Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Ant Colony Optimization Algorithm for Reactive Production Scheduling Problem in the Job Shop System

E. R. R. Kato, O. Morandin, Jr., M. A. S. Fonseca
Department of Computer Science
Federal University of São Carlos (UFSCar)
São Carlos, SP, Brazil

kato@dc.ufscar.br; orides@dc.ufscar.br; marcos_abraao@dc.ufscar.br

***Abstract*—The response time for solution scheduling problem is a import criteria of consider in real manufacturing systems where large-scale scenarios must be evaluated since as unexpected events arise. This work describes a proposed modeling and analyses for the production reactive scheduling problem in a job shop system. The scheduling problem, generally, consists in allocate the production operations with the aim of minimizing the makespan. For that, it was employed an Ant Colony Optimization algorithm applied in a matrix of the feasible solution space problem representation. In the case of a reactive system, the approach should provide good solutions in a short execution time, allowing the analysis of large scenarios in habile times. The results of this paper were compared with the results of other approaches in small and large scenarios.**

***Keywords*—ACO, graph representation, job shop scheduling problem, reactive scheduling.**

I. INTRODUCTION

Among the main components that compound the performance of an enterprise such as quality, speed, reliability, flexibility and cost, one of the factors more emerging in productive systems is the flexibility that the system can provide in terms of switching of products and processing of its operations. In order to obtain a competitive advantage, the flexibility of job shop system considers it, since trying produce according to customer expectations and requirements of.

However, since this degree of flexibility rises, production systems become increasingly complex and difficult to solve for professionals in the planning and control of production, which have motivated various research work in both fields. In this context, focusing on the area of planning, the commonly problems dealt in the literature are found to the scheduling and scheduling production.

In real manufacturing systems, the production is subject to failures and some events as machines breaking, missing resources make the system unworkable. This aspect must be considered, what defines a reactive system, and a re-scheduling or re-sequencing production becomes necessary in a short time allowing the system resumption. The job shop scheduling and sequencing production is the class of NP-Hard problem, justifying the use of various types of heuristics and artificial intelligence (AI) techniques with the objective of finding near-optimal solutions. In literature various studies have been proposed using different methods like Genetic Algorithms (GA) [1], hybrid algorithms with GA and Local Search [2] [3], other heuristics [4] and recently Ant Colony Optimization for combinatorial optimization problems [5].

In [6] and [7] are presented two approaches to the problem of flexible job shop scheduling (FJSP). Both deal with the problem upon two perspectives, according to the model and according to the method of search for solution of the problem. [6] provide a description of the problem in disjunctive graphs, which include the restrictions considered in the problem as the setup time, transportation time, due date and processing time. In the aspect of finding solution is used a meta-heuristic of ant colony optimization (ACO). The results, achieved by applying the method by using benchmarks, shows that the approach is efficient when compared to other heuristics and techniques such as Genetic Algorithms and Tabu Search (TS).

Liouane *et al* [7] use a graph model which it is derived from a matrix representation. The method, based in ACO for finding the solution, is applied to matrix that represents the solution space. After construct the solution by the ACO algorithm, a local search method is applied to improve the solution. According to conclusions of the authors, the approach gives efficient results for the experiments evaluated.

Benbouzid *et al* [8] apply the ACO approach to another class of manufacturing systems. The algorithm is used to determine the sequencing of production in a flow shop system, which consists of a permutation of tasks to be allocated on the machines such that minimize the total production time (or makespan). The construction of the path traveled by the ants determines the solution to the sequence production problem, which can be further enhanced by a local search based on insertion movements. To the problem, is still considered the preventive maintenance which occurs in a given interval of time. The experimental results show that the proposed approach is better than other approach that use GA algorithm. However, when inserting the preventive maintenance in the solution, the results are lower than the approach with GA. According to Ruiz, R. *et al* [9] many studies show this feature, to consider the preventive maintenance on a disjoint in the

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

solution of the problem, in such a way entering the tasks of maintenance after the choice of scheduling operations. In the work, [9] analyze the behavior of different approaches to the integration of preventive maintenance with the solution of the problem. The results show that the GA and ACO approaches obtained the best solutions for the scheduling production jointly preventive maintenance.

