AN ENHANCED GENETIC ALGORITHM FOR THE SINGLE MACHINE TOTAL WEIGHTED TARDINESS PROBLEM

DJAHEL BOUCHRA, LAMICHE CHAABANE

ABSTRACT. In this research paper, we aim to present a hybrid algorithm to solve the single machine total weighted tardiness scheduling problem (SMTWT) which has been proven in the literature as a NP-hard. The developed method called (WMDD-GASA) is based on the weighted modified due date (WMDD) dispatching rule, the genetic algorithm (GA) and the simulated annealing (SA) procedure. In our approach, GA is used to discover the search space globally and it is guided by the WMDD rule to improve the first population chromosome and SA is exploited in mutation step to enhance childs quality. Experimental results using a set of benchmark instances coming from OR-Library for different sizes showed its effectiveness.

Mathematics Subject Classification (2010): 68-XX
Key words: scheduling, WMDD-GASA, WMDD, genetic algorithm, simulated annealing, OR-Library.

Article history:
Received 2 March 2019
Accepted 06 May 2019

1. INTRODUCTION

Research on scheduling problems with objectives related to profitability measures is of great concern [1]. So that the decision-making process to assign a collection of jobs for performing on the existing machines with minimizing some objective function, mean flow time, tardiness, and sum of earliness and tardiness penalties. Among different types of scheduling environments, a single machine total weighted tardiness (SMTWT) problem is considered in this research work [2]. We address the problem of scheduling of n jobs on single machine where the goal is to minimize the total weighted tardiness cost [1].

The single machine total weighted tardiness problem is defined as follows: Consider n jobs to be processed without interruption on a single machine that can handle only one job in a specific time. Each job j, available for processing at zero time, has a positive processing time p_j, a positive weight w_j, and a positive due date d_j. For a given sequence of jobs, the tardiness of job j is defined as $T_j = \max \{0, C_j - d_j\}$,

where C_j is the completion time of job j. The objective of the total weighted tardiness problem is to find a processing order of all the jobs; this order is a schedule that minimizes the sum of the weighted tardiness of all jobs.

$$Minimize \qquad Z = \sum_{j=1}^{n} w_j T_j \qquad (1)$$

Thus, the problem is to schedule n jobs on a single machine to minimize the sum of the weighted tardiness of all the jobs [3]. In order to find an approximate solution of this problem, we propose a hybrid algorithm based on metaheuristics and dispatching rules. The developed technique takes in account the advantage of WMDD rule in the initialization step, the global search done by GA and the local improvement assured by SA.

The reminder of this paper is organized as follows: In section 2, some previous related works are presented. Section 3, 4 and 5 summarizes principal concepts of both GA, SA algorithms and the WMDD rule respectively. Section 6, 7 outlines our GA and SA algorithm, its components for the SMTWT problem. Experimental study is presented in section 8. All obtained results and discussions are given in section 9. Finally, we address our conclusion and our future work in section 10.

2. RELATED WORKS

Since the SMTWT problem was first discussed in the late 1950s, it has attracted much research. This research can be categorized into three main groups: exact solution methods, dispatching rules, and heuristic approaches. Exact solution methods consist of dynamic programming [4] and branch-and-bound [5] algorithms, in addition branch and bound algorithm that solves instances with up to 50 jobs to optimality within practical time and memory limit [5], also dynamic programming are limited by core storage requirements [4]. Both use dominance rules to restrict the search for the optimal solution and engender computational practicality.

A variety of dispatching rules have been proposed to solve problems quickly. J. Kim *et al.* [6] suggested five priority rules for the SMWTTSP-MAT by extending corresponding priority rules for the SMTTSP. Computational experiments on 270 test instances show that the suggested priority rules work much better than existing ones.

M. H. Zahmani *et al* [7] used decision trees to generate new transmission rules for a single machine that was solved using a genetic algorithm. Two research methods are proposed to use the new publishing rules, and a comparative study is presented with other publishing rules of the literature.

S. Budi and S. A. L. Safitri [8] developed the BBO to solve a discrete problem such as SMTWTP. Comparison was done with PSO and modified GA. As a result of this study, BBO has a better performance compared to (PSO). But, compared to modified Genetic Algorithm, it is still worse than modified GA.

