{23}, O_{13}, O_{33}, f$ , each node  $O_{ij}$ , indicates that the job  $i$  is done by the machine  $j$ . Furthermore, the sub index  $i$  of the nodes comprehended between position 2 and position 4 ( $O_{11}, O_{21}, O_{31}$ ), that are (1,2,3), indicates that the first operation of the jobs (1,2,3) is done in the first machine, in the respective order established by the subscript. Likewise, the first subscript  $i$  of the nodes comprehended between position 5 and 7 ( $O_{32}, O_{12}, O_{22}$ ) that are (3,1,2) indicates that the second operation of the jobs (3,1,2) will be carried out by the second machine, in the respective order established by the subscript. A similar process occurs with the nodes comprehended between positions 8 y 10 ( $O_{23}, O_{13}, O_{33}$ ) [39].

**Step 2. Matrix.** In order to facilitate the proposed programming methodology, the graph in Figure 2, with its respective process times, is represented by a matrix. If there is an arch between a pair of nodes ( $O_{ij}, O_{kl}$ ), like in this matrix, the cell value is determined by the row ( $O_{ij}$ ) and the column ( $O_{kl}$ ) will be the same as the amount of the operation in node  $O_{kl}$ . In the other cases, when there is no connection, the amount will be equal to infinity. This matrix will represent the heuristic information described in equation (1) of this methodology.

In a similar manner, as was described in the previous paragraph, and with the objective of defining the probability function  $p$  established in equation (1) of this methodology, a new matrix is established (which is managed in a similar manner to the previous one) with the information on the corresponding traces of pheromones. Initially, this matrix will be started at zero, due to the fact that the ants have not built solutions [39].

**Step 3. Initial solution.** An  $m_a$  artificial ant colony is created in such a manner that they will move in a concurrent and asynchronous manner, between nodes  $O$  and  $f$  through the adjacent problem states, represented in the form of a graph. The path traveled by each ant is a valid solution to the problem if it represents the longest route between nodes  $O$  and  $f$ . This movement is made following a transition rule based on a probability function  $p$ , which is calculated based on the heuristic and pheromone information matrix; as illustrated in equation (1):

$$p_{rs}^k = \begin{cases} \frac{[\tau_{rs}]^\alpha \cdot [\eta_{rs}]^\beta}{\sum_{u \in N_r^k} [\tau_{rs}]^\alpha [\eta_{rs}]^\beta}, & \text{If } S \in N_k(r) \\ 0, & \text{In other case} \end{cases} \quad (1)$$

Where,

$k$ , is the ant of an specific node.

$N_k(r)$ , is the neighborhood reachable by the ant  $k$ , when it is located in node  $r$ .  $\alpha, \beta \in \mathbb{R}$ , are two parameters that ponder the relative importance of the pheromone traces and heuristic information.  $\tau_{rs}$ , is the pheromone trace between node  $r$  and  $s$ .  $\eta_{rs}$ , represents the existing heuristic information between node  $r$  and node  $s$ .

Once an ant has built a valid solution, each pheromone trace in the graph between each pair of nodes (arches), is reduced (in the matrix of pheromone) in a constant factor,  $\tau_{rs} \leftarrow (1-p)\tau_{rs}$ , where  $p$  is the vaporization rate. Subsequently, only the arches of the graph that represent

the solution found by the ant are reinforced, taking this path in opposite sense, according to the route stored in the memory of each ant ( $L_k$ ). A constant quantity of pheromone is deposited (in the matrix of the pheromone) in this inverse path:  $\tau_{rs} \leftarrow \tau_{rs} + \Delta\tau_{rs}^k \quad \forall a_{rs} \in S_k$ , where  $\Delta\tau_{rs}^k = f(C(S_k))$ , and the deposited amount of pheromone depends of the quality  $C(S_k)$  of the solution  $S_k$ , built by ant  $k$ .

With the different interactions in this algorithm, the existing quantity of pheromone in the graph arches (represented in the corresponding matrix). One of the best solutions is to reinforce it, while the quantity of the existing pheromone in the other arches will be penalized by means of the vaporization process. In some cases, especially when  $\beta = 0$ , or when it presents a value close to 0, the solution to the problems depends only on the pheromone traces (as is deduced from equation (1)) producing a stagnation of the problem in the local optimal [29, 39].

**Step 4. The solutions evolution.** To generate a programming sequence between node  $O$  and  $f$  is not enough; it is necessary to consider some of the idle time between the machines, according to the assumption that were previously defined. The consideration of this idle time will allow us to define exactly the time in which each operation should be executed in the different job centers. The manner in which the idle time is considered will hugely influence makespan and the total idle time.

In order to consider the best possible way, the idle time, a strategy based on genetic Algorithm is proposed, illustrated as follows:

Starting with each one of the solutions built by the  $m$  ants and stored in the memory  $Lk$ , two genes are generated. Each collection of vertexes that represents a valid solution will generate the chromosomes of two genes.

For example, the solution:  $(O, O_{11}, O_{21}, O_{31}, O_{32}, O_{12}, O_{22}, O_{23}, O_{13}, O_{33}, f)$  will contain a nodes sequence that represents the programming of the different operations for each job, as was illustrated in the first step of this methodology. Although, the first, second and third operation of each job can be done in any order, it is necessary to keep in mind that two operations from the same job cannot be done in a simultaneous manner, in two different work centers.

In some cases this restriction implies the introduction of some idle time prior to the initiation of some operations, allowing the feasibility of generating a matrix like the one illustrated in Table 1:

Table 1. Introduction of idle times.