Based on research described previously, this work presents the modeling and analysis of a specific manufacturing system, included in the class of job shop, with the aim deal the reactive production scheduling problem, minimizing the makespan value in short time to solution response became in this case the performance a critical factor considered in this work. The problem is formally described in session 2 as well as with the restrictions considered. Session 3 describes the modeling used to represent the problem by making it accessible to the mapping and analysis computational. The ACO approach proposed in this work is described in Session 4. Session 5 presents the results achieved with the approach. And finally, Session 6 presents the main conclusions and considerations on the results achieved.

II. PROBLEM FORMULATION

The job shop may be described as follows. Consider a set of N jobs $J_1, J_2... J_N$, which must be processing by K machines $M_1, M_2... M_K$. Each job J_i consists of a sequence of operations defined by O_{ij} representing the execution order. The processing times O_{ij} of job J_i in machine M_k are pre-defined and represented by P_{ijk} , where i is the index of the job, j is the operation index and k the index that indicates the machine has been allocated to operation. Table I, illustrates an example of this description considering $i = 2$ $k = 3$.

TABLE I. EXEMPLE OF PROCESSING TIME OPERATIONS

		M_1	M_2	M_3
J_1	O_{11}	1.5	0.9	1.1
	O_{12}	2.2	2.1	1.4
J_2	O_{21}	0.1	2.3	1.3
	O_{22}	1.7	1.85	1.5

Each operation O_{ij} has a processing start time, denoted as T_{ij} and a processing completion time TF_{ij} . ST_{ij} represents the setup time for machine k to perform operation O_{ij}. The following restrictions are also considered for the problem:

- Each machine performs only one operation at a time.
- The operations are not preemptable.
- All machines are available at time $t = 0$.
- The machines do not break.
- The transportation time of a resource for the machine M_k is not considered.
- The setup time is the same for all machines.
- The execution order of operations of each job is fixed and cannot be changed.

The problem is to allocate all operations to the machine which consequently allows us to determine the sequence of execution of each job according to an objective function. The objective function used in this work is the makespan, denoted by:

$$C_{Max} = TF_{ij} \tag{1}$$

where C_{Max} denotes the value of the last operation i of last job j. TF_{ij} is processing completion time.

III. PROBLEM MODELLING

The problem representation follows the characteristics inherent of the ACO meta-heuristic. Thus, the solution of the scheduling production problem, which is consequent the path built by artificial ants, denotes the order of products and their respective routes of production. A manufacturing route is a sequence of operations that must be followed in a defined order and production can be defined by J_iR_k where J_i is the i-th job and R_k the k-th route of the job J_i. Information on the routes are known and previously acquired as illustrated in Table II.

TABLE II. EXEMPLE OF ROUTES OPERATIONS OF THE JOBS J_1 J_2 J_3

J_1	M1	M2	M3	-
	M2	M3	-	-
J_2	M2	M3	M4	-
	M1	M2	M3	M4
J_3	M1	M2	M4	-

In this example the jobs J_1 and J_2 has each of two different routes and only one task J_3. As the table indicates, the job J_1 is complete after the execution of its operations on the machines *M1*, *M2* and *M3* or *M2* and *M3*. From this information, the matrix of Table III is constructed, where each value shown represents the index of the machine in the route. For example, the values 1, 2 and 3 of the job J_1 represents the machines *M1*, *M2* and *M3* in Table II and value 0 indicates the absence of other machines for this route. This value 0 is necessary because the number of elements in the route of the job J_2 in his second route, since a matrix must respect the values of their dimensions.

TABLE III. REPRESENTATION OF PRODUCTION ROUTE OF JOBS J_1 J_2 J_3

J_1	1	2	3	0
	2	3	0	0
J_2	2	3	4	0
	1	2	3	4
J_3	1	2	4	0

According to the considerations on the routes listed, the space of solutions to the problem can be represented by a graph G = (V, A), where V denotes the set of vertices whose values indicate the pair J_iR_k defined above and A the set of edges connecting the vertices for the construction of feasible solutions. A feasible solution indicates that a sequence of vertices J_1R_1, J_1R_2 and J_2R_1, as in Figure 1 is not permitted, since the solution is a permutation of jobs (i.e. the same job J_i

must not contain more a time in sequence). For each edge is associated a value that corresponds to the weight to go from node i to node $i+1$. The weight is given by (1), i. e. the value of total production time of job J_i with the route R_k for the job J_t with the route R_s. The Figure 2 represents the values associated to each edge and also indicates a plausible solution given by J_1R_2, J_2R_1 and J_3R_1.