In addition C. Lamiche [9] proposed an intelligent search technique called genetic simulated annealing algorithm , namely, GA is exploited as a global search strategy to

discover solution space, SA algorithm is used as a local search technique to enhance more efficiently the visited attractive regions to improve solution quality for total weighted tardiness problems with 40, 50, and 100 jobs. The results showed that to produce better solutions compared to results given by some other recently literature works. Several methods were suggested Bee Colony Optimization (BCO) algorithm [10], particle swarm optimization [11] and breakout dynasearch algorithm (BDS) [12] to try to solve a SMTWT problem.

3. GENETIC ALGORITHM

The genetic algorithm is top-most entity in any Genetic Algorithm implementation. It is adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. It is responsible for creating its population and evolving it until a termination condition is reached. The genetic algorithm also specifies the replacement strategy. The genetic algorithm has terminator, replacement, evolver, initializer and printer as its moderators.

The idea with GA is to use this power of evolution to solve optimization problems. The father of the original Genetic Algorithm was John Holland who invented it in the early 1970's [5].

4. SIMULATED ANNEALING

The simulated annealing algorithm has been proposed by Kirkpatrick et al. In 1983[13]. SA is a method for solving unconstrained and bound-constrained optimization problems. The method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy.

At each iteration of the simulated annealing algorithm, a new point is randomly generated. The distance of the new point from the current point, or the extent of the search, is based on a probability distribution with a scale proportional to the temperature. The algorithm accepts all new points that lower the objective, but also, with a certain probability, points that raise the objective. By accepting points that raise the objective, the algorithm avoids being trapped in local minima, and is able to explore globally for more possible solutions. An annealing schedule is selected to systematically decrease the temperature as the algorithm proceeds. As the temperature decreases, the algorithm reduces the extent of its search to converge to a minimum.

5. THE WEIGHTED MODIFIED DUE DATE

The WMDD is an effective dispatching rule for SMTWT scheduling problem developed by J. J. Kanet and X. Li [14], which outperforms other well-known rules (e.g., EDD, WSPT, WCRR). The pseudo–code of our WMDD is given below:

PSEUDO-CODE WMDD

1) Set time = 0, S = Ø, J= {1, 2,3….n}.
2) Compute the value of T_j for each job in J as follow.

$$T_j = \frac{max\{P_j\ ,\ d_j - time\}}{W_j}$$

3) Sort the value of T_j in ascending order, and select job k having the minimum value of T_i .
4) Add job k in S, then remove it from J.
5) Update time = time + P_κ
6) Terminate the procedure if J = Ø ; otherwise, go to step 1.

6. GA ALGORITHM FOR SMTWT PROBLEM

6.1. Solution Encoding

For the single machine total weighted tardiness problem, the natural permutation representation of a solution is a permutation of the integers 1…, n of n jobs [3]. For example, for a 10-job problem, Fig 1.

3	4	5	2	1	9	6	7	8	10

Fig 1: A chromosome

6.2. Population Initialization

The initial population is composed of a certain number denoted as P of chromosomes. Therefore, using a better technique to create the initial population would not only improve the average performance of genetic algorithm, but also to reduce computation time dramatically.

Here I will explain the idea which i have used: We distinguish two cases of the initial population first population are generated randomly and corresponds of feasible solutions, second population is applied the weighted modified due date (WMDD) method to construct the first individual in an initial population.

6.3. Selection

Chromosomes (parents) are selected from the population for combining to produce new chromosomes (children), for applying genetic operators [3]. Here we use a Wheel

roulette method that selects parents, each chromosomes has a chance to be selected proportional to his performance. In roulette wheel selection, individuals are given a probability of being selected that is directly proportionate to their fitness.

6.4. Crossover

The role of a crossover operator is to combine elements from two parent chromosomes to generate two child chromosomes. Here we use position-based crossover, the proposed scheme is illustrated in Fig. 2.

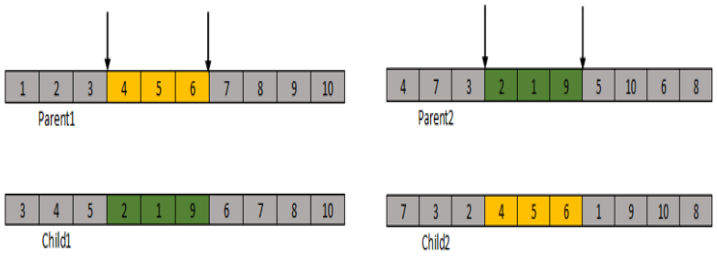

Fig. 2: Scheme of crossover operator.