Center	T. Idle	T. process	T. Idle	T. process	T. Idle	T. process
1	1	O1	0	O3	0	O2
2	1	O2	0	O1	0	O1
3	1	O3	0	O2	1	O3

In Table 1, the values of 1 will define a greater probability of assigning a delay to the related operation. For example in row 3 column 6 (shaded cell) the value of 1, will indicate that if this operation crosses with operation 3 of row 1 (shaded cell) the minimum possible delay should be introduced in the operation of job center 3, that is, row 3 column 6.

If in Table 1 only the idle time is considered, prior to each operation, a 3x3 matrix is generated, which represents the first gene of the found solution, as it is presented in Table 2.

Table 2. First gene.

1	0	0
1	0	0
1	0	1

In a similar manner, combining the different 1s and 0s of Table 2, the generation of a new gene is feasible.

On the other hand, when the evaluation of the makespan and total idle time is necessary, parting from a genes matrix and considering the restriction due to the fact that two operations of the same job cannot be carried out in a simultaneous manner in two different job centers. In some cases as was expressed formerly, it is necessary to introduce an idle time, prior to the initiation of each operation in a job center. In this process, three fundamental cases of crossing of jobs should be considered; these are illustrated in Figure 2:

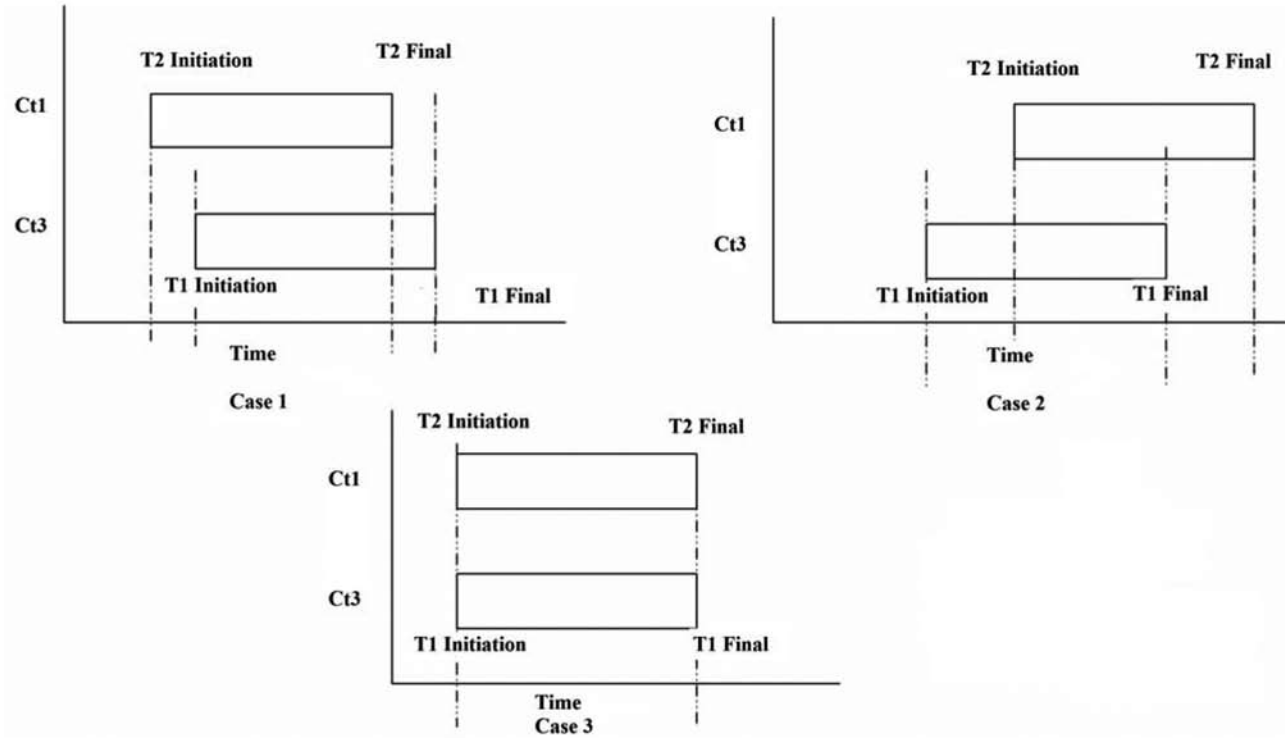


Figure 2. Operation crossing.

Assuming a configuration of 1s and 0s, like the ones illustrated in Table 1, Table 2 and considering the first case of Figure 3, a delay time should be introduced in row 3 column 6 of Table 1, in order to move operation three from the third job center. This delay should be equivalent to:  $\text{delay}(\text{row3}, \text{col6}) = \text{delay}(\text{row3}, \text{col6}) + T2_{\text{final}} - T1_{\text{initiation}}$ . The foregoing delay time will avoid the operations secrecy.

In an opposite case to the previous one, if the work of the first job center is to be moved, then the delay of row 1 would be updated in the following manner:  $\text{Delay row 1, col3} = \text{delay}(\text{row 1, col3}) + T1_{\text{final}} - T2_{\text{initiation}}$ . In an analogical manner, the second and third cases of Figure 3 may be analyzed.

This analysis will begin with the bigger row going to the smaller file. In the case that two operations are preceded by a value of 1, the operation in the biggest row of the matrix will be moved.