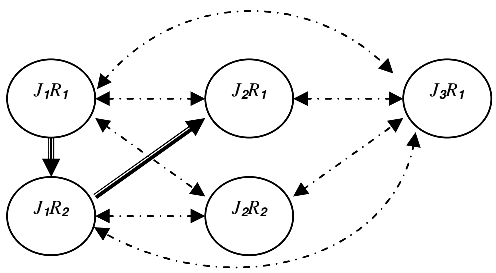
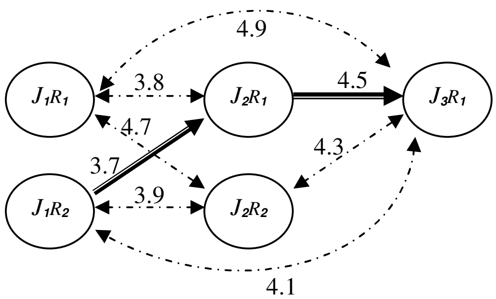


Figure 1. Incorrect jobs sequence



.

Figure 2. Graph for problem representation.

IV. The Proposed Ant Colony Optimisation

The ACO meta-heuristic was established from observations of the behavior of real ants in a colony. Its main feature is that comes from communication between the ants via a substance called pheromone. Insofar as the ants will explore the environment through the identification of food, the pheromone is deposited on the traveled path, influencing in turn the other ants to follow the same path. Over time the convergence of the path becomes clear, indicating to light a path which in turn, is the shortest path between the nest and food. The fundamental work was developed by [10] and then became a meta-heuristic in the work presented in [11]. The following flowchart (Figure 3) illustrates the general flow of the algorithm outlining its main points [5].

Initially the ants are randomly distributed through the space of solutions, represented by Figure 2 in order to build the individual solution to the problem. The number of ants is informed as a parameter to the algorithm and can change according to the modeling and the problem treated. The probability of choosing the next element that composes the solution of the problem is given by the following (2).

$$p_{ij}^{k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in N_i^k} (\tau_{il})^{\alpha}(\eta_{il})^{\beta}} \text{ if } j \in N_i^k, \qquad (2)$$

where $(\tau_{ij})^{\alpha}$ is the amount of pheromone deposited on the edge (i, j). α is the parameter that determines how influential will be the value of pheromone. (η_{ij}) is the heuristic value given by $1/d_{ij}$, where d_{ij} is the value obtained in (1). β is the parameter that determines how influential will be the value of the heuristic used. N_i^k is the feasible neighborhood of ant k (i. e. the set of nodes not yet explored by ant k).

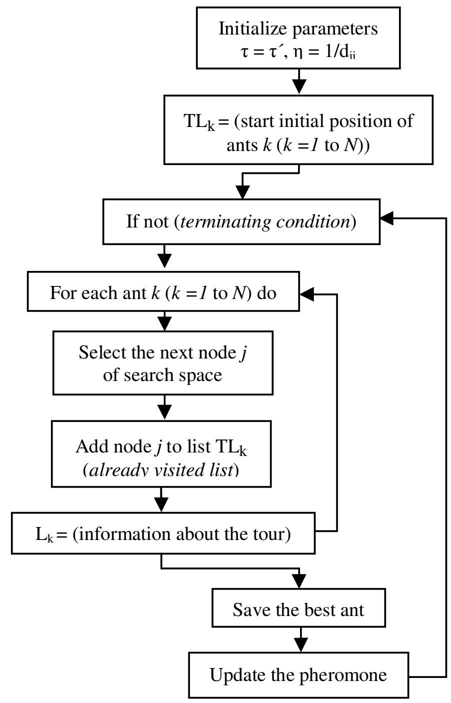


Figure 3. Flow chart of Ant Colony Optimisation.

