International Journal of Computers, Communications & Control
Vol. II (2007), No. 2, pp. 174-184

Ant systems & Local Search Optimization for flexible Job Shop Scheduling Production

Noureddine Liouane, Ihsen Saad, Slim Hammadi, Pierre Borne

Abstract: The problem of efficiently scheduling production jobs on several machines is an important consideration when attempting to make effective use of a multi-machines system such as a flexible job shop scheduling production system (FJSP). In most of its practical formulations, the FJSP is known to be NP-hard [8][9], so exact solution methods are unfeasible for most problem instances and heuristic approaches must therefore be employed to find good solutions with reasonable search time.
In this paper, two closely related approaches to the resolution of the flexible job shop scheduling production system are described. These approaches combine the Ant system optimisation meta-heuristic (AS) with local search methods, including tabu search. The efficiency of the developed method is compared with others.

Keywords: Flexible production, Ant colony, Tabu search, job shop scheduling, makespan, optimisation.

1 Introduction

Modern hybrid heuristics are by their nature non-exhaustive, and so there is often scope for different approaches to better previous solution methods according to the execution speed or the quality of feasible solutions. Traditional approaches to resolve the FJSP are as varied as the different formulations of the problem, but include fast, simple heuristics [2][12], tabu search [15], evolutionary approaches [5] and modern hybrid meta-heuristics that consolidate the advantages of various different approaches [1][13].

The ant colony optimisation (ACO) was described by Dorigo in his PhD thesis [6] and was inspired by the ability and the organisation of real ant colony using external chemical *pheromone trails* acting as a means of communication.

Ant system algorithms have since been widely employed on the NP-hard combinatorial optimisation problems including problems related to Continuous Design Spaces research [4], and job shop scheduling [16]. However, they have not previously been applied to the FJSP described in what follows.

Local search methods encompass many optimisation approaches and have been shown that the efficiency of their use with an ant system approach [7].

The approach described in this paper for the FJSP shows the quality of solutions found, using benchmark problems. The performances of the proposed approach are evaluated and compared with the results obtained from other methods.

In this paper, an application of the ant system algorithms combined by the tabu search heuristic is proposed for solving the FJSP. Thus, The FJSP is described and formulated in section 2. Then, in section 3, The suggested approach by ACO with the tabu search is described. An illustrative example is given in section 4. The last section will be devoted to the presentation of some results and some conclusions relating to this research work.

2 Problem formulation

The FJSP may be formulated as follows. Consider a set of n independent jobs, noted $\Im = \{J_1, J_2, ..., J_n, 1 \leq j \leq J\}$, which are carried out by K machines M_k, $M = \{M_1, M_2, ..., M_k, 1 \leq k \leq K\}$. Each job J_j consists of a sequence of n_j operations $O_{i,j}$,

$i = 1, 2, ...n_j$. Each routing has to be performed to achieve a job. The execution of each operation i of a job J_j requires one ressource selected from a set of available machines. The assignment of the operation $O_{i,j}$ to the machine $M_k \subseteq M$ entails the occupation of the latter one during a processing time, noted $p_{i,j,k}$. The problem is thus to both determine an assignment scheme and a sequence of the operations on all machines that minimize some criteria.

- A set of J independent jobs.
- Each job is characterized by the earliest starting time r_j and the latest finishing time d_j.
- Denote by $pt_{i,j}$ and $r_{i,j}$ respectively the processing time and the ready date of the operation $O_{i,j}$. The $p_{i,j,k}$ represent the processing time $pt_{i,j}$ with the machine M_k.
- A started operation can not be interrupted.
- Each machine can not perform more than one operation at the same time.
- The objective is to find an operation ordering set satisfying a cost function under problem constraints. In this paper, the considered objective is to minimize the makespan C_{max}.

3 ACO and Tabu search for FJSP Scheduling

In this stage, the application of the combined ant systems with tabu search techniques in the resolution of FJSP problem are described.