Subsequently, the two genes generated, will constitute the initial chromosomes [40], which will be the parents of the coming generations through three probable genetic operators: A crossing reproductive operator denominated partial matching crossover (PMX) with a 95% probability of occurring, a genetic mutation operator with a 5% probability

of occurring [41], or substitute operator by inclusion and a stochastic selection operator. The use of the previous operators in a repeated form during a predetermined number of iterations, will allow the generation of a gene, which represents a feasible solution to the problems with the minimum possible makespan and idle time. Taking the former representation into account, the quality of the solution is evaluated according to equations (2) and (3). In these equations, the processing order described by the subscript  $i$  (step 1) in each chromosome is taken as a reference [42]:

$$Fitness_{\text{makespan}} = \min \left( \max_{1 \leq i \leq n} \left( \max_{1 \leq j \leq m} (P_{ij}) \right) \right) \quad (2)$$

$$Fitness_{\text{idle}} = \min \sum_{j=1}^m f_j \quad (3)$$

Where  $n$ , represents the number of jobs;  $m$ , represents the number of machines,  $P_{ij}$  is the makespan of the job  $i$ , in the machine  $j$ .  $f_j$  is the total  $m$  idle time of the machine  $j$ .

The main objective is to minimize the makespan function, because the idle time is a direct consequence to the total process time.

**Step 5. Problems generalization.** In order to analyze the new methodology effectiveness, its results are compared with other based on artificial intelligence methodologies: random techniques, data mining, taboo search and evolutionary algorithms [43 - 49]. Likewise, for guaranteeing the techniques generalization, the designed methodology is used in a general type problem solution, only limited by computational restrictions.

**Step 6. Variance analysis.** With the purpose of guaranteeing the consistency of the solution, it is necessary that the previous process is repeated during a determined amount of times (treatments) taking, in each treatment, the ten best results of the fitness makespan function (equation (2)). In order to determine if the results correspond to statistically equal or different treatments, a variance analysis should be done under the model  $y_i = \mu + T_i + \varepsilon_i$ , where  $y_i$  represents the variable of the answer,  $T_i$ , the effect caused by the  $i^{th}$  treatment,  $\varepsilon_i$ , the  $i^{th}$  experimental error.

**Experimentation – Case of application.**

This methodology was used in a company of the metal mechanic sector with 10 machines, principally dedicated to manufacturing metal mechanics products.

The fundamental problem is centered on an insufficient assignment of loads and subsequently, an inefficient use of resources, resulting in over costs and orders unfulfilled. The job centers in the company, object of study are: center one (To cut, to ring, to drill-1), center two (to forge, to drill-2, to smooth), center three (to mold, to pack, to cool, to weld). The average times in each center are:

Table 3. Average time. Bulb sockets manufacturing.

Job/center	1	2	3
1	2	2	4
2	22	24	6
3	4	29	8

It can be experimentally demonstrated, that the process operation time in a job shop, is a normal probability function with known mean and standard deviations.

**RESULTS AND DISCUSSIONS**

**Step 1. Representation.** Once the enterprise information has been analyzed, as detailed in Table 3, the JSSP problem of this enterprise is schemed in the graph of Figure 3, which illustrates all the possible operation sequences for bulb sockets manufacturing.

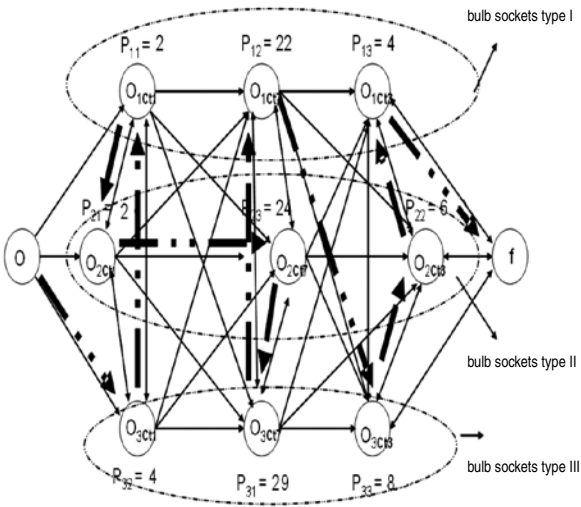


Figure 3. JSSP problem in a graph form.

In an analogous manner to the first point proposed in this methodology, the dotted arrows route represents a possible solution for the problem (not necessarily the optimal one). In the presented solution, indicated by the nodes sequence **O**, **O<sub>3ct1</sub>**, **O<sub>1ct1</sub>**, **O<sub>2ct1</sub>**, **O<sub>2ct2</sub>**, **O<sub>3ct2</sub>**, **O<sub>1ct2</sub>**, **O<sub>3ct3</sub>**, **O<sub>2ct3</sub>**, **O<sub>1ct3</sub>**, **f**, the nodes understood between position 2 and position 4 (**O<sub>3ct1</sub>**, **O<sub>1ct1</sub>**, **O<sub>2ct1</sub>**), indicate that the jobs first operation (3,1,2) will be carried out in this order in the first job shop. A similar process occurs with the nodes understood between the positions 5-7 and 8-10.

**Step 2. Matrix.** With the purpose of facilitating the proposed methodology, the graph of Figure 3, with its respective makespan, is represented in Table 4:

Table 4. Heuristic information from the chart in Figure 3.

