

JOB SHOP METHODOLOGY BASED ON AN ANT COLONY

METODOLOGIA JOB SHOP BASADA EN UNA COLONIA DE HORMIGAS

OMAR CASTRILLON

Professor, Industrial Engineering Program, Universidad Nacional de Colombia, odcastrillong@unal.edu.co

WILLIAM SARACHE

Professor, Industrial Engineering Program, Universidad Nacional de Colombia, wasarache@unal.edu.co

JAIME GIRALDO

Professor, Industrial Engineering Program, Universidad Nacional de Colombia, jaiagiraldog@unal.edu.co

Recibido para revisar abril 23 de 2008, aceptado diciembre 15 de 2008, versión final enero 21 de 2009

ABSTRACT: The purpose of this study is to reduce the total process time (Makespan) and to increase the machines working time, in a job shop environment, using a heuristic based on ant colony optimization. This work is developed in two phases: The first stage describes the identification and definition of heuristics for the sequential processes in the job shop. The second stage shows the effectiveness of the system in the traditional programming of production. A good solution, with 99% efficiency is found using this technique.

KEYWORDS: Scheduling, Heuristics, Simulation, Makespan Time, Idle Time.

RESUMEN: El objetivo del presente trabajo es disminuir el tiempo total de proceso e incrementar el tiempo de trabajo de las maquinas, en un ambiente Job Shop, por medio de una heurística basada en la optimización de colonia de hormigas. Este trabajo es desarrollado en dos fases. En la primera es descrita la identificación y definición de la heurística para los procesos de secuenciación en ambientes Job shop. En la segunda etapa, es mostrada la efectividad del sistema en la programación tradicional de la producción. A través de esta técnica una buena solución, con el 99% de efectividad, es encontrada

PALABRAS CLAVE: Planificación, Heurística, Simulación, Tiempo de Proceso, Tiempo Muerto.

1. INTRODUCTION

It is common to find the use of heuristics [1,2] which in a very generic sense refer to those techniques methods or intelligent procedures needed to perform a task, and not the result of a rigorous formal analysis, but the knowledge of an expert. The analysis of heuristics leads to meta heuristics (beyond heuristic) which are strategies to improve very general heuristic procedures with a high performance. In a slightly different sense, the term hyperheuristics encompasses several heuristic

techniques and select the most appropriate heuristics to solve a given optimization problem. Techniques like hyperheuristics, handle the selection of heuristic methods on a lower level; and depending on the state of the solution, it determines at each step, the heuristic method to be applied.

Currently, the use of heuristics in the processes of production scheduling and programming in Job Shop environmental surroundings is not widely spread; though these are hyper heuristics based on

methodologies such as heuristics, algorithms, genetic algorithms [3,4], intelligent swarm [5,6] [7], ant colony optimization (ACO) [8,9,10, 11,12,13,14], and immune systems [15], among others [16]. But, the use of these techniques does not always lead to a good solution in the processes of production scheduling and programming. Sometimes it is important to design strategies to help find suitable solutions for a problem. This last aspect constitutes the central objective of this article, in which a technique based on ACO, is analyzed.

In the problems JSSP, N work must be done by M machines. This aspect has several programming problems. By using this ACO technique, it will be possible to solve some of the main problems in this area, such as: distribution of resources, inefficient assignment of machines, inadequate arrangement and sequence of I lots in each one of the J machines, not fulfilling the terms of delivery, inappropriate evaluation of demand, difficulty in handling the purchase orders, deficient control of inventories, and frequent actions of pushing.

Multiple heuristics have been developed to solve work unbalance in the capacity of job shop and dissatisfaction in quality conditions, but these dynamics are static and low efficient and present problems when the number of lots (I) and machines (J), change considerably, so it could be said that there are no techniques for a general solution; and even the informatics simulation techniques are difficult to apply due to the very high number of possible solutions to the problem (I^J).

Currently, the great dynamics of the artificial intelligence techniques arise as an alternative to the problem, as they establish newer and better solutions, starting from the already existing solutions, allowing great versatility in the solution of this sort of problems.