The construction of the solution is based on the choice of the nodes following a number of interactions give by the solution representation. The information such as nodes already visited and sum of the objective function values are save in artificial ants to each interaction. After the construction of possible solutions, given by each of the ants, the value obtained by the best ant, that is the best solution, is save. The trail is updated by the following expression, proposed in [12]:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{\mu=1}^{\sigma-1} (\sigma - \mu)\frac{Q}{L_{\mu}} + \sigma \frac{Q}{L_{bs}} \qquad (3)$$

where $(1-\rho)\,\tau_{ij}$ define the evaporation of the trail (i, j), and the parameter ρ $[0 < \rho \leq 1]$ is the evaporation rate applied to the value of pheromone τ_{ij}. σ is the number of elite ants and μ ranking of the μ-th ant elite. Q/L_{μ} represents the pheromone update on the edge belonging to the path of σ elite ants. Q is a constant and Q/L_{bs} represent the pheromone update on the edge belonging to the path of the best ant (best-so-far ant). The algorithm follows the flow shown in Figure 3 until the number of cycles, defined as a parameter, is reached. After the second interaction the pheromone values begin to act on the edges of the search space and, consequently, the choice of the nodes that comprise the solution. In order to avoid stagnation, (i. e. all ants converge to the same route) a process of evaporation is first applied by reducing the values of the pheromone trails and then the deposit of pheromone is employed.

A. Mapping of the Representation

The mapping of the representation corresponds to a computational transcript of the model presented in Session 3 jointly with the restrictions of the problem defined in Session 2.

As mentioned, each node of the representative model corresponds of juxtaposed pairs of jobs/routes J_iR_k. The construction of a feasible solution is to establish a coherent order of nodes such that it minimizes the total time of production in a habile execution time. From this information and the problem restrictions, the matrix of Table IV can be defined in order to map the details of the model.

TABLE IV. MAPPING CONSIDERING MODEL AND PROBLEM RESTRICTION

	J_1R_1	J_1R_2	J_2R_1	J_2R_2	J_3R_1
J_1R_1	0	0	1	1	1
J_1R_2	0	0	1	1	1
J_2R_1	1	1	0	0	1
J_2R_2	1	1	0	0	1
J_3R_1	1	1	1	1	0

The matrix consists in N +1 lines by N +1 columns whose values of the first row and N columns indicate the N pairs of job/route obtained by the representation illustrated in Figure 2. Similarly, the values of the first column with the N lines also indicate the N pairs of job/route. The zero values of the matrix lead to a consistency restriction with the problem solution which is not necessary to calculate the makespan for identical jobs but with different routes. Thus, given the matrix of table IV, the calculation of weight to the edges of Figure 2 can be made from these considerations. Figure 5 illustrates this principle.

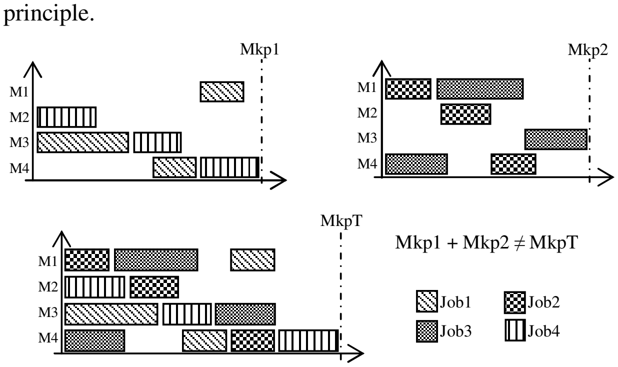


Figure 4. Makespan value considerations

Once defined the values for the weights of the edges, the search method can be applied on the solutions space matrix by the approach described in the following session.

	J_1R_1	J_1R_2	J_2R_1	J_2R_2	J_3R_1
J_1R_1	0	0	1	1	1
J_1R_2	0	0	1	1	1
J_2R_1	1	1	0	0	1
J_2R_2	1	1	0	0	1
J_3R_1	1	1	1	1	0

calculation of makespan

	J_1R_1	J_1R_2	J_2R_1	J_2R_2	J_3R_1
J_1R_1	0	0	3.8	4.7	4.9
J_1R_2	0	0	3.7	3.9	4.1
J_2R_1	3.8	3.7	0	0	4.5
J_2R_2	4.7	3.9	0	0	4.3
J_3R_1	4.9	4.1	4.5	4.3	0

Figure 5. Matrix representing the weights of the edges.