Nodes	0	O <sub>1ct1</sub>	O <sub>1ct2</sub>	O <sub>1ct3</sub>	O <sub>2ct1</sub>	O <sub>2ct2</sub>	O <sub>2ct3</sub>	O <sub>3ct1</sub>	O <sub>3ct2</sub>	O <sub>3ct3</sub>	f
0	Inf	2	Inf	Inf	2	Inf	Inf	4	Inf	Inf	Inf
O <sub>1ct1</sub>	Inf	Inf	22	Inf	2	24	inf	4	29	inf	Inf
O <sub>1ct2</sub>	Inf	Inf	Inf	4	inf	24	6	Inf	29	8	Inf
O <sub>1ct3</sub>	Inf	Inf	Inf	Inf	Inf	Inf	6	Inf	Inf	8	0
O <sub>2ct1</sub>	Inf	2	22	Inf	Inf	24	inf	4	29	inf	Inf
O <sub>2ct2</sub>	Inf	Inf	22	4	Inf	Inf	6	Inf	29	8	Inf
O <sub>2ct3</sub>	Inf	inf	Inf	4	Inf	Inf	Inf	Inf	Inf	8	0
O <sub>3ct1</sub>	inf	2	22	Inf	2	24	Inf	Inf	29	inf	Inf
O <sub>3ct2</sub>	Inf	Inf	22	4	inf	24	6	Inf	Inf	8	Inf
O <sub>3ct3</sub>	Inf	Inf	inf	4	Inf	Inf	6	Inf	Inf	Inf	0
f	Inf	inf	Inf	inf	Inf	inf	inf	inf	inf	inf	Inf

In the matrix described in the Table 4, each row and each column represents a node of the graph. If there is an arch between a pair of nodes ( $O_{ij}$ ,  $O_{kl}$ ), it is represented with a value equal to the operation value in node  $O_{kl}$ . In some other cases, when there is no connection, the matrix will have an infinite value. The matrix is built in an analogous way, with the information from the pheromone traces. This matrix will initially be at zero, due to the fact that the ants will not have built solutions (Table 5).

Table 5. Information from the pheromone traces.

Nodes	O	$O_{1C1}$	$O_{1C7}$	$O_{1C3}$	$O_{2C1}$	$O_{2C7}$	$O_{2C3}$	$O_{3C1}$	$O_{3C7}$	$O_{3C3}$	f
O	Inf	0	Inf	Inf	0	Inf	Inf	0	Inf	Inf	Inf
$O_{1C1}$	Inf	Inf	0	Inf	0	0	inf	0	0	inf	Inf
$O_{1C7}$	Inf	Inf	Inf	0	inf	0	0	Inf	0	0	Inf
$O_{1C3}$	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	0	0
$O_{2C1}$	Inf	0	0	Inf	Inf	0	inf	0	0	inf	Inf
$O_{2C7}$	Inf	Inf	0	0	Inf	Inf	0	Inf	0	0	Inf
$O_{2C3}$	Inf	inf	Inf	0	Inf	inf	Inf	Inf	Inf	0	0
$O_{3C1}$	inf	0	0	Inf	0	0	Inf	Inf	0	inf	Inf
$O_{3C7}$	Inf	Inf	0	0	inf	0	0	Inf	Inf	0	Inf
$O_{3C3}$	Inf	Inf	inf	0	Inf	Inf	0	Inf	Inf	Inf	0
f	Inf	inf	Inf	inf	Inf	inf	inf	inf	Inf	inf	Inf

**Step 3. Initial solution.** As each one of the ants proposed for this problem builds a valid solution, the amount of pheromone will be updated along the entire solution route, with a value directly proportional to the quality of the solution, as is illustrated in step 3 of the proposed methodology. The best solutions built by the ants in this stage are shown in Table 6.

Table 6. The best solutions built by the ants.

Job center 1			Job center 7			Job center 3		
$O_{1C1}$	$O_{2C1}$	$O_{3C1}$	$O_{1C7}$	$O_{2C7}$	$O_{3C7}$	$O_{3C3}$	$O_{1C3}$	$O_{2C3}$
$O_{2C1}$	$O_{1C1}$	$O_{3C1}$	$O_{1C7}$	$O_{3C7}$	$O_{2C7}$	$O_{2C3}$	$O_{1C3}$	$O_{3C3}$
$O_{3C1}$	$O_{2C1}$	$O_{1C1}$	$O_{3C7}$	$O_{2C7}$	$O_{1C7}$	$O_{3C3}$	$O_{1C3}$	$O_{2C3}$

**Step 4. The solutions evolution.** Taking each one of the solutions constructed by the ant colony as a reference, two genes are generated, which a genetic algorithm will cause to evolve, until they obtain a definitive solution that allows us to calculate the makespan and the idle time as was illustrated in step 4 of this methodology. The most representative solutions are illustrated by the following sequences:  $O_{3C1}$ ,  $O_{1C1}$ ,  $O_{2C1}$ ,  $O_{1C2}$ ,  $O_{2C2}$ ,  $O_{3C3}$ ,  $O_{1C3}$ ,  $O_{3C3}$ ,  $O_{2C3}$ .

If only one representation is considered, in the total makespan function, the best solution can be represented by Figure 4:

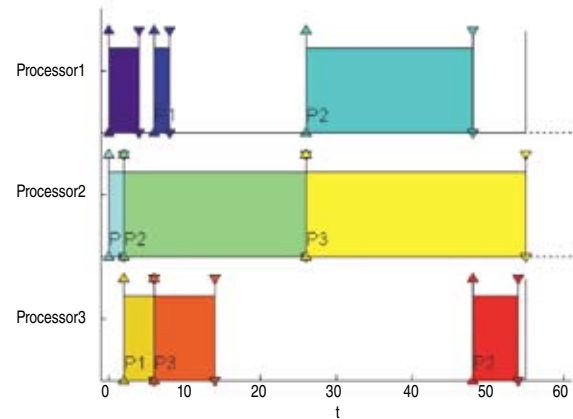


Figure 4. Optimal solution. Makespan = 55.

