A MODIFIED INVASIVE WEED OPTIMIZATION ALGORITHM FOR MULTIOBJECTIVE FLEXIBLE JOB SHOP SCHEDULING PROBLEMS

Souad Mekni and Besma Chaar Fayech

National School of Engineering of Tunis, Tunisia, LR-ACS-ENIT
msouadpop@gmail.com,Besma.fayechchaar@insat.rnu.tn

ABSTRACT

In this paper, a modified invasive weed optimization (IWO) algorithm is presented for optimization of multiobjective flexible job shop scheduling problems (FJSSPs) with the criteria to minimize the maximum completion time (makespan), the total workload of machines and the workload of the critical machine. IWO is a bio-inspired metaheuristic that mimics the ecological behaviour of weeds in colonizing and finding suitable place for growth and reproduction. IWO is developed to solve continuous optimization problems that's why the heuristic rule the Smallest Position Value (SPV) is used to convert the continuous position values to the discrete job sequences. The computational experiments show that the proposed algorithm is highly competitive to the state-of-the-art methods in the literature since it is able to find the optimal and best-known solutions on the instances studied.

KEYWORDS

Flexible job shop scheduling problem, Multiobjective optimization, Metaheuristics, Smallest Position Value, Invasive Weed Optimization.

1. INTRODUCTION

Solving a NP-hard scheduling problem with only one objective is a difficult task. Adding more objectives obviously makes this problem more difficult to solve. In fact, while in single objective optimization the optimal solution is usually clearly defined, this does not hold for multiobjective optimization problems. Instead of a single optimum, there is rather a set of good compromises solutions, generally known as Pareto optimal solutions from which the decision maker will select one. These solutions are optimal in the wider sense that no other solution in the search space is superior when all objectives are considered. Recently, it was recognized that Invasive Weed Optimization (IWO) was well suited to multiobjective optimization.

The invasive Weed Optimization algorithm developed by Mehrabian and Lucas [1] in 2006 is a newly stochastic optimization approach inspired from a common phenomenon in agriculture: colonization of invasive weeds. IWO is an appropriate competitor for other evolutionary algorithms. In fact, it is simple and easy to understand and program. It has strong robustness and fast global searching ability.

David C. Wyld et al. (Eds) : SAI, CDKP, ICAITA, NeCoM, SEAS, CMCA, ASUC, Signal - 2014
pp. 51–60, 2014. © CS & IT-CSCP 2014 DOI : 10.5121/csit.2014.41106

Some of the distinctive properties of IWO in comparison with other numerical search algorithms are the way of reproduction, spatial dispersal, and competitive exclusion. These properties are presented in details in section 3. Section 2 introduces and formulates the flexible job shop scheduling problem .The experiments are provided in section 4. Finally, brief conclusions and future perspectives are discussed in section 5.

2. MATHEMATICAL FORMULATION

The problem of flexible job shop scheduling (FJSSP) belongs to the NP-hard family [2]. It presents two difficulties. The first one is the assignment of each operation to a machine, and the second one is the scheduling of this set of operations in order to optimize our criteria. The result of a scheduling algorithm must be a schedule that contains the start times and allocation of resources to each operation. The data, constraints and objective of our problem are as follows:

2.1. Data

- M represents a set of m machines. A machine is called $M_k (k = 1,...,m)$, each M_k has a Workload called W_k.
- N represents a set of n jobs. A job is called $j_i (i = 1,...n)$, each job has a linear sequence of n_i operations.
- $O_{i,j}$ represents the operation number j of the job number i. The realization of each operation $O_{i,j}$ requires a machine M_k and a processing time $p_{i,j,k}$. The starting time of $O_{i,j}$ is $t_{i,j}$ and the ending time is $t_{f_{i,j}}$.

2.2. Constraints

- Machines are independent of one another.
- A machine can be unavailable during the scheduling (case of machine breakdown).
- Jobs are independent of one another.
- In our work, we suppose that: each job j_i can start at the date $t = 0$ and the total number of operations to perform is greater than the number of machines.

2.3. Criteria

We have to minimize Cr_1, Cr_2 and Cr_3:

- The makespan: Cr_1
- The total workload of machines: Cr_2
- The workload of the most loaded machine: Cr_3

In this paper, the objective is to find a schedule which has a minimum makespan, a minimum total workload of machines and a minimum workload of the critical machine. The sum of these three objectives is taken as the objective function. To measure the quality of solutions found, we use the lower bounds (BCr_1 for makespan, BCr_2 for total machine workload, and BCr_3 for the workload of the most loaded machine) proposed in [3].