3.1 Construction Graph and Constraints

Generally, the FJSP can be represented by a bipartite graph with two categories of nodes: $O_{i,j}$ and M_k. A task is mapped to a $O_{i,j}$ node; a machine is mapped to a M_k. There is an edge between the $O_{i,j}$ node and the M_k node if and only if the corresponding task can be assigned to the corresponding machine while respecting the availability of the machine and the precedence constraints among the operations of different jobs. The cost of assignment is directly related to the processing time of the task upon the machine.

To model the process in a more straightforward manner, we use the construction graph that is derived from the utilization matrix. Below is a sample construction graph.

Table 1: Construction graph of 4 machines and 7 tasks.

	M_1	M_2	M_3	M_4
$O_{1,1}$	10	7	6	13
$O_{2,1}$	4	5	8	12
$O_{3,1}$	9	5	6	12
$O_{1,2}$	15	12	8	6
$O_{2,2}$	9	5	7	13
$O_{1,3}$	7	16	5	11
$O_{2,3}$	9	16	8	11

With this construction graph, we can transform the FJSP into a traveling ant problem. Specifically, given the representative table of n rows and m columns, and each of its cells is associated with $p_{i,j,k}$, representing this one distance among $O_{i,j}$ and M_k. An ant seeks to travel across the table in such a way that all of the following constraints will be satisfied: one and only one cell is visited for each of the rows. In the rest of this paper, "tour" and "solution" are used interchangeably; a pair of (operation, machine) means: operation is assigned to machine, table 2.

Table 2: Solution of Construction graph table 1

	M_1	M_2	M_3	M_4
$O_{1,1}$			6	
$O_{2,1}$	4			
$O_{3,1}$		5		
$O_{1,2}$				6
$O_{2,2}$		5		
$O_{1,3}$			5	
$O_{2,3}$			8	

3.2 Ant systems scheduling

The Ant system approach was inspired by the behaviour of the real ants. The ants depose the chemical *pheromone* when they move in their environment, they are also able to detect and to follow pheromone trails.

In our case, the pheromone trail describes how the ant systems build the solution of the FJSP problem. The probability of choosing a branch at a certain time depends on the total amount of pheromone on the branch, which in turn is proportional to the number of ants that used the branch until that time. The probability P^f_{ijk} that an ant will assign an operation $O_{i,j}$ of job J_j to an available machine M_k. Each of the ants builds a solution using a combination of the information provided by the pheromone trail τ_{ijk} and by the heuristic function defined by $\eta_{ijk} = p_{i,j,k}$.

Formally, the probability of picking that an ant f^{th} will assign an operation $O_{i,j}$ of job J_j to the machine M_k is given in equation 1.

$$P^f_{ijk} = \begin{cases} \frac{(\tau_{ijk})^\alpha * (\eta_{ijk})^{-\beta}}{\sum\limits_{l \in D} (\tau_{ijk})^\alpha * (\eta_{ijk})^{-\beta}} & if\ j \in D \\ 0 & if\ j \notin D \end{cases} \tag{1}$$

In this equation, D denotes the set of available non-executed operations set and where α and β are parameters that control the relative importance of trail versus visibility. Therefore the transition probability is a trade-off between visibility and trail intensity at the given time.

3.3 Updating the pheromone trail

To allow the ants to share information about good solutions, the updating of the pheromone trail must be established. After each iteration of the ant systems algorithm, equation 2 describes in detail the pheromone update used when all ants have completed an own scheduling solution denote L^{ants}, that represent the length of ant tour. In order to guide the ant systems towards good solutions, a mechanism is required to assess the quality of the best solution. The obvious choice would be to use the best makespan $L^{min} = C_{max}$ of all solutions given by a set of ant.

$$\Delta\tau^f_{ijk} = \begin{cases} \frac{L^{\min}}{L^{Ants}} & if\ i,j,k \in L^{Ants} \\ 0 & otherwise \end{cases} \tag{2}$$

After all of the ants have completed their tours, the trail levels on all of the arcs need to be updated. The evaporation factor ρ ensures that pheromone is not accumulated infinitely and denotes the proportion of ŚoldŠ pheromone that is carried over to the next iteration of the algorithm. Then for each edge the pheromone deposited by each ant that used this edge are added up, resulting in the following pheromone-level-update equation:

$$\tau_{ijk} = \rho.\tau_{ijk} + \sum_{f=1}^{N^{BA}} \Delta\tau_{ijk}^{f} \tag{3}$$

where N^{BA} defines the number of ants to use in the colony.