Idle Time. = 64.

Note: The processor two (center two) does not present idle times. It is the optimal solution.

The solution quality will depend on an adequate evaluation function (equation (2) and equation (3)), based on a previous representation of the same, which minimizes the proposed objective, makespan. Consequently, the calculation of this function evaluation was proposed based on a genetic algorithm which would determine the initiation time of each job in each one of the different centers. The fundamental problem is centered in deciding if it is necessary to consider an idle time before each job is attended; It will be represented by a 1, the contrary case will be represented by a zero. The solution proposed in Figure 4, can be represented by the data from Table 7:

Table 7. Idle time representation for the best solution.

	P1	P2	P3
C1	1	1	0
C2	0	0	0
C3	1	1	0

If only a random idle time is considered in Table 7, a 3 x 3 matrix is generated, which represents the first gene for the genetic Algorithm; as expressed in Table 8:

Table 8. First gene.

1	1	1
0	0	0
0	1	1



In an analogous manner, another gene can be generated which will represent a feasible solution to the problem, as expressed in Table 9:

Table 9. Second gene.

0	0	0
1	0	0
1	1	0

In general, any combination of 1s and 0s in Table 9, will allow us to obtain a feasible solution to the problem. These solutions, as were described in the proposed methodology, will start an evolving process by means of a genetic algorithm until they reach a solution that represents the minimum possible value of the makespan function.

In this case, and after following the proposed methodology, the gene described in Table 10 was obtained, which schematizes the solution of Figure 4.

Table 10. Final gene.

1	1	0
0	0	0
1	1	0

**Step 5. The problems generalization.** The diverse bibliographical references written by the authors of this paper [43-49], evidence that the technique presented has an effectiveness better than 95% and is equivalent to other artificial intelligence techniques. In general, it is feasible to apply this technique to problems with a large number of orders and machines, being limited only by computational restrictions. In this way, Table 11 illustrates a 12 order and 9 job center problem, which has  $12!9 = 1.32 \cdot 10^{78}$  possible solutions.

Table 11. JSSP12X9 Problem.  $12!9 = 1.32 \cdot 10^{78}$  Possible solutions.

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	P <sub>8</sub>	P <sub>9</sub>	P <sub>10</sub>	P <sub>11</sub>	P <sub>12</sub>	Sum
C <sub>1</sub>	4	4	6	5	6	6	8	4	10	4	4	6	67
C <sub>2</sub>	5	8	4	5	6	8	5	8	9	5	5	4	72
C <sub>3</sub>	8	9	5	6	6	11	4	4	8	13	7	8	89
C <sub>4</sub>	4	7	5	4	6	6	6	9	4	9	5	7	72
C <sub>5</sub>	5	6	4	7	6	5	7	8	5	7	6	6	72
C <sub>6</sub>	6	8	6	5	6	4	8	7	6	5	5	5	71
C <sub>7</sub>	7	9	5	8	6	8	9	5	7	5	7	4	80
C <sub>8</sub>	6	7	6	5	6	7	4	4	5	1	5	1	57
C <sub>9</sub>	5	6	4	6	6	4	5	6	5	9	6	4	66

Figure 5 exhibits the best solution achieved with this technique for the problem presented in Table 11.

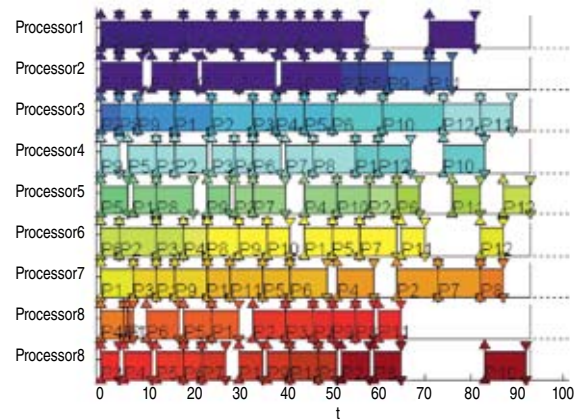


Figure 5. Best solution. Problem Table 11. Makespan = 93, Idle time = 191.

Given that the idle time is a total process time direct consequence, in Figure 5 shows that the third processor (center 3) is under optimized. In Table 11, the total amount of machine 3 process time is 89, hence, the optimal process time would never be inferior to 89. This way, the process time calculated in the solution shown in Figure 6 (93), approaches the optimal solution, at least in a 95.6%. This is  $89/93 = 95.6\%$ .

**Step 6. Variance analysis.** In order to analyze the results consistency, the former process was repeated 10 times (treatments) taking in each treatment (other techniques for problems solving) the ten best results of the fitness function (makespan). Variance analysis was carried out under the model  $\gamma i = \mu + T i + \varepsilon i$ , where  $\gamma i$  represents the variable of the response  $T i$ , the effect caused by the  $i^{th}$  treatment,  $\varepsilon i$ , the  $i^{th}$  experimental error indicate (Tables 12 y 13).

Table 12. Process repeats. Problem Table 3.