6.5. Mutation

The role of a mutation operator is to provide and maintain diversity in a population so that other operators can continue to work. For that, it alters one or more genes with a probability equal to the mutation rate.

The mutation procedure used heuristic algorithm Simulated annealing (SA).

SA algorithm is a technic, which gets model and reference the annealing process of the melt metals during the cooling.

6.6. Replacement

The replacement strategy which are used here consists of the selection of the best chromosomes of the current population and their offspring [2]. They will form a new population to survive into the next generation, this step is very important to construct the next generation for the genetic algorithm.

7. SA FOR SMTWT PROBLEM

Generate a random solution: you can do this however you want. The main point is that it's random - it doesn't need to be your best guess at the optimal solution. Calculate its cost using some cost function you've defined.

Generate a random neighboring solution: "Neighboring" means there's only one thing that differs between the old solution and the new solution. Effectively, you switch two elements of your solution and re-calculate the cost. The main requirement is that it be done randomly.

Calculate the new solution's cost: use the same cost function as above. You can see why it needs to perform well - it gets called with each iteration of the algorithm.

Move to the new solution: if the new solution has a smaller cost than the old solution, the new one is better. This makes the algorithm happy, it's getting closer to an optimum. It will "move" to that new solution, saving it as the base for its next iteration.

Maybe move to the new solution: this is where things get interesting. To avoid that problem, it sometimes elects to keep the worse solution. To decide, the algorithm calculates something called the "acceptance probability" and then compares it to a random number [15]. The pseudo–code of our SA is given below:

PSEUDO-CODE OF SA ALGORITHM

Initialize $T_{initial}$, T_{final} and α and IT_{max}
Generate an initial solution $Seq_{initial}$
Create $Seq_{current}$ from initial solution $Seq_{initial}$
Calculate the cost C of $Seq_{current}$
$Seq_{better} = Seq_{current}$
$C_{better} = C_{Seqcurrent}$
$T = T_{initial}$
$I = 0$
While ($T > T_{final}$ && $I < IT_{max}$) **do**
  Create Seq_{new} using interchange neighborhood strategy
  Calculate $C(Seq_{new})$
  Difference $(C(Seq_{new}) - C(Seq_{current}))$
  If (difference ≤ 0) then
    If $C(Seq_{new}) < C(Seq_{better})$ then
      $Seq_{better} = Seq_{new}$
      $C_{better} = C(Seq_{new})$
    End if
$Seq_{current} = Seq_{new}$
  Else
    Boltzmann probability = $\exp^{(-difference/T)}$
      If (Boltzmann probability) > random(0,1) then
        $Seq_{current} = Seq_{new}$
      End if
  End if
  $T = \alpha.T$
End while
Return the optimal sequence Seq_{better} and its cost C_{better}

The pseudo-code of proposed algorithm WMDD-GASA can be summarized as follows:

PSEUDO-CODE OF WMDD-GASA ALGORITHM

Initialize $P_{opulation}$, C_{rate} ,M_{rate}, It_{max}
Generate P_{size} sequences randomly
Apply WMDD rule to the first chromosome
I=0
Repeat
  Select two parents
  Apply crossover operator with C_{rate} probability

Apply SA algorithm as a mutation operator according to the P_{mut} value
Evaluate all chromosomes (parents and childs) using the fitness function
Arranged parents and childs in decreasing order using their fitness function
Save the best $P_{opulation}$ chromosomes in Bestpop
Replace P_{size} with Bestpop
I=I+1
Until i=It_{max}

8. IMPLEMENTATION AND COMPUTATIONAL RESULTS

8.1. Experimental Settings

The proposed hybrid algorithms are coded in C++ on a personal computer with Quad-core Processor 1.4 GHz and 4GB memory with Visual Studio 2017 compiler.

In order to demonstrate the feasibility, the proposed WMDD-GASA strategy was tested for 10 SMTWT instances of 40 and 50 and 100 job problem sizes from the OR-library. We can note here, that the relative deviation rate (DEV) is calculated by the following equation:

$$DEV\% = 100 * (TWT_{Algorithm} - TWT_{Best})/TWT_{Best}$$

In addition, all key parameter values of our developed GASA algorithm are reported in table 2:

TABLE 2: Parameter Settings for Experiments.