3.4 Tabu search optimisation

A simple tabu search was also implemented for this optimisation FJSP problem. The proposed approach is to allow the ants to build their solutions as described in section 3.2 and then the resulting solutions are taken to a local optimum by the local search mechanism.

Each of these ant solutions is then used in the pheromone update stage. The local search is performed on every ant solution, every iteration, so it needs to be fairly fast. In the case of the FJSP problem, the method is to pick the machine responsible to the C_{max} and check if any operations $O_{i,j}$ could be swapped between other machines which would result in a lower makespan.

Following their concept, the local search considers one problem machine at a time and attempts to swap one operation from the problem machine with any other (non-problem) machine in the solution (non-problem operations). Then the ants are used to generate promising scheduling production solutions and the tabu search algorithm is used to try to improve these solutions.

The tabu search is performed on each problem machine and continues until there is no further improvement in the makspean value of the solution. An example of this algorithm is shown in section 4.

3.5 The set up parameter values

The set up parameter values used in the ant system scheduling algorithms are often very important in getting good results, however the appropriate values are very often entirely problem dependent [7], and cannot always be derived from features of the problem itself:

- α determines the degree to which pheromone trail is used as the ants build their solution. The lower the value, the less ‘attention’ the ants pay to the pheromone trail, but the higher values implicate the ants then perform too little exploration, after testing values in the range 0.1-0.75 this algorithm works well with relatively high values (around 0.5-0.75).

- β determines the extent to which heuristic information is used by the ants. Again, values between 0.1-0.75 were tested, and a value around 0.5 appeared to offer the best trade-off between following the heuristic and allowing the ants to explore the research space.

- τ is the value to which the pheromone trail values are initialized. Initially the value of the parameter should be moderately high to encourage initial exploration, while the pheromone evaporation procedure will gradually stabilise the pheromone trail.

- ρ is the pheromone evaporation parameter and is always set to be in the range $[0 < \rho < x]$. It defines how quickly the ants ‘forget’ past solutions. A higher value makes for a more aggressive search; it tests a value of around 0.5-0.75 to find good solutions.

- N^{BA} defines the number of ants to use in the colony, a low value speeds the algorithm up because less search is done, a high value slows the search down, as more ants run before each pheromone update is performed. A value of 10 appeared to be a good compromise between execution speed and the quality of the solution achieved.

It is interesting to note that for each value of parameters the ant systems scheduling meta-heuristics yields a good solution. Moreover, its convergence speed depends essentially on the number of used ants N^{BA}.

3.6 Building a solution steps

The main steps in the strategy of the FJSP system by ant systems and tabu search algorithm are given below.

- Initialize parameters N^{BA}, α, β, τ_0, ρ.
- Create an initial *solution* and an empty tabu list of a given size.

  In order to generate feasible and diverse solutions, initial ants are represented by solutions issued from heuristic rules [12] (SPT, DL, FIFO, etc) and a random method. Heuristics are used to approximate an optima solution as near as possible.
- **Repeat the following steps until the termination criteria are met:**
  - Find *new solution* by ant systems procedure scheduling given in section 3.2.
  - Evaluate the quality of the new solution.
  - If a *new solution* is improved then the current *best solution* becomes *new solution*
  - else If no *new solution* was improved then apply the tabu search optimisation given in section 3.4.
  - Add *solution* to the tabu list, if the tabu list is full then delete the oldest entry in the list.
  - Apply the updating pheromone trail procedure given in section 3.3.
- **END Repeat**

4 Illustration example

Let us consider a flexible job shop scheduling problem, this example is to execute three jobs J_j (*j=1,2,3*) and six machines M_k ($k = 1,\ldots,6$) described in table 1.

Applying the ant systems meta-heuristic, the simulation propose four different scheduling with *Cmax* = 19 ut (unit of time), shown in table 2 to 7.