T. Process	Process repeats										Sum
	1	2	3	4	5	6	7	8	9	10	
Ant-Gen	55	55	57	55	55	55	55	55	55	59	556
Gen	55	57	55	55	55	55	57	55	55	59	558
Random	55	55	57	55	55	55	55	55	55	55	552
Data Mining	55	55	55	55	55	55	55	59	55	57	556

Table 13. Variance analysis results.

Source	G L	SC	CM	F cal	F Tab
Total Sum		55,90			
Treatment	3,00	1,90	0,63	0,42	2,87
Experimental Error	36,00	54,00	1,50		
Total	39,00	55,90	2,13		
Insignificant model					

The variance analysis pointed out in the Tables 12 and 13, shows that the F distribution value, calculated in (0,05; 3; 36), is lower than the F distribution real values, calculated in the same point. Consequently, it could be affirmed that the obtained results by the different employed techniques, are statistically the same, with a reliability of 99.5%.

With the literature reviewed in this study [50,51], it can be established that the design of hyperheuristics for finding a solution to sequencing problems in job shop environments has not been widely spread. However, it is notorious the use of some heuristics [52], but getting inferior results with the ones used in this analysis.

Similar results have been achieved with heuristics designed under the concept of diffused logic [53]. These heuristics allow the supposition of a delivery date for a job, in a determined interval ( $d_{i1}$ ,  $d_{i2}$ ), in which the satisfaction degree for the job  $j_i$  is a decreasing function. According to the delayed function of the job  $T_i$ , where the last job represents an increasing function according to the unfulfillment of the job.

In the same way other results have been obtained with taboo algorithms. In this analysis [54] the main objective is the minimization of two criteria, the makespan and the maximum lateness, with the purpose of finding an approximation to the Pareto frontier. The author developed a fast and elitist genetic algorithm based on NSGA-II for solving the problem. Other authors developed a metaheuristic for minimizing the makespan, using a simple taboo search algorithm, getting a 90% of approximation to the optimal solution [55].

Currently, there are different methodologies for solving job shop problems. However, the different literature reviewed awards us the conclusion that there is not a solution based on an ant colony in which the evaluation function is helped by a genetic algorithm [2, 3, 6, 22, 23, 54, 55]. With results similar to the ones obtained by other artificial intelligence

techniques, such as: genetic algorithms, random algorithms and data mining (Tables 12 and 13).

## CONCLUSIONS

The ant colonies and genetic algorithms constitute an excellent technique for solving the sequencing process problems in JSSP environments in which the optimum or a close value to the optimum with an approximation percentage superior to 95%, are found. If these techniques are applied in a combined form, as illustrated in this article, it is possible to improve the orders sequence in a Job Center, thus optimizing or sub-optimizing its programming.

As illustrated in the variance analysis, this combination of an ant colony and a genetic algorithm allows us to find an optimum value in the total process time (makespan); with similar results using other techniques like genetic algorithm, random algorithm, taboo algorithm and data mining.

Finally, the artificial intelligence techniques proposed in this study for the solution of a JSSP problem, can be applied to solve general type problems (Table 11) and other industrial problems, such as: Vehicles routing, Nets learning, combinations design for logical circuits, packages routing in communication nets, and in general, for the solution of any NP-Hard, multi-objective problem.

## ACKNOWLEDGEMENTS

The authors are grateful to Ángela González Palacio and Eliana López López, Industrial Engineering students, in appreciation of all their collaboration.

## REFERENCES

- [1] J. Valente. "An analysis of the importance of appropriate tie breaking rules in dispatch heuristics". *Pesquisa Operacional*. Vol. 26, Issue 1, pp. 169-180. 2006.
- [2] E. Candido and F.K. Miyazawa. "Practical comparison of approximation algorithm for scheduling problems". *Pesquisa Operacional*. Vol. 24, Issue 2, pp. 227-252. 2004.
- [3] J. Valente and R. Alves. "A note on scheduling on a single processor with variable speed". *Pesquisa Operacional*. Vol. 23, Issue 3, pp. 457-462. 2003.
- [4] J.B. Santana, C.C. Rodríguez, otros. "Metaheurística, una revisión actualizada". Universidad de la Laguna. San Cristóbal, España. 2004.