Parameter	Value
Population size	25
Number of iteration	2000
Crossover rate	0.7
Mutation rate	0.2
Initial temperature	0.001
Final temperature	100
A	0.98

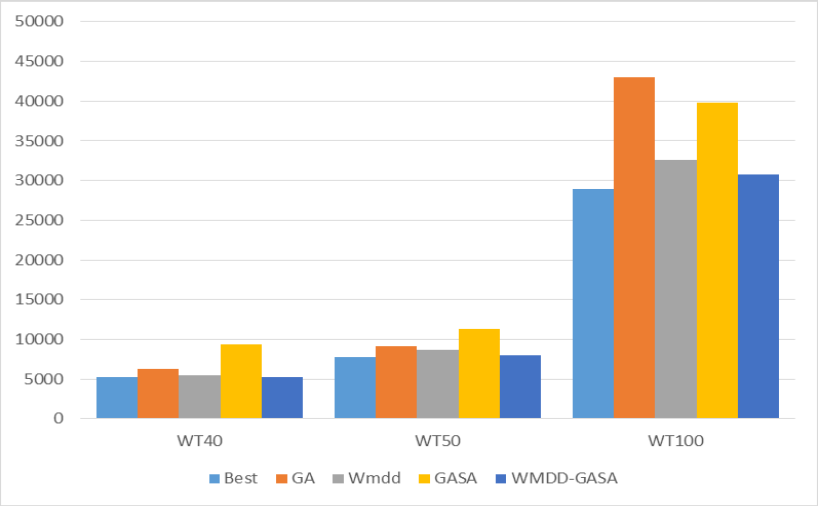

Fig. 3: Comparative results.

PROBLEM	BEST	GA	DEV%	WMDD	DEV%	GASA	DEV%	WMDD-GASA	DEV%
WT40-1	913	1570	71,96	1135	24,32	2391	161,88	956	4,71
WT40-2	1225	1778	45,14	1571	28,24	4083	233,31	1335	8,98
WT40-3	537	1255	133,71	573	6,70	4457	729,98	573	6,70
WT40-4	2094	2493	19,05	2288	9,26	3534	68,77	2116	1,05
WT40-5	990	1162	17,37	1362	37,58	5244	429,70	990	0,00
WT40-6	6955	8056	15,83	7158	2,92	12523	80,06	6955	0,00
WT40-7	6324	7750	22,55	6666	5,41	11954	89,03	6571	3,91
WT40-8	6865	8656	26,09	7105	3,50	10692	55,75	7007	2,07
WT50-1	2134	2656	24,46	2442	14,43	3928	84,07	2134	0,00
WT50-2	1996	2379	19,19	2281	14,28	4638	132,36	2082	4,31
WT50-3	2583	2641	2,25	2757	6,74	3423	32,52	2583	0,00
WT50-4	2691	4432	64,70	2819	4,76	6887	155,93	2691	0,00
WT50-5	1518	2740	80,50	2128	40,18	4493	195,98	1604	5,67
WT50-6	26276	30115	14,61	31388	19,46	30342	15,47	27705	5,44
WT50-7	11403	14081	23,49	12144	6,50	16661	46,11	11571	1,47
WT50-8	8499	10306	21,26	9208	8,34	15543	82,88	8854	4,18
WT100-1	5988	10249	71,16	6717	12,17	10972	83,23	6328	5,68
WT100-2	6170	12181	97,42	6922	12,19	7049	14,25	6308	2,24
WT100-3	4267	7578	77,60	5292	24,02	10576	147,86	4414	3,45
WT100-4	5011	11022	119,96	5571	11,18	6042	20,57	5216	4,09
WT100-5	5283	10592	100,49	6133	16,09	11659	120,69	5952	12,66
WT100-6	58258	83571	43,45	65041	11,64	83542	43,40	62281	6,91
WT100-7	50972	72605	42,44	56646	11,13	66110	29,70	53380	4,72
WT100-8	59434	84685	42,49	66371	11,67	72548	22,06	63355	6,60

TABLE 3: Numerical results for the problems.

8.3. Comparison of WMDD-GASA with other works

In second experiments, we compare our proposed method named (WMDD-GASA) with another work which is cited in [16] by using the same problem instances.

all obtained result on SMTWT instances of 40 and 50 and 100 job problem sizes coming from the OR-library are summarized in table 4 and Fig. 4 below.