The solution given in the table 7 has a makespan equal to 19 ut. The machine M_5 is the cause of this value of makespan. To solve this problem, the tabu search optimisation is applied for this solution. Indeed, this method finds the operation $O_{2,2}$ for job J_2 on M_2 that can be swapped with other machines which will reduce makespan to 18 ut. And this method finds that the operation $O_{1,3}$ for the job J_1 executed by M_2 and can be swapped with M_5 who will execute the operation $O_{2,2}$ for the job J_2. Finally, the obtained solution by the tabu search is better than before, table 8.

Table 3: Example benchmark 3 jobs - 6 machines : processing time and ordering operation.

		M_1	M_2	M_3	M_4	M_5	M_6
J_1	$O_{1,1}$	10	7	6	13	5	1
	$O_{2,1}$	4	5	8	12	7	11
	$O_{3,1}$	9	5	6	12	6	17
	$O_{4,1}$	7	8	4	10	15	3
J_2	$O_{1,2}$	15	12	8	6	10	9
	$O_{2,2}$	9	5	7	13	4	7
	$O_{3,2}$	14	13	14	20	8	17
J_3	$O_{1,3}$	7	16	5	11	17	9
	$O_{2,3}$	9	16	8	11	6	3
	$O_{3,3}$	6	14	8	18	21	14

Table 4: Solution 1 for benchmark 3 jobs - 6 machines.
$N^{BA} = 10; \alpha = 0.5; \ \beta = 0.5; \ \tau_0 = 0.01; \rho = 0.5$

S_1	O_1	O_2	O_3	O_4
J_1	M_6: [0,1]	M_1: [1,5]	M_2: [11,16]	M_6: [16,19]
J_2	M_4: [0,6]	M_2: [6, 11]	M_5: [11,19]	***
J_3	M_3: [0,5]	M_3: [5,13]	M_1: [13; 19]	***

Table 5: Solution 2 for benchmark 3 jobs - 6 machines.
$N^{BA} = 10; \alpha = 0.75; \ \beta = 0.25; \ \tau_0 = 0.01; \rho = 0.5$

S_2	O_1	O_2	O_3	O_4
J_1	M_6: [0,1]	M_1: [1,5]	M_2: [5,10]	$M6$: [10,13]
J_2	M_4: [0,6]	M_5: [6,10]	M_5: [10,18]	***
J_3	M_3: [0,5]	M_3: [5,13]	M_1: [13,19]	***

Table 6: Solution 3 for benchmark 3 jobs - 6 machines.
$N^{BA} = 10; \alpha = 0.25; \ \beta = 0.75; \ \tau_0 = 0.01; \rho = 0.5$

S_3	O_1	O_2	O_3	O_4
J_1	M_6: [0,1]	M_1: [1,5]	M_5: [5,11]	M_6: [11,14]
J_2	M_4: [0,6]	M_2: [6,11]	M_5: [11; 19]	***
J_3	M_3: [0,5]	M_6: [5,8]	M_1: [8,14]	***

Table 7: Solution 4 for benchmark 3 jobs - 6 machines.
$N^{BA} = 10; \alpha = 0.3; \ \beta = 0.7; \ \tau_0 = 0.01; \rho = 0.5$

S_4	O_1	O_2	O_3	O_4
J_1	M_6: [0,1]	M_1: [1,5]	M_5: [5,11]	M_6: [10,13]
J_2	M_4: [0,6]	M_2: [6,11]	M_5***: [11,19]***	***
J_3	M_3: [0,5]	M_6: [5,8]	M_1: [8,14]	***

Table 8: Tabu search optimisation solution.

	O_1	O_2	O_3	O_4
J_1	M_6; 0; 1	M_1; 1; 5	M_2***; 5; 10***	M_6; 10; 13
J_2	M_4; 0; 6	M_5***; 6; 10***	M_5***; 11; 18***	***
J_3	M_3; 0; 5	M_6; 5; 8	M_1; 8; 14	***

Table 9: Results of problem sets solution.