- [5] F. Martins e S. Limberger. "Uma heurística de trocas para o problema de sequenciamento de tarefas em processadores uniformes". *Pesquisa Operacional*. Vol. 20, Issue 1, pp. 31-42. 2000.
- [6] E. Colin e T. Shimizu. "Algoritmo de programação de máquinas individuais com penalidades distintas de adiantamento e atraso". *Pesquisa Operacional*. Vol. 20, Issue 1, pp. 19-30. 2000.
- [7] J.N.D. Gupta and J.C. Ho. "Minimizing flowtime subject to optimal makespan on two identical parallel machines". *Pesquisa Operacional*. Vol. 20, Issue 1, pp. 05-17. 2000.
- [8] J. Valente and R. Alves. "Efficient polynomial algorithm for special cases of weighted early/tardy scheduling with release dates and a common due date". *Pesquisa Operacional*. Vol. 23, Issue 3, pp. 443-456. 2003.
- [9] J.N. Figueiredo e C. Clóvis. "Aplicação de métodos de busca em grafos com nós parcialmente ordenados à locação de torres de transmissão". *Pesquisa Operacional*. Vol. 23, Issue 1, pp. 209-220. 2003.
- [10] G.M. Arantes Jr, F.M.G. França e C.A. Martinhon. "Gerando orientações acíclicas com algoritmos probabilísticos distribuídos". *Pesquisa Operacional*. Vol. 25, Issue 3, pp. 301-312. 2005.
- [11] A. Kumar, A. Prakash, R. Shankar and M.K. Tiwari. "Psycho-Clonal algorithm based approach to solve continuous flow shop scheduling problem". *Expert Systems with Applications*. Vol. 31, Issue 3, pp. 504-514. October, 2006.
- [12] L. Monch, R. Schabacker, D. Pabst and J.W. Fowler. "Genetic algorithm-based sub problem solution procedures for a modified shifting bottleneck heuristic for complex job shops". *European Journal of Operational Research*. Vol. 177, Issue 3, pp. 2100-2118. March, 2007.
- [13] Z.X. Guo, W.K. Wong, S.Y.S. Leung, J.T. Fan and S.F. Chan. "Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: A case study based on the apparel industry". *Computers & Industrial Engineering*. Vol. 50, Issue 3, pp. 202-219. 2006.
- [14] H. Ohta and T. Nakatani. "A heuristic job-shop scheduling algorithm to minimize the total holding cost of completed and in-process products subject to no tardy jobs". *International Journal Production Economics*. Vol. 101, Issue 1, pp. 19- 29. May, 2006.
- [15] F. van den Bergh and A.P. Engelbrecht. "A study of particle swarm optimization particle trajectories". *Information Sciences*. Vol. 176, Issue 8, pp. 937-971. April, 2006.
- [16] M.F. Tasgetiren, Y.C. Liang, M. Sevkli and G. Gencyilmaz. "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem". *European Journal of Operational Research*. Vol. 177, Issue 3, pp. 1930-1947. March, 2007.
- [17] Z. Lian, X. Gu and B. Jiao. "A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan". *Applied Mathematics and Computation*. Vol. 175, Issue 1, pp. 773-785. April, 2006.
- [18] C.J. Liao, C.T. Tseng and P. Luarn. "A discrete version of particle swarm optimization for flow shop scheduling problems". *Computers & Operations Research*. Vol. 34, Issue 10, pp. 3099-3111. October, 2007.
- [19] Z. Lian, B. Jiao and X. Gu. "A similar particle swarm optimization algorithm for job-shop scheduling to minimize makespan". *Applied Mathematics and Computation*. Vol. 183, Issue 2, pp. 1008-1017. December, 2006.
- [20] K.L. Huang and C.J. Liao. "Ant colony optimization combined with taboo search for the job shop scheduling problem". *Computers & Operations Research*. Vol. 35, Issue 4, pp. 1030-1046. April, 2008.
- [21] M. Zandieh, S.M.T. Fatemi and S.M. Moattar. "An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times". *Applied Mathematics and Computation*. Vol. 180, Issue 1, pp. 111-127. 2006.
- [22] C.F.M. Toledo, P.M. França, R. Morabito e A. Kimms. "Um modelo de otimização para o problema integrado de dimensionamento de lotes e programação da produção em fábricas de refrigerantes". *Pesquisa Operacional*. Vol. 27, Issue 1, pp. 155-186. 2007.
- [23] D.S. Yamashita e R. Morabito. "Um algoritmo exato para o problema de programação de projetos com custo de disponibilidade de recursos e múltiplos modos". *Pesquisa Operacional*. Vol. 27, Issue 1, pp. 27-49. 2007.
- [24] S.A. de Araújo e M.N. Arenales. "Dimensionamento de lotes e programação do forno numa fundição automatizada de porte médio". *Pesquisa Operacional*. Vol. 23, Issue 3, pp. 403-420. 2003.
- [25] G.C.F. Pileggi, R. Morabito e M.N. Arenales. "Abordagens para otimização integrada dos problemas de geração e seqüenciamento de padrões de corte: caso unidimensional". *Pesquisa*