As it was expressed above, there are different artificial intelligence techniques, such as: genetic algorithms, algorithms based on ACO, algorithms based on intelligent particles, expert systems, neuron nets, diffused logic, heuristics, and taboo search and its variants. These techniques have been used in the solution of production scheduling problems.

The algorithms based on ACO, are related in literature [17, 18, 19, 20, 21, 22] to the solution of problems NP- complete [23], with variable characteristics in time, combining optimization, routing of communication networks, multi-objective optimization and problems in production scheduling.

Likewise, genetic algorithms have risen as a new alternative for solving problems with production scheduling. They have been used, in this last field to find the sequence of the rules of priority that will allow the optimization of a desired objective. Initially, by means of a priority rule, an initial sequence is generated to schedule the orders in each of the job shops; the order of this sequence is consecutively modified by means of a genetic algorithm. The effectiveness of the new population (new order on the sequence) is evaluated using an evaluation function named fitness, previously defined on the basis of the total *Makespan* and its feasible solutions: Inadmissible, semi-active, active and without delay. Other strategies are based on the use of genetic algorithms, the sequence of rules that allow the optimization of the desired objective.

Generally, in industry, artificial intelligence has led to the solution of a great variety of problems, such as: the optimization of any calculable function, independently of being analytic or digital; solution of the classic problem of a traveller, where the objective is to visit m different cities, connected among themselves, choosing the shortest route in a closed circuit, meaning that the initial and final city should be the same; planning class schedules; planning public transportation; planning airplane landing; loading of job shops; analyses of queues in dynamic programming problems. New heuristics based on the taboo search are applied to the solution of the problems related to the Job shop Scheduling (JSS) [24].

The central objective of this article is the use of ACO algorithms to find a good solution to the JSSP problem in the diverse fields of the industry.

Finally, the results of this analysis let us show how these techniques facilitate the finding of a good solution in the JSSP.

2. METHODOLOGY

For the solution of a Job Shop Scheduling problem (*JSSP*), several techniques related to artificial intelligence have been described. Nevertheless, in this section, a new methodology is proposed, based on ACO as outlined in the introduction.

Step 1. Formally, the Job Shop Scheduling Problem (*JSSP*) can be represented by a chart, with a node for each operation $i \in V$. (a collection of all the tasks) where, θ and f are two nodes representing the beginning and the end of all the tasks. For each two consecutive operations in the same job $(i, j) \in A$ (collection of pairs of all the operations in which its precedence relation is determined by the scheduling technique of each job), there is a directed arch and the operations θ and f are respectively the first and last operations of all the jobs. For each pair of operations using the same machine $\{i, j\} \in E_k$ (collection of operations executed by machine k) there are two arches (i, j) and (j, i) in opposite directions indicating the operation to be done first. Each node i has an associated weight pi indicating the execution time of the operation i . Figure 1 represents the disjointed chart associated with problem of M machines and N jobs ($N \times M$).

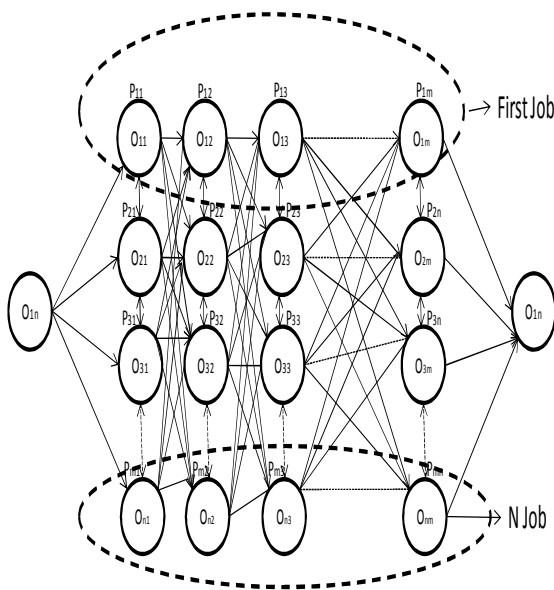


Figure 1. JSSP problem in a chart form

Where O_{ij} , represents job i in the machine j and P_{ij} represents the process time of the job i in the machine j . This graph cannot show all connections. They are represented by a matrix.