	Results problem sets from [11]		
Problem Set	Kacem et al	GENACE	Ant systems and tabu search optimisation
FJSP 4-5	16	11	11
FJSP 10-7	***15***	***12***	***11***
FJSP 10-10	7	7	7
FJSP 15-10	23	12	12
	Results problem sets from [3]		
FJSP 10-6	***32***	***29***	***28***
FJSP 10-15	86	68	68

Table 10: Result example : FJSP 10-10 from [11]
$N^{BA} = 10; \alpha = 0.1;\ \beta = 0.9;\ \tau_0 = 0.1; \rho = 0.25$

*	O_1	O_2	O_3
J_1	M_7 :[0,2]	M_3 :[2,3]	M_4 :[3,4]
J_2	M_1 :[1,3]	M_{10} :[3,4]	M_{10} :[4,6]
J_3	M_{10} :[0,1]	M_8 :[1,2]	M_8 :[2,4]
J_4	M_9 :[0,1]	M_3 :[3,6]	M_4 :[6,7]
J_5	M_9 :[1,3]	M_9 :[3,4]	M_4 :[4,5]
J_6	M_6 :[1,3]	M_9 :[4,6]	M_9 :[6,7]
J_7	M_1 :[0,1]	M_3 :[1,2]	M_4 :[2,3]
J_8	M_5 :[0,2]	M_2 :[2,5]	M_2 :[5,7]
J_9	M_3 :[0,1]	M_7 :[2,3]	M_6 :[3,4]
J_{10}	M_6 :[0,1]	M_4 :[1,2]	M_7 :[3,5]

Table 11: Result example : FJSP 10-6 from [3]
$N^{BA} = 10; \alpha = 0.25; \ \beta = 0.75; \ \tau_0 = 0.2; \rho = 0.5$

*	O_1	O_2	O_3	O_4	O_5	O_6
J_1	M_6 :[0,1]	M_6 :[4,9]	M_1 :[9,10]	M_2 :[15,21]	M_6:[23,26]	M_2 :[27,28]
J_2	M_4 :[1,3]	M_6 :[9,12]	M_5 :[13,15]	M_3 :[16,19]	M_3 :[19,21]	M_1 :[21,24]
J_3	M_1 :[2,3]	M_1 :[5,7]	M_2 :[14,15]	M_5 :[15,17]	M_4 :[21,22]	M_5 :[24,27]
J_4	M_5 :[0,2]	M_6 :[3,4]	M_2 :[8,14]	M_1 :[14,15]	M_4 :[15,20]	M_4 :[20,21]
J_5	M_1 :[0,2]	M_2 :[2,5]	M_4 :[5,11]	M_1 :[11,13]	M_4 :[13,15]	M_3 :[24,27]
J_6	M_5 :[2,5]	M_3 :[8,12]	M_6 :[12,14]	M_1 :[15,17]	M_2 :[21,27]	M_6 :[27,28]
J_7	M_4 :[0,1]	M_5 :[5,7]	M_1 :[7,9]	M_5 :[10,13]	M_6 :[20,23]	M_4 :[24,27]
J_8	M_3 :[1,4]	M_2 :[5,8]	M_1 :[10,12]	M_5 :[17,20]	M_5 :[20,24]	M_1 :[24,27]
J_9	M_3 :[4,8]	M_3 :[12,16]	M_6 :[16,17]	M_1 :[17,21]	M_3 :[21,24]	M_4 :[27,28]
J_{10}	M_6 :[1,3]	M_1 :[3,5]	M_5 :[7,10]	M_4 :[11,12]	M_6 :[17,20]	M_4 :[22,24]

5 Validation and comparison

All ant systems and tabu search optimisation results presented are for 1000 iterations with 10 the number of ants, and each run was performed 10 times. The algorithms have been coded in VB language and tested using a P4 Pentium processor 2.4 GHz and Windows XP system.

To illustrate the effectiveness and performance of the algorithm proposed in this paper, six representative benchmark FJSP instances (represented by problem $n \times m$) based on practical data have been selected to compute. These benchmark instances are all taken from of Brandimarte [3] and from Kacem [11] as well as those from GENACE [14]. The different results obtained by proposed approach is presented and compared with the other methods in table 9.