3. INVASIVE WEED OPTIMIZATION ALGORITHM FOR FJSSP

The IWO algorithm was proposed by Mehrabian and Lucas [1] in 2006, and since then, it has been successfully utilized in different practical optimization problems such as optimal positioning of piezoelectric actuators [4], demanding a recommender system [5], Studying electricity market dynamics [6], Design of an E shaped MIMO antenna [7] and encoding sequences for DNA computing [8].

3.1. Invasive Weed Optimization Algorithm

A weed is any plant growing where it is not wanted. Weeds have shown very robust and adaptive nature which turns them to undesirable plants in agriculture. A common belief in agronomy is that “The Weeds Always Win”. The harder people try, the better they get [1]. Recently, many studies are carried out with inspirations from ecological phenomena for developing optimization techniques. The new algorithm that is motivated by a common phenomenon in agriculture is colonization of invasive weeds. The flow chart of this algorithm is shown in Figure1 and the details of IWO are addressed as follows:

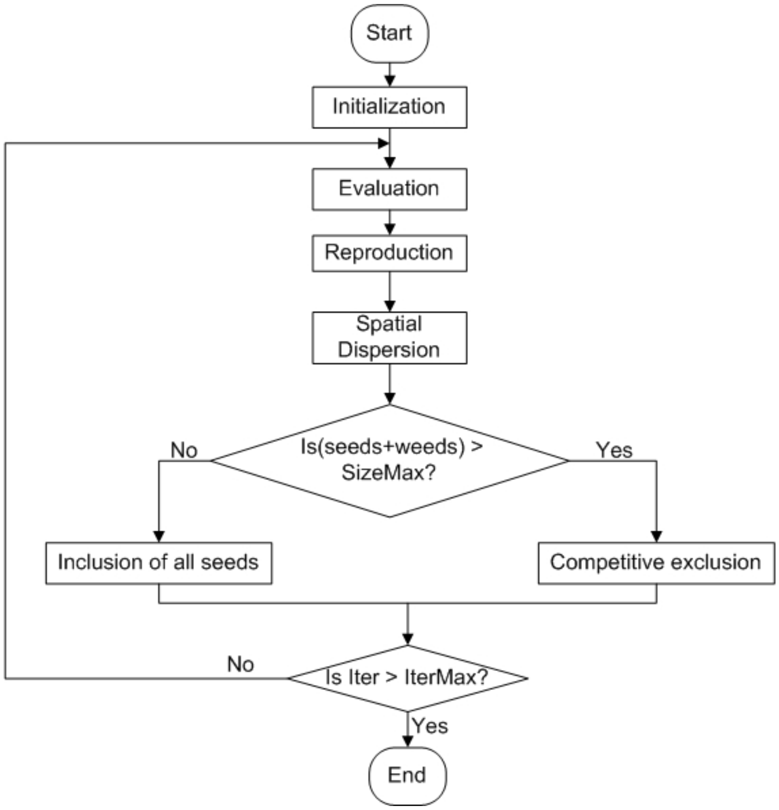


Figure 1. Flow Chart of IWO

3.1.1. Initialization

A population of initial solutions (weeds) is randomly generated over the search space.

3.1.2. Evaluation

The fitness of each weed in the population is calculated.

3.1.3. Reproduction

Each weed in the population is allowed to produce seeds depending on its comparative fitness in the population. In other words, a weed will produce seeds based on its fitness, the worst fitness and the best fitness in the population. In such way, the increase of number of seeds produced is linear. The number of seeds for each weed varies linearly between $S_{\min}$ for the worst plant and $S_{\max}$ for the best plant. Figure 2 illustrates the procedure of reproduction.

