International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 11 No. 3 Jun. 2015, pp. 623-632

http://www.ijias.issr-journals.org/

A hybrid algorithm to solve the single-machine scheduling problem

Asma Karray[1], Mohamed Benrejeb[1], and Pierre Borne[2]

[1]LARA Automatique, Ecole Nationale d'Ingénieurs de Tunis, Tunis, Tunisia

[2]LAGIS, Ecole Centrale de Lille, Villeneuve d'Ascq, France

ABSTRACT: This paper deals with the multi-objective single-machine scheduling problem in agro-food industry. To solve this problem, a new hybrid algorithm is proposed. This new algorithm named SHGA/SA is composed of two well-known metaheuristics: genetic algorithms and simulated annealing. The results show that our new approach can be used to solve the single-machine scheduling problem efficiently and in a short computational time. Also, the results show that the hybrid algorithm outperforms both the GA and SA.

KEYWORDS: Hybridization, genetic algorithms, simulated annealing, single-machine scheduling problem, agro-food industry.

1 INTRODUCTION

A scheduling problem is a combinatorial problem which distributes tasks to resources in order to achieve a definite performance measure. In scheduling problem, one or more purposes must be accomplished; the resources are limited and should be shared by tasks. It can also be defined by the manufacturing environments like single-machine, job-shop, flow-shop and open-shop. In this paper, we deal with a single-machine environment, an NP-hard in the strong sense [1], [2], [3].

Over the years, metaheuristics has shown their adaptability and their efficiency to solve scheduling problems, these include simulated annealing [4], tabu search [5] genetic algorithm [6] and colony optimization [7], particle swarm [8]. Besides exploring metaheuristics, the researchers have switched to consider another type of high level algorithms. These algorithms combine metaheuristics and/or other methods to obtain hybrid metaheuristics. The aim of hybridization is to exploit the combined methods in order to achieve better results for the optimization problem in terms of quality of the solution and the execution time.

Three main aspects are taken into account: the type of methods to hybridize, the level of hybridization and the execution sequence [9] [10]. The first refers to the type of the methods to be hybridized, it might be: combining two metaheuristics or metaheuristics with exact method. Hybridization is low level when a given function of a metaheuristics is replaced by another metaheuristics. On the other hand, when the different metaheuristics are self-contained and there is no direct relationship to the internal workings of a metaheuristics, it's a high-level hybridization.

For the third aspect, there are three different execution sequences: the sequential hybridization where a set of algorithms is applied one after another, each one using the output of the previous one as its input, the parallel synchronous hybridization where some metaheuristics or exact methods are used as an operator of another method and the parallel asynchronous hybridization which involves several search algorithms performing independently and cooperating to find a global optimum.

In this study, a sequential high-level hybridization is considered and two metaheuristics are taken in account Genetic Algorithms (GAs) and Simulated Annealing algorithms (SA). The proposed algorithm will be tested on a single machine scheduling problem in agro-food industry with the objectives to minimize the completion time, the expiration cost of product and the distribution discount coast. In addition, the proposed algorithm will be compared with genetic algorithms and simulated annealing developed by us in earlier studies to evaluate the effectiveness of the new algorithm.

The remainder of the paper is organized as follows. In Section 2, we briefly present the single-machine scheduling problem in agro-food industry considered in this work. In section 3, the proposed hybrid method is presented, followed by the implementation of SHGA/SA in section 4. The experimental study and some computational results are given in Section 5. We conclude in Section 6 with some remarks and indications for future work.

2 PROBLEM DESCRIPTION

In this section, we describe the problem formulation and the notations used in this paper. Scheduling problem, as defined by Baker [11], is a combinatorial problem that assigns tasks for resources, over a timeline.

In this paper, we deal with a single-machine scheduling problem in agro-food industry. In agro-food manufacturing, similarly to other industries workshops, each product has a particular release and due date. Though, these products have other particularities, as the validity date limit, which make their production systems different [12].

The objectives taken in account for our agro-food workshops are:

- the makespan,
- the cost minimization of expired products,
- the minimization of the time intervals between the finishing date and expedition date.