- Operacional. Vol. 25, Issue 3, pp. 417-447. 2005.
- [26] J.M.V. de Carvalho, M<sup>a</sup> Sameiro de Carvalho e J.A. Oliveira. "Programação de operações em sistemas com tempo de processamento variável". Pesquisa Operacional. Vol. 22, Issue 3, pp. 323-344. 2002.
- [27] L. Belluzzo e R. Morabito. "Otimização nos padrões de corte de chapas de fibra de madeira reconstituída: um estudo de caso". Pesquisa Operacional. Vol. 25, Issue 3, pp. 391-415. 2005.
- [28] F.E. Doy, G. Bressan, G.H. de A. Pereira e M.N. Magalhães. "Simulação do serviço de correio eletrônico através de um modelo de filas". Pesquisa Operacional. Vol 26, Issue 2, pp. 241-253. 2006.
- [29] V. Pureza e R. Morabito. "Uma heurística de busca tabu simples para o problema de carregamento de paletes do produtor". Pesquisa Operacional. Vol. 23, Issue 2, pp. 359-378. 2003.
- [30] I. Fernández de Viana, F. Herrera, S. Alonso y O. Cordón García. "La metaheurística de optimización basada en colonia de hormigas modelos y nuevos enfoques". Departamento de ciencias y computación e Ingeniería Artificial. E.T.S. Ingeniería Informática. Granada, España. 2004.
- [31] A. Azaron, H. Katagiri, K. Kato and M. Sakawa. "Longest path analysis in networks of queues: Dynamic scheduling problems". European Journal of Operational Research. Vol. 174, pp. 132-149. 2006.
- [32] A. El-Bouri, N. Azizi and S. Zolfaghari. "A comparative study of a new heuristic based on adaptive memory programming and simulated annealing: The case of job shop scheduling". European Journal of Operational Research. Vol. 177, Issue 3, pp. 1894-1910. March, 2007.
- [33] C. Yong, P. Li, Z. Guan and Y. Rao. "A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem". Computers & Operations Research. Vol. 34, Issue 11, pp. 3229-3242. November, 2007.
- [34] J.P. Watson, A.E. Howe and L.D. Whitley. "Deconstructing Nowicki and Smutnicki's i-TSAB tabu search algorithm for the job-shop scheduling problem". Computers & Operations Research. Vol. 33, Issue 9, pp. 2623-2644. 2006.
- [35] J. Banks. "Principles of Simulation". In J. Banks (Ed.). Handbook of Simulation; Principles, Methodologies, Advances, Applications and Practice. John Wiley & Sons Inc. New York, United States. 1998.
- [36] W. Hopp and M. Spearman. "Factory Physics". MacGraw Hill, 2<sup>nd</sup> Edition. New York, United States. April, 2000.
- [37] B. Roy and B. Sussman. "Les problèmes d'ordonnancement avec contraintes disjunctives". Technical Report Notes, SEMA. Paris, France. 1964.
- [38] S. López de Haro, P. Sánchez Martín y J. Conde Collado. "Secuenciación de tareas mediante metaheurísticos". VIII Congreso de Ingeniería de Organización. Leganés, España. Septiembre 2004.
- [39] O. Castrillón, W. Sarache and J. Giraldo. "Job Shop Methodology Based on an Ant Colony". Dyna. Vol. 76, Issue 159, pp. 177-184. 2009.
- [40] C. Jou. "A genetic algorithm with sub indexed partitioning genes and its application to scheduling production of parallel machines". Computer & Industrial Engineering - Elsevier. Vol. 48, pp. 39-54. 2005.
- [41] C. Biertwirth. "A generalized permutation approach to job shop scheduling with genetic algorithm". OR Spektrum. Vol. 17, Issue 2-3, pp. 87-92. 1995.
- [42] Y. Tsujimura, Y. Mafune and M. Gen. "Effects of Symbiotic Evolution in Genetic Algorithm for Job Shop Scheduling". Proceedings of the 34<sup>th</sup> Annual Hawaii International Conference on System Sciences. Japan. 2001.
- [43] O. Castrillón, J. Giraldo and W. Sarache. "Solución de un problema Job Shop con un agente inteligente". Ingeniería y Ciencia. Vol. 5 N<sup>o</sup> 10, pp. 75-92. Diciembre 2009.
- [44] O. Castrillón, W. Sarache y J. Giraldo. "Técnicas de programación de la producción. Aplicación en ambientes job-shop". Universidad Nacional de Colombia. N<sup>o</sup> 1. Bogotá, Colombia. 2009.
- [45] O. Castrillón, W. Sarache y J. Giraldo. "Secuenciación en ambientes job-shop por medio de sistemas expertos y agentes inteligentes". 8<sup>va</sup> Conferencia Iberoamericana en Sistemas, Cibernética e Informática. Orlando, Florida, EEUU. 2009.
- [46] O. Castrillón, W. Sarache y J. Giraldo. "Análisis de un problema Job-shop por medio de un sistema experto y un agente inteligente". 3<sup>rd</sup> International Conference on Industrial Engineering and Industrial Management. XIII Congreso de Ingeniería de Organización. Barcelona-Terrassa, España. 2009.
- [47] O. Castrillón, W. Sarache y J. Giraldo. "Técnicas aleatorias en la optimización de procesos job-shop". 9<sup>na</sup> Conferencia Iberoamericana en Sistemas, Cibernética e Informática. Orlando, Florida, EEUU. 2010.

- [48] O. Castrillón y L. Grajales. "Inteligencia artificial en los procesos de planificación de la producción en Sistemas job-shop". Tendencias en ingeniería del software e inteligencia artificial. Universidad Nacional. N° 3. Medellín, Colombia, 2010.
- [49] O. Castrillón, J. Rodrigo y E. Ávila. "Algoritmos evolutivos en ambientes job-shop". Tendencias en ingeniería del software e inteligencia artificial. Universidad Nacional. N° 3. Medellín, Colombia. 2010.
- [50] V. Caraffa, S. Ianes, T.P. Bagchi and C. Sriskandarajah. "Minimizing makespan in a blocking flow shop using genetic algorithm". International Journal of Production Economics. Vol. 70, Issue 2, pp. 101-115. March, 2001.
- [51] Y. Tsujimura, Y. Mafune and M. Gen. "Effects of Symbiotic Evolution in Genetic Algorithm for Job Shop Scheduling". Proceedings of the 34<sup>th</sup> Annual Hawaii International Conference on System Sciences. Hawaii, United States. 2001.
- [52] R.B. Acero y J.F. Torres. "Aplicación de una heurística de búsqueda tabú en un problema de programación de tareas en líneas flexibles de manufactura". Universidad de los Andes. Bogotá, Colombia. 2003.
- [53] S. Petrovic and M.J. Geiger. "A Fuzzy scheduling problem with Dynamic Job Priorities and an Extension to Multiple Criteria". Automatic Scheduling, Optimization and Planning group (ASAP). University of Nottingham. Great Britain. 2003.
- [54] G. Vilmot and J.C. Billaut. "A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem". European Journal of Operational Research. Vol. 190, Issue 2, pp. 398-411. October, 2008.
- [55] C.F. Liaw. "An efficient simple metaheuristic for minimizing the makespan in two-machine no-wait job shops". Computers & Operations Research. Vol. 35, Issue 10, pp. 3276-3283. October, 2008.