The solution of 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 so the resulting chart will not be cyclic and the total length of the longest route between the node θ and the node f will be minimum. If an orientation of arches gives place to a cyclic chart, the corresponding orientation or solution becomes unfeasible.

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

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

Step 3. A colony of L artificial ants are created in such manner that they will move in a concurrent and asynchronous manner, between nodes θ and f through the adjacent states of the problem represented in the form of a chart. The path travelled by each ant is a valid solution to the problem only if it represents the longest route between nodes θ and f . This movement (ants) is made by

following a transition rule based on a higher probability function Q , which is calculated based on the heuristic and pheromone matrix of information; as illustrated in equation 1:

$$Q_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [P_{ij}]^\beta}{\sum_{u \in N_i^k} [\tau_{ij}]^\alpha [P_{ij}]^\beta} \cdot If & S \in N_k(i) \\ 0, & In \ other \ case \end{cases} \quad (1)$$

Where,

k , is the ant of a specific node.

N_i^k , is the neighbourhood reachable by the ant k , when located in node i .

$\alpha, \beta \in \mathfrak{R}$, are two parameters that ponder the relative importance of the traces of pheromone and the heuristic information.

τ_{ij} , is the trace of pheromone between node i and node j . (associated to node).

P_{ij} , the job process times i in machine j .

Once an ant has built a valid solution, each trace of pheromone in the chart between each pair of nodes (arches), is reduced (in the matrix of pheromone) in a constant factor, $\tau_{ij} \leftarrow (1-X)\tau_{ij}$, where X is the rate of vaporization. Subsequently, only the arch of the chart representing the solution found by the ant is 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: $\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^k \quad \forall a_{ij} \in S_k$,

where $\Delta\tau_{ij}^k = f(C(S_k))$, is the deposited amount of pheromone, which depends on the quality $C(S)$ (Makespan) of solution S .

One of the best solutions can be found using the different interactions of the algorithm in the route with most amount of pheromone.

The best solutions are reinforced while the others are penalized through the vaporization process.

Step 4: A diagram of Gant is defined for each of the sequences determined in the previous step, which establishes the order of the processes in time, in each of the different machines. Establishing this diagram, the total process time and the total idle time are calculated under the following fitness functions:

$$Makespan = \min(\max(\max(S_{ij}))) \quad (2)$$

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

Where N represents the number of orders. M , represents the number of machines, S_{ij} is the time of work processing i in machine j and f_j is the total idle time of machine j .

Step 5: It is necessary to repeat the previous process during a determined amount of times (treatments) taking into consideration the five best results of the fitness makespan function in each treatment (equation 2) to guarantee the consistency of the solution. An analysis of variance should be done under the model $y_g = \mu + T_g + \varepsilon_g$, where y_g represents the variable of the answer, T_g , the effect caused by the g^{th} treatment, ε_g , and the g^{th} experimental error to determine if the results correspond to statistically equal or different treatments.

3. EXPERIMENTATION

The possible forms of programming N order in M machines are determined by the following equation: $N!^M$. So it is necessary to properly choose a good field of study, to confirm the value found with its optimal value. Table 1, was drafted to establish a suitable experimentation field. It shows the possible forms of programming N orders in M machines:

Table 1. Sec. N orders in M machines ($N!^M$)

N/M	6	7	8	9	$N!^M$
1	1	1	1	1	.
2	64	128	256	512	.
3	46.656	279.936	1.679.616	10.077.696	.
4	191.102.976	4.586.471.424	1.10075E+11	2.64181E+12	.
.
N

An analysis of table 1 allows the selection of a set of problems in which it is feasible to calculate an ideal solution in a reasonable time. Based on this sample a problem JSSP 3x9 was chosen to be analyzed using this technology and its solution to be compared to the optimal solution. (Table 2).

Table 2. JSSP 3X9.