We use the following notation:

O_{ij} : j^{th} operation of product i

t_{ij} : effective starting time of O_{ij}

r_{ij} : earliest starting time O_{ij}

γ_{ij} : latest starting time O_{ij}

p_{ij} : processing time of O_{ij}

P_i : end-product number i

c_{ijk} : k^{th} component used by O_{ij}

v_{ijk} : validity date limit of the component

C_{P_i} : ending processing time of P_i

p_{ijk}^{rev} : income of the component c_{ijk}

$d_{P_i}^{liv}$: delivery date of P_i

dv_{P_i} : lifespan of P_i

dr_{P_i} : back delay of P_i

$d_{P_i}^{ven}$: unit sale price P_i

$C_{P_i}^{stk}$: storage cost by time unit of P_i

The criteria C1, C2 and C3 considered in this paper are:

- the completion time of the last job to leave the system,

$$C1 = C_{\max} = \max_{1 \le i \le n}\left(C_{P_i}\right) \tag{1}$$

- the expiration cost of product,

$$C2 = \sum_{i}\sum_{j}\sum_{k} P_{ijk}^{rev}\left(\frac{\max(0, t_{ij} - v_{ijk})}{\left(t_{ij} - v_{ijk}\right)}\right) \quad \textbf{(2)}$$

- the distribution discount costs,

$$C3 = \sum_{i} \max\left(0, d_{P_i}^{liv} - C_{P_i}\right) \times \left(\frac{P_{P_i}^{ven}}{Dv_{P_i} - Dr_{P_i}} + C_{P_i}^{stk}\right) \quad \textbf{(3)}$$

These criteria have to be optimized in order to minimize the objective function F, expressed as follow:

$$F = \alpha_1 C1 + \alpha_2 C2 + \alpha_3 C3 \quad \textbf{(4)}$$

such that: $\alpha_i \succ 0, i = 1, 2, 3$ and $\alpha_1 + \alpha_2 + \alpha_3 = 1$

where α_1, α_2 and α_3 represent the confidence coefficients that privilege one function instead of another.

In this paper, a trade off between several criteria is optimized while at the same time respecting the following constraints of the problem [13]:

- the precedence constraints: resulting from products routing,
- the disjunctive constraints: when a resource is required by several tasks at the same time.

3 Hybrid Of Genetic Algorithms And Simulated Annealing

3.1 Background on the hybridizations of GAs and SA

GAs and SA are both known as efficient approaches toward problem solving including the single-machine scheduling. GAs is a population-based method that begins with a population of solutions in parallel, although it has poor convergence properties. The SA has better convergence since it is a single-based method.

In this regard, many researches aimed to combine GAs and SA to have a more efficient optimization method with a good convergence. Chen and Fen [14] proved that combining of GAs and SA had better performance for ten difficult optimization problems than either GAs or SA independently. Based on parallel simulated annealing in [15] [16], Baydar[17] proposed a parallel simulated annealing using the survival of the fittest method. Wanget al. [18] developed a new hybrid of GA and SA which incorporate simulated annealing into genetic algorithms to escape from local optima.

3.2 The proposed hybrid method: SHGA/SA

The different approaches to hybridize GAs and SA defined in section II.1had its own efficiencies since the good characteristics of each metaheuristics are maintained. In this paper, a new hybrid method based on GAs and SA, SHGA/SA is proposed. The SHGA/SA is a sequential hybridization using the genetic algorithms to realize a good exploration and the simulated annealing for the exploitation task. The sequential hybrid algorithm considered here pipelines the GAs and the SA. The SA is applied after the GAs to exploit the result of the previous exploration of the search space by the GAs.

The issue of SHGA/SA is on deciding when to stop the GAs and trigger the SA, and which individuals must be selected for the SA to act on. The solution adopted is to wait for the stabilization of the fitness in the population of the GAs. Then the best individuals are the starting points of the SA. The proposed hybrid sequential GAs and SA mechanics is schematically presented in figure 1 and the flow chart of SHGA/SA in figure 2.

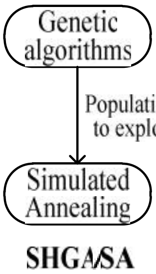

Fig. 1. The scheme of SHGA/SA

The steps of SHGA/SA are as follows:

Step 1: standard implementation of the GAs to resolve a multi-objectives single-machine scheduling problem: The first population of solutions is generated randomly. Fitness is used to select the better solutions from the current population. The offspring populations undergo the operations of crossover and mutation in order to create a population of new solutions. The process is repeated until a good solution is met.

Step 2: implementation of the TS sequentially after the GAs: use the best individual found in the final population from step 1 as the initial solution. Set the initial temperature, then a loop is started until the threshold is reached: a neighbor is selected by making a small change of the current solution; using the Metropolis rules, the move to the neighbor is accepted or not; afterwards the temperature is slowly decreased until the thermodynamics equilibrium is met.