Concerning the FJSP instances, the different results show that the solutions obtained are generally acceptable and satisfactory. The values of the different objective functions show the efficiency of the suggested approach, table 9. Moreover, the proposed method enables us to obtain good results in a polynomial computation time. In fact, the efficiency of this approach can be explained by the quality of the ant system algorithms combined by the tabu search heuristic to the optimization of solutions.

6 Conclusion

In this paper, a new approach based on the combination of the ant system with tabu search algorithm for solving flexible job-shop scheduling problems, is presented. The results for the reformulated problems show that the ant systems with local search meta-heuristic can find optimal solutions for different problems that can be adapted to deal with the FJSP problem. The performances of the new approach are evaluated and compared with the results obtained from other methods. The obtained results show the effectiveness of the proposed method. Ant system algorithms and the tabu search techniques described are very effective and they alone can outperform all the alternative techniques.

References

[1] A. C. F. Alvim, F. Glover, C. C. Ribiero and D. J. Aloise. A hybrid improvement heuristic for the bin packing problem, 2002. Available from: http://citeseer.nj.nec.com/557429.html.

[2] T. D. Braun, H. J. Siegel, N. Beck, L. L. Bölöni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao, D. Hensgen and R. F. Freund. A comparison of eleven static heuristics for

mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing*, 61(6):810-837, 2001.

[3] Brandimarte P., Routing and Scheduling in a Flexible Job-Shop by Tabu Search, *Annals of Operations Research*, vol. 2, pp. 158-183, 1993.

[4] Bilchev, G., Parmee, I.C. : The Ant Colony Metaphor for Searching Continuous Design Spaces. Lecture Notes in Computer Science, 993,pp. 25-39, 1995.

[5] A. L. Corcoran and R. L. Wainwright. A parallel island model genetic algorithm for the multiprocessor scheduling problem. In *Selected Areas in Cryptography*, pp. 483-487, 1994.

[6] M. Dorigo. Optimization, Learning and Natural Algorithms. PhD thesis, DEI, Polytecnico di Milano, Milan, 1992.

[7] M. Dorigo and T. Stützle. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In Glover, F. and Kochenberger, G., editors, *Handbook of Metaheuristics*, vol. 57 of *International Series in Operations Research and Management Science*, pp. 251-285. Kluwer Academic Publishers, 2002.

[8] M. Garey, D. Johnson, R. Sethi. The Complexity of Flow Shop and Job-shop Schedules. *Mathematics of Operations Research*, vol. 1(2), pp. 117-129, 1976.

[9] M. Garey and D. Johnson. Computers and Intractability: A Guide to the theory of NP-Completeness. *Freeman and Company*, San Francisco, 1979.

[10] I. Kacem, S. Hammadi and P. Borne. Approach by Localization and Multiobjective Evolutionary Optimization for Flexible Job-shop Scheduling Problems. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 32(1), pp. 1-13, 2002.

[11] I. Kacem, S. Hammadi and P. Borne. Pareto-optimality Approach for Flexible Job-Shop Scheduling Problems: Hybridization of Evolutionary Algorithms and Fuzzy Logic. *Mathematics and Computer in Simulation*, vol. 60, pp. 245-276, 2002.

[12] N. Liouane, S. Hammadi and P. Borne. Robust methodology for scheduling production in uncertain environment. *IMACS Multi-Conference on Computational Engineering in Systems Applications, CESA'98*, Hammamet, 1998.

[13] K. Mesghouni. Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de production. Thèse de Doctorat, Université de Lille1, Lille, 1998.

[14] J. C. Tay and N. B. Ho. GENACE: An Efficient Cultural Algorithm for Solving the Flexible Job-Shop Problem. *Proceedings of the IEEE Congress of Evolutionary Computation*, pp. 1759-1766, 2004.

[15] A. Thiesen,. Design and evaluation of tabu search algorithms for multiprocessor scheduling. *Journal of Heuristics*, Vol. 4, pp.141-160, 1998.