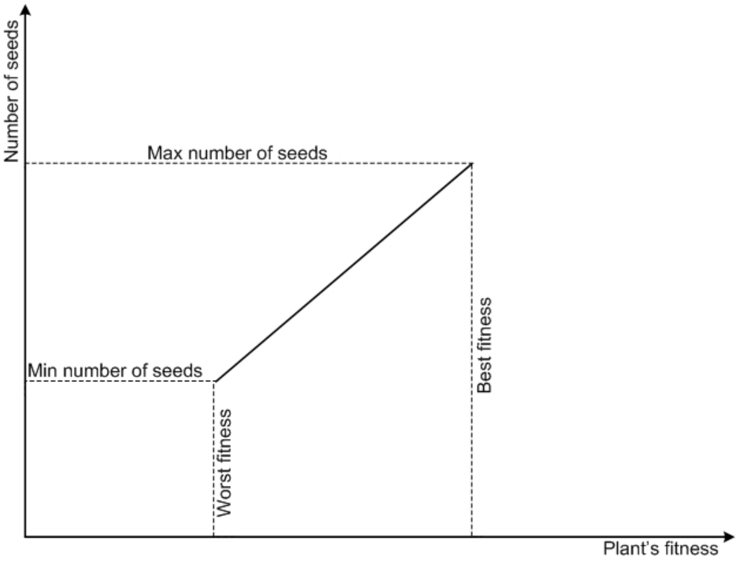


Figure 2. Procedure of reproduction

The equation for determining $Weed_{num}$ the number of seeds produced by each weed is presented in equation (1):

$Weed_{num} = S_{\min} + (S_{\max} - S_{\min})\frac{f - f_{worst}}{f_{best} - f_{worst}}$	Equation (1)

Where f is the fitness of the weed considered, f_{worst} and f_{best} are respectively the worst and the best fitness in the population. For better clarification, the application of equation (1) is shown in Figure 3. In this figure, it is assumed that weed$_5$ and weed$_1$ are the worst and best weeds between a population containing five weeds. So, the number of seeds around Weed$_5$ is equal to $S_{\min}$ and the number of seeds around Weed$_1$ is equal to $S_{\max}$.

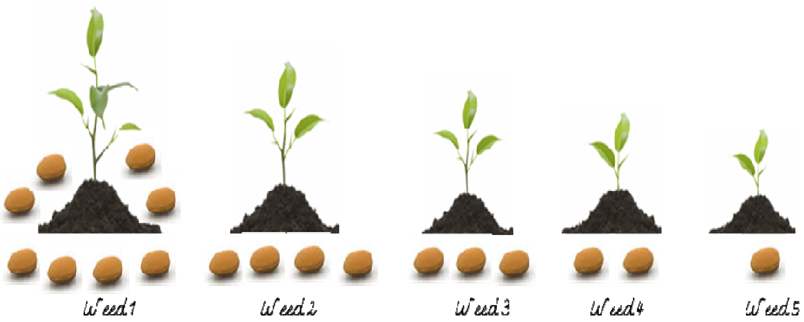


Figure 3. Schematic reproduction procedure for a problem with 5 weeds

3.1.4. Spatial Dispersal

This step ensures that the produced seeds will be generated around the parent weed, leading to a local search around each plant. The generated seeds are randomly spread out around the parent weeds according to a normal distribution with mean equal to zero and variance σ^2. The standard deviation of the seed dispersion σ decreases as a function of the number of iterations $iter$. The equation for determining the standard deviation for each generation is presented in equation (2):

$$\sigma_{iter} = \frac{(iter_{\max} - iter)^n}{(iter_{\max})^n}(\sigma_{initial} - \sigma_{final}) + \sigma_{final}$$

Equation (2)

Where $iter_{\max}$ is the maximum number of iterations. σ_{iter} is the standard deviation at the current iteration and n is the nonlinear modulation index. Obviously, the value of σ defines the exploration ability of the weeds. Therefore, as $iter$ increases, the exploration ability of all weeds is gradually reduced. At the end of the optimization process, the exploration ability has diminished so much that every weed can only fine its position [9].

3.1.5. Competitive exclusion

After a number of iterations, the population reaches its maximum, and an elimination mechanism is adopted: The seeds and their parents are ranked together and only those with better fitness can survive and become reproductive. Others are being eliminated.

3.2. Weed representation of FJJSP

The original IWO is developed to solve continuous optimization problems, but it can not be applied to discrete problems directly: individuals must be encoded appropriately to solve scheduling problems. In this paper, we implement a coding that takes into account all the constraints and the specifities of the problem. For the (n jobs, m machines, O operations) FJSSP, each plant is represented by four components: each component contains $2 \times O$ number of dimensions. Figure 5, Figure 6 and Figure 7 illustrate the solution representation of a weed corresponding to (3 jobs, 5 machines, 8 operations) FJSSP described in Figure 4. The 1st and 2nd halves of the 1st row of the weed (Figure 6 and Figure 7) represent operations as repetition of jobs (Figure 5). For example (J_1, J_1, J_1) represents ($O_{1,1}, O_{1,2}, O_{1,3}$), ($J_2, J_2, J_2$) represents ($O_{2,1}, O_{2,2}, O_{2,3}$), and so on. The 2nd row of (Figure 6 and Figure 7) represents weed's position. Each dimension of this row in Figure 6 maps one operation and each dimension of this row in