N/M	M1	M2	M3	M4	M5	M6	M7	M8	M9
P1	5	10	8	10	13	10	8	6	11
P2	10	9	11	7	12	7	11	5	7
P3	13	8	8	16	11	8	9	8	5

The following basic parameters were established using experimental tests: $\alpha = 1$ and $\beta = 2$, and ants = 100, quantity of generations = Z;

4. RESULTS

Step 1. Once the information of the enterprise, and object of this study are analyzed, the JSSP problem of this enterprise is schemed in the chart of figure 2, which illustrates all the operations sequences.

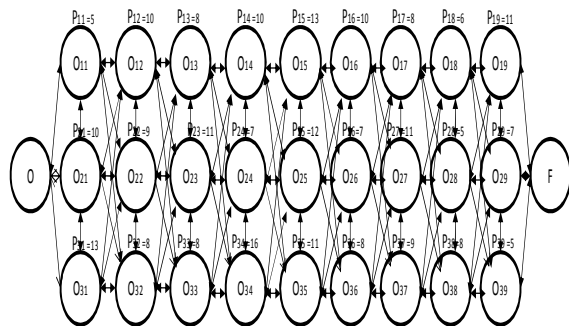


Figure 2. JSSP problem in graphic form

Note: It is not possible to represent all the connections in this graph. They are represented by a matrix.

Possible solutions are represented by the sequence of nodes in a similar manner to the first point proposed in this methodology.

Step 2. Chart of Figure 2, with its respective makespan, is represented in Table 3 and 4 to facilitate the proposed methodology:

Table 3. Heuristic information

	O	O11	O12	O13	O14	O15	O16	O17	O18	O19	O21	O22	O23	O24
O	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O11	0	0	10	8	10	13	10	8	6	11	10	9	11	7
O12	0	5	0	8	10	13	10	8	6	11	10	9	11	7
O13	0	5	10	0	10	13	10	8	6	11	10	9	11	7
O14	0	5	10	8	0	13	10	8	6	11	10	9	11	7
O15	0	5	10	8	10	0	10	8	6	11	10	9	11	7
O16	0	5	10	8	10	13	0	8	6	11	10	9	11	7
O17	0	5	10	8	10	13	10	0	6	11	10	9	11	7
O18	0	5	10	8	10	13	10	8	0	11	10	9	11	7
O19	0	5	10	8	10	13	10	8	6	0	10	9	11	7
O21	0	5	10	8	10	13	10	8	6	11	0	9	11	7
O22	0	5	10	8	10	13	10	8	6	11	10	0	11	7
O23	0	5	10	8	10	13	10	8	6	11	10	9	0	7
O24	0	5	10	8	10	13	10	8	6	11	10	9	11	0
O25	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O26	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O27	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O28	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O29	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O31	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O32	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O33	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O34	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O35	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O36	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O37	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O38	0	5	10	8	10	13	10	8	6	11	10	9	11	7
O39	0	5	10	8	10	13	10	8	6	11	10	9	11	7
F	inf	5	10	8	10	13	10	8	6	11	10	9	11	7

Table 4. Heuristic information

	O25	O26	O27	O28	O29	O31	O32	O33	O34	O35	O36	O37	O38	O39	F
O	12	7	11	5	7	13	8	8	16	11	8	9	8	5	inf
O11	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O12	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O13	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O14	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O15	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O16	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O17	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O18	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O19	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O21	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O22	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O23	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O24	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O25	0	7	11	5	7	13	8	8	16	11	8	9	8	5	0
O26	12	0	11	5	7	13	8	8	16	11	8	9	8	5	0
O27	12	7	0	5	7	13	8	8	16	11	8	9	8	5	0
O28	12	7	11	0	7	13	8	8	16	11	8	9	8	5	0
O29	12	7	11	5	0	13	8	8	16	11	8	9	8	5	0
O31	12	7	11	5	7	0	8	8	16	11	8	9	8	5	0
O32	12	7	11	5	7	13	0	8	16	11	8	9	8	5	0
O33	12	7	11	5	7	13	8	0	16	11	8	9	8	5	0
O34	12	7	11	5	7	13	8	8	0	11	8	9	8	5	0
O35	12	7	11	5	7	13	8	8	16	0	8	9	8	5	0
O36	12	7	11	5	7	13	8	8	16	11	0	9	8	5	0
O37	12	7	11	5	7	13	8	8	16	11	8	0	8	5	0
O38	12	7	11	5	7	13	8	8	16	11	8	9	0	5	0
O39	12	7	11	5	7	13	8	8	16	11	8	9	8	0	0
F	12	7	11	5	7	13	8	8	16	11	8	9	8	5	0