B. ACO for production reative scheduling problem with flexible routes

The algorithm proposed in this paper follows the structure presented in Session 4. The coding of the algorithm was developed using the resources of the software Matlab (The MathWorks, Inc.). The main reasons for choosing the tool is in its robustness, and offer ease of handling of matrices and vectors.

Initially the ants are distributed taking as beginning point the first column and one of the lines of the search space matrix. The choice of the next node, given by (2), keeps on iteratively until all the ants complete the journey. After the construction of the solutions, given by each ant individually, the values of the pheromone matrix are updated using the expression (3). These procedures are performed until the specified number of cycles is reached as shown in the flow of the algorithm in Figure 3.

Among the information saved for each ant, are the concatenation of the values of the vertices of the path traveled and the sum of the makespan value. This sum of the makespan values does not match with the actual time of production since the calculation is done considering only one pair of jobs. Thus, after building the solution the calculation of makespan is repeated considering all jobs to be performed in the production system given for the sequence save in ants (see Figure 4).

The following session describes the parameters used in the algorithm for the tests as well as some important considerations about the solution prepared by each ant.

V. EXPERIMENTAL RESULTS

As a way of evaluating the proposal, this approach was compared to another approach that uses the genetic algorithms (GA) technique for production reactive scheduling problem [13]. For each scenario were performed 35 tests (i. e. 35 runs for the same problem) on a Core 2 Duo 3.0 GHz with 2GB of RAM and Windows XP operating system.

The scenarios proposed in this work corresponds a two instances of problem, one with three jobs and six machines and another instance with nine jobs and nine machines. In the first instance of the problem exists at least one possible rote for each job with three to five machines each as illustrated in Table V. In the second instance, each job has two possible routes with five to seven machines each (see Table VI). Both scenarios were generate randomly

TABLE V. PRODUCTION ROUTES FOR FIRST INSTANCE

Jobs		Routes
J_1	R_{11}	$M_1\,M_2\,M_3\,M_4\,M_5$
	R_{12}	$M_1\,M_2\,M_3\,M_6$
J_2	R_{21}	$M_1\,M_4\,M_5\,M_6$
	R_{22}	$M_2\,M_4\,M_5\,M_6$
	R_{23}	$M_3\,M_4\,M_5\,M_6$
J_3	R_{31}	$M_1\,M_5\,M_6$
	R_{32}	$M_2\,M_5\,M_6$
	R_{33}	$M_3\,M_4\,M_5\,M_6$

The products routes were randomly generated, as well as operation times, which ranged between 400 and 500 time units (TU).

TABLE VI. PRODUCTION ROUTES FOR SECOND INSTANCE

Jobs		Routes
J_1	R_{11}	$M_1\,M_2\,M_4\,M_5\,M_7\,M_9$
	R_{12}	$M_3\,M_4\,M_5\,M_6\,M_8\,M_9$
J_2	R_{21}	$M_1\,M_2\,M_3\,M_4\,M_5\,M_6\,M_7$
	R_{22}	$M_2\,M_3\,M_5\,M_7\,M_8\,M_9$
J_3	R_{31}	$M_4\,M_5\,M_6\,M_7\,M_8$
	R_{32}	$M_2\,M_3\,M_7\,M_8\,M_9$
J_4	R_{41}	$M_2\,M_3\,M_4\,M_5\,M_6\,M_7$
	R_{42}	$M_1\,M_5\,M_6\,M_8\,M_9$
J_5	R_{51}	$M_4\,M_5\,M_6\,M_8\,M_9$
	R_{52}	$M_1\,M_2\,M_3\,M_5\,M_6$
J_6	R_{61}	$M_2\,M_4\,M_5\,M_6\,M_7\,M_8\,M_9$
	R_{62}	$M_1\,M_3\,M_6\,M_7\,M_8\,M_9$
J_7	R_{71}	$M_1\,M_2\,M_4\,M_5\,M_6\,M_9$
	R_{72}	$M_1\,M_2\,M_3\,M_7\,M_8\,M_9$
J_8	R_{81}	$M_4\,M_5\,M_6\,M_7\,M_8\,M_9$
	R_{82}	$M_3\,M_4\,M_5\,M_7\,M_8\,M_9$
J_9	R_{91}	$M_3\,M_5\,M_6\,M_7\,M_8\,M_9$
	R_{92}	$M_2\,M_4\,M_6\,M_7\,M_8\,M_9$

For the tests with the GA approach, the following values were used for the parameters of the genetic algorithm: size of the population = 30, crossing rate = 0,8 (80%) mutation rate = 0,05 (5%) and generations = 100. These values were obtained in [13]. The 35 tests for the second instance can be seen at Table VII. The first column indicates the test number (from 1 to 35). The next columns represent the obtained makespan value by proposal and by GA. The makespan were counted in time units (TU), and the response obtaining time calculated in seconds for each test, respectively.