Figure 7 maps one machine. At this step, we use the Smallest Position Value (SPV) rule [10] to find the permutation of jobs. The smallest component of the weed's position in Figure 6 is -8 which corresponds to job number 1, thus J_1(or the first operation of J_1) is scheduled first. The second smallest component of the weed's position is -5,2 which corresponds to job number 2, therefore J_2(or the first operation of J_2)is the second in ordering, etc. The 2nd row of Figure 6 contains a random number in the interval $[0, m]$ that indicates after being rounded to its nearest integer the machine to which an operation is assigned during the course of IWO. The 3rd row of Figure 6 indicates the sequence of jobs in the ordering and the 3rd row of Figure 7 indicates the corresponding machines. Finally, the last row of Figure 6 indicates operations in the order and the last row of Figure 7 indicates starting times. In conclusion, the weed itself presents a solution as it shown in 3rd and 4th row of Figure 6 and Figure 7: First, the operation $O_{1,1}$ of job J_1 is executed by the machine M_1 at time $t = 0$, and then the operation $O_{2,1}$ of job J_2 is executed by the machine M_1 at time $t = 1$, and so on.

		M_1	M_2	M_3	M_4	M_5
J_1	$O_{1,1}$	1	9	3	7	5
	$O_{1,2}$	3	5	2	6	4
	$O_{1,3}$	6	7	1	4	3
J_2	$O_{2,1}$	1	4	5	3	8
	$O_{2,2}$	2	8	4	9	3
	$O_{2,3}$	9	5	1	2	4
J_3	$O_{3,1}$	1	8	9	3	2
	$O_{3,2}$	5	9	2	4	3

Figure 4. Example of $(3\ J, 5\ M, 8\ O)$ *FJSSP*

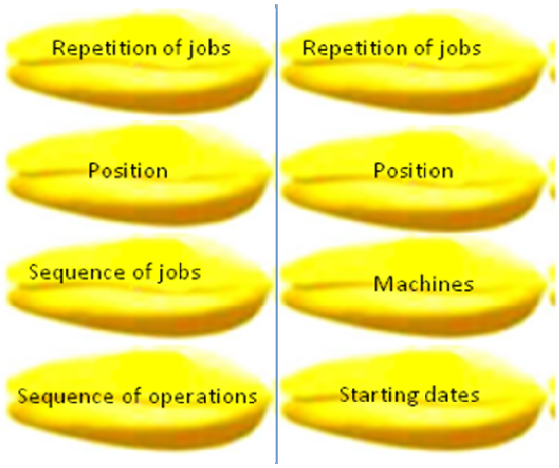
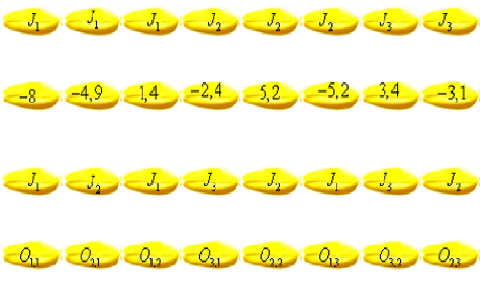


Figure 5. Weed representation


J_1 J_1 J_1 J_2 J_2 J_2 J_3 J_3
−8 −4,9 1,4 −2,4 5,2 −5,2 3,4 −3,1
J_1 J_2 J_1 J_3 J_2 J_1 J_3 J_2
$O_{1,1}$ $O_{2,1}$ $O_{1,2}$ $O_{3,1}$ $O_{2,2}$ $O_{1,3}$ $O_{3,2}$ $O_{2,3}$

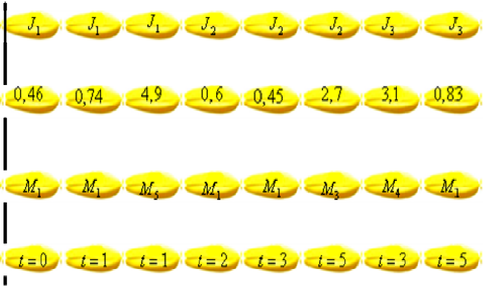

J_1 J_1 J_1 J_2 J_2 J_2 J_3 J_3
0,46 0,74 4,9 0,6 0,45 2,7 3,1 0,83
M_1 M_1 M_5 M_1 M_1 M_3 M_4 M_1
$t=0$ $t=1$ $t=1$ $t=2$ $t=3$ $t=5$ $t=3$ $t=5$