Each row and each column represents a node of the chart in the matrix of Table 3 and 4. If there is an arch between a pair of nodes (O_{ij}, O_{kl}), this is represented by a value equal to

the value of the operation in node O_{ki} . In some other cases, the matrix will have an infinite value when there is no connection. The matrix is built in an analogous way, using the information from the pheromone traces. This matrix will be initially at 0.1, due to the fact that the ants have not built solutions.

Step 3. As each of the ants proposed for this problem, builds a valid solution, the amount of pheromone will be updated along the route of the solution, with a value directly proportional to the quality of the solution, as illustrated in step 3 of the proposed methodology. The best solutions built by the ants in this stage is shown: $O_1, O_{11}, O_{21}, O_{31}, O_{12}, O_{22}, O_{32}, O_{13}, O_{23}, O_{33}, O_{14}, O_{24}, O_{34}, O_{35}, O_{15}, O_{25}, O_{16}, O_{26}, O_{36}, O_{17}, O_{27}, O_{37}, O_{28}, O_{18}, O_{38}, O_{39}, O_{29}, O_{19}$, f. Makespan = 86 and idle = 315. (1th Solution). $O_1, O_{11}, O_{21}, O_{31}, O_{12}, O_{22}, O_{32}, O_{13}, O_{23}, O_{33}, O_{14}, O_{24}, O_{34}, O_{15}, O_{25}, O_{35}, O_{16}, O_{26}, O_{36}, O_{37}, O_{17}, O_{27}, O_{38}, O_{18}, O_{28}, O_{29}, O_{19}, O_{39}$, f. Makespan = 86 and idle = 321.

Step 4. A diagram of Gant is defined for each of the sequences established in the previous step, which determines the order of the processes in time, in the different machines (figures 3 -5). The total process time and the total idle time are calculated by establishing this diagram, (equation 2 y 3).

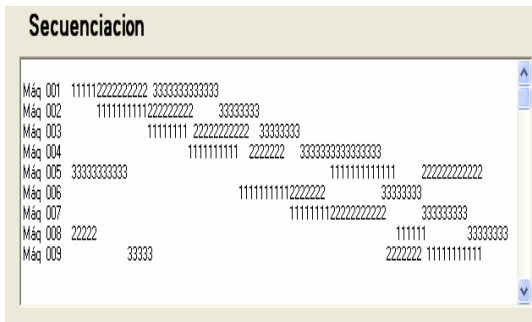


Figure 3. C. Gant. First solution. TP = 86, IT 315

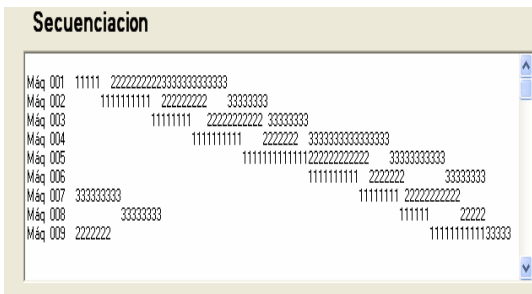


Figure 4. C. Gant. Second solution. TP = 86, IT 321

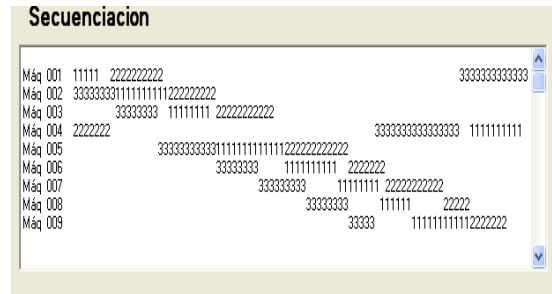


Figure 5. C. Gant. Third solution. TP = 86, IT 322

Step 5. A colony of L ants was taken into consideration for the generation of the previous solution, allowing the evolution of the algorithm until it was not possible to find a better solution. The former process was repeated 10 times (treatments) taking in each treatment, the five best results of the fitness function makespan (Equation 2 and 3):