TABLE 4: Comparative Results

PROBLEM	BEST	ACS[16]	DEV%	WMDD-GASA	DEV%
WT40-1	913	913	0,00	913	0,00
WT40-2	1225	1431	16,82	1335	8,98
WT40-3	537	537	0,00	573	6,70
WT40-4	2094	2163	3,30	2116	1,05
WT40-5	990	1090	10,10	990	0,00
WT40-6	6955	8151	17,20	6955	0,00
WT40-7	6324	9083	43,63	6571	3,91
WT40-8	6865	11474	67,14	7007	2,07
WT50-1	2134	2832	32,71	2134	0,00
WT50-2	1996	2557	28,11	2082	4,31
WT50-3	2583	2583	0,00	2583	0,00
WT50-4	2691	3278	21,81	2691	0,00
WT50-5	1518	2568	69,17	1604	5,67
WT50-6	26276	34167	30,03	27705	5,44
WT50-7	11403	13668	19,86	11571	1,47
WT50-8	8499	9713	14,28	8854	4,18
WT100-1	5988	8795	46,88	6328	5,68
WT100-2	6170	7724	25,19	6308	2,24
WT100-3	4267	5672	32,93	4414	3,45
WT100-4	5011	6426	28,24	5216	4,09
WT100-5	5283	7709	45,92	5952	12,66
WT100-6	58258	76424	31,18	62281	6,91
WT100-7	50972	83231	63,29	53380	4,72
WT100-8	59434	90968	53,06	63355	6,60

From this table, we can see clearly that the results produced by our approach are better compared with those obtained by ACS algorithm. Effectively, the relative deviation rate is improved significaintly by the proposed approach, this remarque is due to the advantages gained by the integartion of SA in mutation step.

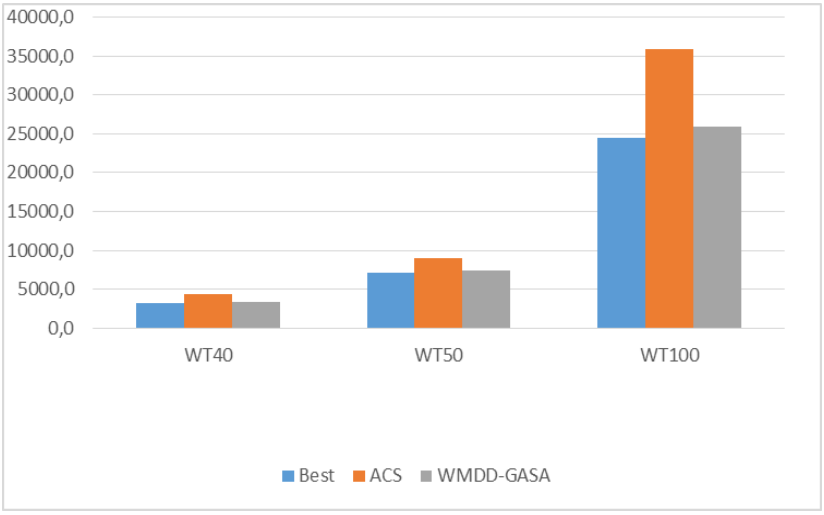

Fig. 4: WMDD-GASA vs. ACS algorithm results

9. Results and Discussion

The parameters of crossing over and mutation are set to 0,07 and 0,02 respectively. The number of populations varies problem to problem. We obtained the results by limiting the iteration number by 2000 iterations.

Results show that there is a parallel trend between number of jobs and total tardiness. If the number of jobs increases, total tardiness of single machine scheduling problem increases too. The increase in population and number of iterations lets us observe working of algorithm better. Also, it is noticed that problems are generated randomly and the results should be considered in this way. we note that the results of the proposed method two cases, In the first case of low total number of jobs provides a better solution for the lower and higher iteration numbers and the lower of population sizes as well as the higher in population sizes. In a second, the state of the high iteration's number, we note that the proposed method yields better solutions for all sizes of problem and population. The results quality converges to the better one when number of iterations increases.

Finally, repair mechanism is embedded to genetic algorithm for solving machine scheduling problems. This repair mechanism presents an efficient approach to producing efficient chromosomes.