Figure 6. The first half of the weed

Figure 7. The second half of the weed

3.3. Pseudo-code of solving FJSSP by IWO algorithm

```
Begin{
  • Initialize population of weeds, set parameters;
  • Current_iteration=1;
  While (Current_iteration< Max_iteration)do
  {
    • Compute the best and worst fitness in the population
    • Compute the standard deviation std depending on iteration
    For each weed w in the population W
    {
      • Compute the number of seeds for w depending on its
        fitness
      • Select the seeds from the feasible solutions around the
        parent weed w in a neighborhood with normal distribution
        having mean=0 and standard deviation=std;
      • Add seeds produced to the population W
      If (|W|>Max_SizePopulation)
        {
        • Sort the population W according to their fitness
        • W=SelectBetter(weed,seed,Max_SizePopulation)
        }End if
    }End for
    Current_iteration=Current_iteration+1;
  }End while
}End
```

Figure 8. Pseudo code of IWO

4. EXPERIMENTAL RESULTS

Our approach is implemented in C++ on an Intel(R) Core(TM) i3 CPU M370@2,40 GHz machine. The non deterministic nature of IWO algorithm makes it necessary to carry out multiple runs on the same problem instance in order to obtain meaningful results. We run our algorithm twenty times from different starting solutions and tested it on a number of instances from literature. The convergence of IWO depends on the selection of three parameters: the initial standard deviation $\sigma_{initial}$, the final standard deviation σ_{final} and the non linear modulation index n .The chosen parameters for IWO are given in table 1.

Table 1. Parameters of IWO.

Parameters	**values**
Number of initial population	50
Maximum number of population	200
Maximum number of iterations: $iter_{\max}$	5000
Maximum number of seeds	5
Minimum number of seeds	1
$\sigma_{initial}$	10
σ_{final}	0,5
Non linear modulation index: n	3

To illustrate the effectiveness and performance of the algorithm used in this paper, we choose different instances of the problem of flexible job shop scheduling problem taken from Kacem [11]. Solutions in the literature to the instances presented in table 2 are presented in table3.

Table 2. Instances of Kacem.

Instances	**n(jobs)**	**m(machines)**
Instance 1	3	5
Instance 2	4	5
Instance 3	10	7
Instance 4	10	10
Instance 5	15	10
Instance 6	8	8

From table 3, we conclude that the obtained solutions are generally of a good quality. This is noted while comparing them with the existing approaches in the literature (for example Xia approach[12]) and also while comparing obtained values of the criteria with the computed lower bounds [3]. In fact, for instance 1, instance 2 and instance 3 our value of makespan Cr_1 is near the lower bound BCr_1, our value of total machine workload Cr_2 is near the lower bound BCr_2 and our value of the workload of the critical machine Cr_3 is near the lower bound BCr_3.

For instance 4, instance 5 and instance 6 our values of criteria are near lower bounds and similar or better (instance 4) than solutions found in [12].

Table 3. Solutions in Literature.

Instances	Lower Bounds	Xia et al [12]	IWO
Instance 1	BCr_1 =4 BCr_2 =11 BCr_3 =2	- - -	Cr_1 =5 Cr_2 =13 Cr_3 =5
Instance 2	BCr_1 =11 BCr_2 =32 BCr_3 =6	- - -	Cr_1 =11 Cr_2 =32 Cr_3 =10
Instance 3	BCr_1 =11 BCr_2 =60 BCr_3 =8	- - -	Cr_1 =11 Cr_2 =61 Cr_3 =11
Instance 4	BCr_1 =7 BCr_2 =41 BCr_3 =4	Cr_1 =7 Cr_2 =44 Cr_3 =6	Cr_1 =7 Cr_2 =42 Cr_3 =6
Instance 5	BCr_1 =10 BCr_2 =91 BCr_3 =9	Cr_1 =12 Cr_2 =91 Cr_3 =11	Cr_1 =12 Cr_2 =91 Cr_3 =11
Instance 6	BCr_1 =12 BCr_2 =73 BCr_3 =9	Cr_1 =15 Cr_2 =75 Cr_3 =12	Cr_1 =14 Cr_2 =77 Cr_3 =12

5. CONCLUSIONS

In this paper, the performance of the Invasive Weed Optimization technique is investigated for solving the multiobjective flexible job shop scheduling problem. The main highlighting features in IWO are: it is simple and easy to understand and program and it has strong robustness and fast global searching ability.