TABLE VII. OBTAINED RESULTS

#	Proposal Makespan	Proposal Time	GA Makespan	GA Time
1	6051	2.15	4686	3.25
2	5809	1.95	5188	3.30
3	5612	2.01	4676	3.14
4	5914	1.98	5122	3.27
5	5770	2.02	5363	3.33
6	6268	1.97	4676	3.13
7	5760	1.99	5082	3.23
8	5809	2.01	4673	3.14
9	5848	1.98	4962	3.23
10	5760	2.05	5096	3.30
11	5888	2.07	4673	3.08
12	5821	1.99	5123	3.14
13	5760	2.00	5123	3.33
14	6010	1.98	4780	3.18
15	6051	1.99	5115	3.25
16	6096	2.04	5542	3.25
17	6026	2.01	5057	3.34
18	5591	2.04	4924	3.17
19	5613	2.05	5694	3.24
20	6179	1.97	4902	3.13
21	5693	2.00	5172	3.25
22	4613	2.05	4768	3.27
23	6236	2.01	4901	3.23
24	5821	1.99	5120	3.39
25	5878	1.98	5224	3.26
26	6056	2.09	5181	3.18
27	5616	1.98	5006	3.13
28	6005	2.06	5432	3.26
29	5952	1.98	5053	3.36
30	5788	2.06	5042	3.20
31	5760	1.97	5122	3.20
32	5632	2.01	5096	3.22
33	5850	2.05	5115	3.19
34	5856	2.03	4901	3.19
35	5914	1.97	4930	3.27
Average	**5837**	**2,08**	**5043**	**3,23**

The Table VIII joins the results for both scenarios. The first column indicates the scenario length and the next columns represent the approaches with their respective results and the response obtaining time calculated in seconds. For the ACO approach, the parameters used were: $N = 15$ (size of ants), $\alpha = 5$ and $\beta = 0.7$ (pheromone and heuristic influence), $\rho = 0.07$ (evaporation rate) and $\sigma = 3$ (elite number). The results of Table VIII shows that the ACO were not better than the GA consider the average of makespan value however, the average of execution time of the proposed method was 2.08 seconds for the large scenario. Other evaluation consisted of new parameters configuration of the ACO approach.

TABLE VIII. OBTAINED RESULTS BY THE ACO AND GA

Scenario	Proposal	Makespan	Time
3x6	GA	2202	0.49
	ACO	2198	0.28
9x9	GA	5043	3.23
	ACO	5837	2.08

The Table IX demonstrates other parameters evaluated. The parameters used were: $N = 20$ (size of ants), $\alpha = 1$ and $\beta = 0.7$ (pheromone and heuristic influence), $\rho = 0.1$ (evaporation rate) and $\sigma = 3$ (elite number). With this results it was noted that establishing a low influence of pheromone with a rapid evaporation, the results tend to improve and approaching the results obtained by GA.

TABLE IX. OBTAINED RESULTS BY THE ACO AND GA

Scenario	Proposal	Makespan	Time
3x6	GA	2202	0.49
	ACO	2198	0.28
9x9	GA	5043	3.29
	ACO	5659	1.54

Although the proposal only approximate of results obtained of the GA approach, one of the objectives of this work could be achieved by reducing the response time for large-scale scenarios which is an advantage in real problems.

VI. CONCLUSION

This paper proposes an Ant Colony Optimization (ACO) algorithm for reactive production scheduling problem in a job shop system with flexible routes. The problem is dealt on two perspectives, on the modeling and to the method of finding the solution. Although the proposed approach models the problem in a comprehensive manner so as to facilitate the computational analysis, the results only were close of objective function value obtained by other approach. The makespan value and solution response time were compared with results of other method that used Genetic Algorithm (GA) technique for production reactive scheduling. Both approaches consider a production reactive system, which leads to a short time to get the results, putting the response time as a critical factor. In this case the ACO approach was better than GA since the response obtaining time was short for large scale scenarios, reflecting real cases in manufacturing system.