Table 5. Evaluation of the Idle Time

Treatments	T. Idle	T. Idle	T. Idle	T. Idle	T. Idle
1	336	340	323	315	321
2	335	327	323	327	332
3	340	326	322	315	331
4	315	315	323	328	321
5	340	332	322	323	323
6	332	332	321	321	323
7	336	315	321	327	340
8	323	328	323	315	321
9	321	336	331	323	321
10	328	331	329	327	327

Note: This evaluation was carried out using the idle time, because the algorithm always generates the optimal process time (Time process 86.)

Variance analysis was carried out under the model $\gamma i = \mu + T i + \varepsilon i$, where γi represents the variable of the response $T i$, the effect caused by the i^{th} treatment, εi , and i^{th} the experimental error, to determine if the results of Table 5 correspond to statistically equal or different problems,. The results of this process are summarized in Table 6:

Table 6. Analysis of ANOVA

FV	GL	SC	CM	Fcalc	Ftab(0.05,40,9)
Trat	9	349,22	38,80	0,74	2,12
Err. Expe	40	2098,8	52,47		
Total	49	2448,02			

The results shown in Table 6 indicate that there is a significant level of 99.5% among the ten treatments. It is important to highlight that the information contains the necessary conditions, independence and normality, to apply this test.

Finally, the best solution (figure 6), shows an approximation regarding the ideal solution of 100 % in the variable total time process and of 99.36% regarding the variable idle time.

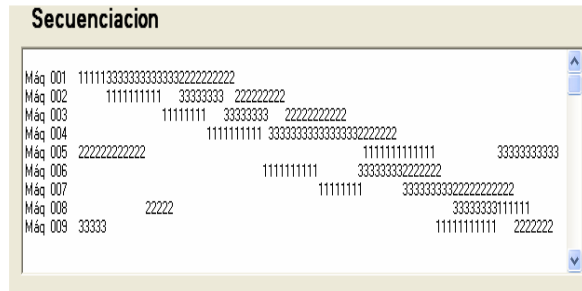


Figure 6. Optimal solution. TP = 86, IT 313

5. CONCLUSION

The ACO constitutes an excellent technique for the solution of the sequential process in JSSP, environments, in which some of the best solutions are found, with a 99% of approximation to optimal solution.

This study seeks to encourage the use of these artificial intelligent technologies in different companies; specially in developing countries, where the production system comprises a great number of manual operations, preventing them from reaching high competitive levels compared to world standards.[25]

It is important, for future studies, to take into consideration any given variation in the makespan and in the idle time; this can point differences in the total process time (makespan), especially when the experiment is repeated several times. These variations will allow the estimation of a job deadline, in a particular interval (d_{i1} , d_{i2}), in which the degree of satisfaction for the job J_i is a decreasing function, according to the delayed function for the job T_i . Excellent results have been produced this way. [26,27].

REFERENCES

- [1] VALENTE, J. M. An analysis of the importance of appropriate tie breaking rules in dispatch heuristics. *Pesqui. Oper.*, 26 - 1,169-180, 2006.
- [2] VAN, F. E. A study of particle swarm optimization particle trajectories. *Information Sciences.*, 176, 937 -971, 2006.
- [3] AKHILESH, K. AND ANUJ, P. Psycho-Clonal algorithm based approach to solve continuous flow shop scheduling problem. *Expert Systems with Applications.*, 31, 504–514, 2006.
- [4] PEZZELLAA, F. AND MORGANTIA, G. A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers & Operations Research.*, 35, 3202 – 3212, 2008.
- [5] FATIH, M. AND LIANG, Y. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research.*, 177-3,1930-1947, 2007.
- [6] LIAN, Z. AND GU, X. A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan. *Applied Mathematics and Computation.*, 175, 773-785, 2006.
- [7] LIAO, C. AND TSENG, C. A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers & Operations Research.*,34- 10, 3099 - 3111, 2007.
- [8] HUANG, K. AND LIAO, C. Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers & Operations Research.*, 35-4,1030-1046-2008.
- [9] MARCO, D. AND STUTZLE, T. *Ant Colony Optimization*, MIT Press, *massachusetts* , 2004.