10. Conclusion

This paper investigates a new hybrid algorithm called WMDD-GASA, in which the GA with WMDD as a guided rule for the first chromosome in the population is chosen and the SA procedure is applied in mutation step for solving the SMTWT problem. Numerical experiments, on various problems size with the 120 benchmark instances (8

instances for each problem size of 40, 50, and 100 jobs) taken from OR Library, showed the capability of the proposed approach to give better results compared with those obtained by GASA initialized randomly.

As a future work, we aim to apply another dispatching rules for initialization step or to use other mutation operators to improve the performance of the method. In addition, we can compare our technique with other recently published works in order to prove its effectiveness.

REFERENCES

[1] S.E. Jayanthi and S. Anusuya ,*Minimization of Total Weighted Earliness and Tardiness using PSO for One Machine Scheduling*, International Journal of Pure and Applied Mathematical Sciences 10(2017) 35-44.

[2] K. Wanatchapong, *Solving the Single Machine Total Weighted Tardiness Problem Using Bat-Inspired Algorithm*, IEEE International Conference on Industrial Engineering and Engineering Management, 2015

[3] L.Ning, M. Abdelrahman and R. Srin, *Genetic Algorithm for Single Machine Total Weighted Tardiness Scheduling Problem*, International Journal of Intelligent Control and Systems 10 '2005), 218-225.

[4] L. Schrage, and K. R. Baker, *Dynamic Programming Solution of Sequencing Problem with Precedence Constraints*, Operations Research 26 (1978), 444-449.

[5] A. Jouglet, P. Baptiste and J. Carlier, *Exact Procedure for Single Machine Total Cost Scheduling*, IEEE International Conference on Systems, Man and Cybernetics, Oct 6-9, 2002

[6] J. Kim and J. Y. Bang, S. K. Lim and J. Y. Lee, *Priority Rules for the Single Machine Total Weighted Tardiness Scheduling with Maximum Allowable Tardiness*, Advanced Science and Technology Letters 141'2016), 31-38.

[7] M. H. Zehmani, B. Atmani and A. Bekrar, *Efficient Dispatching Rules Based on Data Mining for The Single Machine Scheduling Problem*, Jan Zizka et al. (Eds) : ICAITA, SAI, CDKP, Signal, pp. 199–208, CS & IT-CSCP 2015

[8] S. Budi, A. L. Safitri, *Biogeography-based optimization (BBO) algorithm for single machine total weighted tardiness problem (SMTWTP)*, Procedia Manufacturing, 4 (2015), 552-557.

[9] C. Lamiche, *An Intelligent Cooperative Approach Applied to Single Machine Total Weighted Tardiness Scheduling Problem*, Journal of Information and Computing Science 12 (2017), 270-279.

[10] T. Davidovic, D. Teodorovic and M. Selmic, *Bee Colony Optimization Part I: The Algorithm Overview,* Yugoslav Journal of Operations Research 25 (2015), 33–56,

[11] D. Anghinolfi, M. Paolucci, *A new discrete particle swarm optimization approach for the single-machine total weighted tardiness scheduling problem with*

sequence-dependent setup times, European Journal of Operational Research 193(2009), 73-85.
[12] D. Junwen Ding , L. Zhipeng, T.C.E. Cheng and L. Xu, *Breakout dynasearch for the single-machine total weighted tardiness problem*, Computers & Industrial Engineering 98(2016), 1-10.
[13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by Simulated Annealing*, Science 220 (1983), 671-680.
[14] J. J. Kanet and X. Li, *A Weighted Modified Due Date Rule for Sequencing to Minimize Weighted Tardiness*, J. Sched., vol. 7(2004), 261–276.
[15] D. Kirrkpatrick, Gelatti, and Vecchi, *Optimization by Simulated Annealing*, 1983 Science 220 (1983), 671-680.
[16] A. Serbencu, V. Mînzu, D. Cernega, *A Hybrid Metaheuristic for Solving Single Machine Scheduling Problem*, 6th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2009

LABORATOIRE D'ANALYSE DES SIGNAUX ET SYSTÈMES (LASS), UNIVERSITY OF M'SILA, M'SILA, ALGERIA.
E-mail address: bouchra.djahel@univ-msila.dz
DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF M'SILA, M'SILA, ALGERIA.
E-mail address: chaabane.lamiche@univ-msila.dz
Lamiche07@gmail.com