One suggestion for improving the makespan value results consists of applying a local search method for the solution found by the ACO algorithm. As shown by the work of [7] and [8] this combination presents significant improvements. However, taking care to the system not lose performance in response time.

Another suggestion would be on the modeling of the proposed heuristic, so as to establish besides a partial evaluated (i. e. makespan between two jobs), a vision of the problem as a whole. Thus, the classification of the routes would be found on the improved real of problem solution and not only on the sum of makespan between two jobs.

REFERENCES

[1] Morandin, O. Jr., Kato, E.R.R., Deriz, A.C., and Sanches, D.S. “An Adaptive Genetic Algorithm Based Approach for Production Reactive Scheduling of Manufacturing Systems” In: *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*,2008. p. 1461-1466.

[2] Chen, Y.-W.; Lu, Y.-Z.; Yang, G.-K. Hybrid Evolutionary Algorithm with Marriage of Genetic Algorithm and Extremal Optimization for Production Scheduling. *The International Journal of Advanced Manufacturing Technology.* vol. 36, 959-958. DOI-10.1007/s00170-006-0904-9, Springer, London, April, 2008.

[3] Gonçalves, J. F., Magalhães Mendes J. J., and Resende, M. G. C. A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem. *European Journal of Operational Research*, vol. 167, 2002.

[4] Chen, J.-W. Sequencing Heuristic for Scheduling Jobs with Dependent Setups in a Manufacturing System. *The International Journal of Advanced Manufacturing Technology.* vol. 38, 176-184. DOI-10.1007/s00170-007-1070-4, Springer, London, 2008.

[5] Colorni, A., Dorigo, M., Maffioli, F., Maniezzo, V., Righini, G. and Trubian M. "Heuristics from Nature for Hard Combinatorial Optimization Problems", *International Transactions in Operational Research*, 1996. vol. 3, Pag. 1-21

[6] Rossi, A. and Dini, G. “Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method” *Robot. Comput.-Integr. Manuf.* 23, 5 (Oct. 2007), 503-516.

[7] Liouane, N.; Saad, I.; Hammadi, S.; Borne, P. "Ant systems & Local Search Optimization for flexible Job Shop Scheduling Production" in: *International Journal of Computers, Communications & Control Vol. II*, No. 2, pp. 174-184, 2007.

[8] Benbouzid S. F.; Ammi, I.; Varnier, C.; Zerhouni, N. “Applying Ant Colony Optimization for the Joint Production and Preventive Maintenance Scheduling Problem in the Flowshop Sequencing Problem.” *2008 3rd ICTTA*, Apr 7-11 2008.

[9] Ruiz, R.; Garcia-Diaz, J. C.; Maroto, C. “Considering Scheduling and Preventive Maintenance in the Flowshop Sequencing Problem” *Computers and Operations Research.* v 34, p 3314-3330, Elsevier Ltd, Oxford, OX5 1GB, United Kingdom, November, 2007. DOI= http://dx.doi.org/10.1016/j.cor.2005.12.007

[10] Dorigo, M. (1992) “Optimization, learning and natural algorithms”, PhD thesis, Dipartiment di Elettronica, Politecnico di Milano, Italy, 1992.

[11] Dorigo, M. and Dicaro, G. (1999) “Ant Colony Optimisation : A new metaheuristic”, *In Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on , 1999.* Volume: 2, p. 1470-1477.

[12] Bullnheimer, B.; Hartl R. F.; and Strauss, C. (1997) “A New Rank Based Version of the Ant System: A Computational Study”, Technical report, Institute of Management Science, University of Vienna, Austria. Later published in *Central European Journal for Operations Research and Economics*, 7(1):25-38.

[13] Morandin, O. Jr., Kato, E.R.R., Deriz, A.C., and Sanches, D.S. “A Search Method using Genetic Algorithm for Production Reactive Scheduling Manufacturing Systems”. *IEEE International Symposium on Industrial Electronics*, Cambridge, UK, June, 2008, in press.