- [10] YAGMAHAN, B. AND MUTLU, M. Ant colony optimization for multi-objective flow shop scheduling problem. *Computers & Industrial Engineering.*, 54, 411 – 420, 2008.
- [11] DORIGO, M. AND COLORNI, A. The Ant System: Optimization by a Colony of Cooperating agents. *IEEE Transactions on Systems Man, and Cybernetics.*, 26, 29- 41, 2006.
- [12] GAMBARDELLA, M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation.*, 1, 53-66, 1.997.
- [13] VENTRESCA, M. Ant Colony Optimization for Job Shop Scheduling Problem, MISTA Press, Marbella, 2004.
- [14] PURIS, A. AND BELLO, R. Two-Stage ACO to Solve the Job Shop Scheduling Problem. 12th Iberoamerican Congress on Pattern Recognition, CIARP. Valparaiso, Chile, Tomo 1, 447-456, Junio 2007.
- [15] ZANDIEH, M. FATEMI, S. An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation.*, 180, 111–127, 2006.
- [16] YAMASHITA, D. S. AND MORABITO, R. U. algoritmo exato para o problema de programação de projetos com custo de disponibilidade de recursos e múltiplos modos. *Pesqui. Oper.*, 27 -1, 27-49, 2007.
- [17] AZARON, A. AND KATAGIRI, H. Longest path analysis in networks of queues: Dynamic scheduling problems. *European Journal of Operational Research.*, 174, 132–149, 2006.
- [18] DI CARO, G., DORIGO, M. AND GOLDBERG, E. The Ant Colony Optimization Meta-heuristic, *Mcgraw-Hill'S Advanced Topics In Computer Science New ideas in optimization.*, 1, 11-32, 1999.
- [19] DORIGO, M., MANIEZZO, V. AND COLORNI, A. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems Man. and Cybernetics.*, 26, 29-41, 1996.
- [20] GAMBARDELLA, L. AND DORIGO, M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1, 53-66, 1.997.
- [21] HOLGER, H. AND STÜTZLE, T. MAX-MIN Ant System. *Future Generation Computer Systems.*, 16 -8, 889-914, 2000.
- [22] STÜTZLE, T. AND DORIGO, M. The Ant Colony Optimization Metaheuristic: Algorithms, Applications and Advances. In: *Handbook of Metaheuristics*, e. F. Glover and G. Kochenberger, Editors. Kluwer Academic Publishers., 1, 251-285, 2003.
- [23] MICHAEL, R. AND JOHNSON, D. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, H. Freeman and Company Press, San Francisco, 1.979.
- [24] YONG, C. AND LI, P. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Computers & Operations Research.*, 34-11, 3229-3242, 2007.
- [25] ARCE, J., FERNÁNDEZ, D., LLATA, J.R. et al. Aplicación de la inteligencia artificial en sistemas automatizados de producción. *Revista Iberoamericana de Inteligencia Artificial*, 1, 100-110, 2000.
- [26] PETROVIC, S. AND GEIGER, M. J. A Fuzzy scheduling problem with Dynamic Job Priorities and an Extension to Multiple Criteria. *International Conference, Decision Support in an Uncertain and Complex World: The IFIP TC8/WG8.3.*, Athens, Grecia, Tomo 1, 637-646, Junio 2003.
- [27] LI, S. G. AND RONG, Y. L. The reliable design of one-piece flow production system using fuzzy ant colony optimization. *Computers & Operations Research.*, 36, 1656 – 1663, 2009.