Experimental results are encouraging since that the proposed algorithm is able to find relevant solutions minimizing makespan, total machine workload and the biggest machine workload on the studied instances. A more comprehensive study on a large number of instances should be made to test the efficiency of the proposed solution technique. Further investigation is needed to fully reveal the ability of IWO in tackling scheduling problems and solving other optimization problems. Future research should pay more attention to the hybridization of IWO and other metaheuristics in order to benefit from advantages of each algorithm.

REFERENCES

[1] A R, Mehrabian. & C, Lucas, (2006) “A novel numerical optimization algorithm inspired from weed colonization”, Ecological Informatics, Vol.1, pp355-366.

[2] M , Sakarovitch, (1984) “Optimisation combiantoire. Méthodes mathématiques et algorithmiques. Hermann, Editeurs des sciences et des arts, Paris.

[3] R, Dupas, (2004) " amelioration de performances des systems de production: apport des algorithms évolutionnistes aux problems d'ordonnancement cycliques et flexibles, Habilitation , Artois university.

[4] A R, Mehrabian & A, Yousefi-Koma (2007) "Optimal Positioning of Piezoelectric actuators on a smart fin using bio-inspired algorithms", Aerospace Science and technology, Vol 11, pp 174-182.

[5] H, Sepehri Rad & C, Lucas (2007) " A recommender system based on invasive weed optimization algorithm", IEEE Congress on Evolutionary Computation, CEC 2007, pp 4297-4304.

[6] M, Sahaheri-Ardakani & M.Rshanaei & A, Rahimi-Kian & C, Lucas (2008) " A study of electricity market dynamics using invasive weed colonization optimization", in Proc.IEEE Symp. Comput.Intell. Games, pp 276-282.

[7] A, R, Mallahzadeh & S, Es'haghi & A, Alipour (2009) "Design of an E shaped MIMO antenna using IWO algorithm for wireless application at 5.8 Ghz", Progress in Electromagnetic Research, PIER 90, pp 187-203.

[8] X, Zhang & Y,Wang & G, Cui & Y, Niu & J, Xu (2009) "Application of a novel IWO to the design of encoding sequence for DNA Computing, Comput. Math. Appl. 57, pp 2001-2008.

[9] Z, D, Zaharis & C, Skeberis & T, D, Xenos (2012) "Improved antenna array adaptive beamforming with low side lobe level using a novel adaptive invasive weed optimization method" , Progress in Electromagnetics Research , Vol 124, pp 137-150.

[10] S, Mekni & B, Châar Fayéch & M, Ksouri (2010) " TRIBES application to the flexible job shop scheduling problem", IMS 2010 10th IFAC Workshop on Intelligent Manufacturing Systems, Lisbon, Portugal, July 1st -2nd 2010.

[11] I, Kacem & S, Hammadi & P, Borne (2002) " Approach by localization and multiobjective evolutionary optimization for flexible job shop scheduling problems. IEEE Trans Systems, Man and Cybernetics, Vol 32,pp 245-276.

[12] W, Xia & Z, Wu (2005) " An effective hybrid optimization approach for multiobjective flexible job shop scheduling problems, Journal of Computers and Industrial Engineering, Vol 48,pp 409-425.

AUTHORS

Souad Mekni: received the diploma of Engineer in Computer Science from the Faculty of Science of Tunis (Tunisia) in 2003 and the Master degree in Automatic and Signal Processing from the National Engineering School of Tunis (Tunisia) in 2005. She is currently pursuing the Ph.D.degree in Electrical Engineering at the National Engineering School of Tunis. Her research interests include production scheduling, genetic algorithms, particle swarm optimization, multiobjective optimization, Invasive Weed Optimization and artificial intelligence.

Besma Fayéch Chaâr: received the diploma of Engineer in Industrial Engineering from the National Engineering School of Tunis (Tunisia) in 1999, the D.E.A degree and the Ph.D degree in Automatics and Industrial Computing from the University of Lille (France), in 2000, 2003, respectively. Currently, she is a teacher assistant in the Higher School of Sciences and Techniques of Tunis (Tunisia). Her research interests include scheduling, genetic algorithms, transportation systems, multiagent systems and decision-support systems.