[16] S. van der Zwaan and C. Marques. Ant colony optimisation for job shop scheduling. In *Proceedings of the Third Workshop on Genetic Algorithms and Artificial Life, GAAL 99*,1999.

Noureddine Liouane[1], Ihsen Saad[2,3], Slim Hammadi[2], Pierre Borne[2]

[1]ATSI : Ecole Nationale d'Ingénieurs de Monastir,
rue Ibn El Jazzar, 5019 Monastir, Tunisie

[2]Ecole Centrale de Lille, Cité scientifique
Laboratoire d'Automatique, Genie Informatique et Signal
BP 48, 59651 Villeneuve d'Ascq Cedex, France

[3]Ecole Nationale d'Ingénieurs de Tunis
Unité de recherche LARA-Automatique
BP 37, Le Belvédère, 1002 Tunis, Tunisie

E-mail: noureddine.liouane@enim.rnu.tn, ihsen.saad@enit.rnu.tn,
slim.hammadi@ec-lille.fr, pierre.borne@ec-lille.fr

Noureddine Liouane was born in Kairouan, Tunisia, in 1963. He received the Master degree of science electrical genius in 1988 from the "Ecole Normale Supérieur de l'Enseignement Technique de Tunis". He received the Ph.D. degree from Ecole Centrale de Lille, France, in 1998. He is currently "Maitre Assistant" in the "Ecole Nationale d'Ingénieurs de Monastir" and Director of the "Institut Supérieure des Sciences Appliquées et de la Technologie de Kairouan". His research is related to the evolutionary optimization methods for discrete events systems, computer science and operational research.

Ihsen Saad was born in Monastir, Tunisia, in 1977. He received the Engineer Diploma degree in electrical automatic control engineering from the "Ecole Nationale d'Ingénieur de Gabès", Tunisia, in 2002. He obtained the Master of automatic and signal treatment in 2004 at the "Ecole Nationale d'Ingénieur de Tunis". He is currently preparing the Ph.D. degree in automatic and computer science within the framework of LAGIS-EC-Lille and LARA-ENIT cooperation. His research is related to the evolutionary optimization methods for discrete events systems, computer science and operational research.

Slim Hammadi is a full Professor of production planning and control at the Ecole Centrale de Lille (French "Grande Ecole"). Born in Gafsa (Tunisia) in 1962, he has obtained by 1988 the Master degree in Computer science from the University of Lille (France). Pr Hammadi obtained a P.h.D degree in job-shop scheduling and control in 1991 at Ecole Centrale de Lille. He is a senior member of IEEE/SMC and has served as a referee for numerous journals including the IEEE Transactions on SMC. Pr. S. Hammadi was Co-Organizer of a Symposium (IMS) of the IMACS/IEEE SMC Multiconference CESA'98 held in Hammamet (Tunisia) in April 1998. He has organized several invited sessions in different SMC conferences where he was session chairman. He was chairman of the International congress on "Logistic and Transport" LT'2004, MHOSI'2005 and LT'2006. His teaching and research interests focus on the areas of production control, production planning, computer science, discrete and dynamic programming and computer integrated manufacturing.

Pierre Borne was born in Corbeil, France in 1944, he received the Master degree of Physics in 1967, the Masters of Electronics, of Mechanics and of Applied Mathematics in 1968. The same year he obtained the Diploma of "Ingénieur IDN" (French “Grande Ecole”). He obtained the PhD in Automatic Control of the University of Lille in 1970 and the DSc of the same University in 1976. He became Doctor Honoris Causa of the Moscow Institute of Electronics and Mathematics (Russia) in 1999 and of the University of Waterloo (Canada) in 2006. He is author or co-author of about 200 Journal articles and book chapters, and of 30 plenary lectures and of more than 250 communications in international conferences. He has been the supervisor of 64 PhD thesis and is author of 20 books. He has been President of the IEEE/SMC society in 2000 and 2001. He is presently Professor “de classe exceptionnelle” at the Ecole Centrale de Lille and director of the french pluriformations national group of research in Automatic Control.