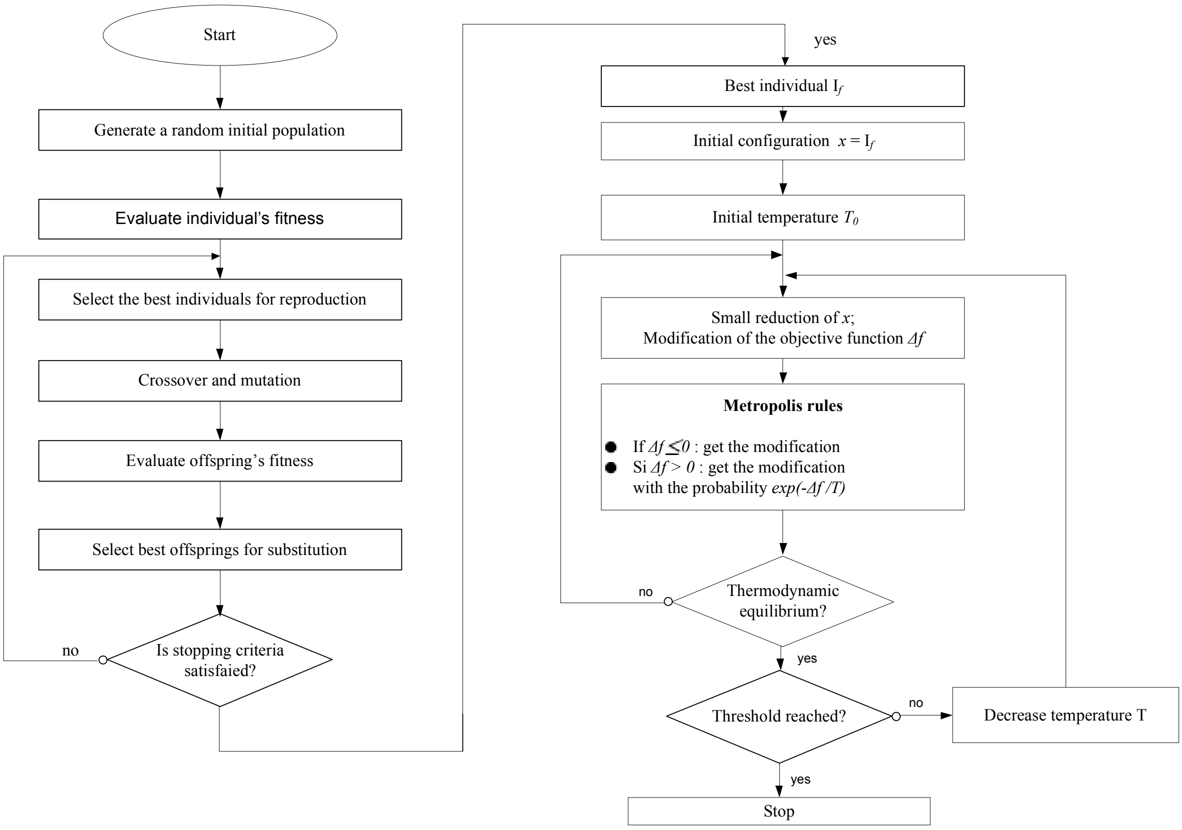

Fig. 2. SHGA/SA flowchart

4 Implementation Details Of SHGA/SA

In this section, the details of the proposed method SHGA/SA is presented. Choosing the appropriate parameters is in important step in the design of algorithm. In fact, a good configuration could lead to the global optimum in a short time while a worse setting could get the algorithm trapped in a local minimum or waste a long time running before finding a good solution.

Also, to efficiently solve the scheduling problem, it is important to select a proper solution representation. Indeed, the solution representation of a combinatorial optimization problem had a significant impact on the final outcome solution scheme, and should not only depend on the special characteristics of the problem itself but on the solution method as well [19].

4.1 THE SOLUTION REPRESENTATION:

In literature, we found two ways to represent a schedule: indirect and direct. In indirect representation, the chromosome contains an encoded schedule. A decoder is needed to transform the chromosome into a feasible schedule [20], [21].With, the direct representation, the chromosome represents the scheduling. The coding scheme taken is direct representation. It's inspired by the List Operations Code (LOC) proposed by Kacem [22].

The LOC representation guarantees to obtain a feasible solution after the crossover and mutation operations. Each chromosome is represented by a set of genes which defines the rank, the release time, the processing time and the latest finishing time of operation. The figure 3 shows the coding scheme.

$o_{i1}, r_{i1}, p_{i1}, \gamma_{i1}$	$o_{i2}, r_{i2}, p_{i2}, \gamma_{i2}$	......	$o_{ij}, r_{ij}, p_{ij}, \gamma_{ij}$

Fig. 3. Coding scheme

4.2 THE INITIAL POPULATION

In the literature, there are different methods and heuristics to generate the initial population [23]. Among these methods, we choose to generate randomly the initial population. A random number generation is executed to have initial sequences.

4.3 THE SELECTION OPERATOR

After the fitness of each chromosome is calculated, the second step is to select the better chromosomes for the crossover and mutation step. Though, there are many selection methods like the roulette wheel selection, the tournament selection and the rank selection. Goldberg and Deb show that the tournament has better convergence and time-complexity properties than the others [24]. Hence, the tournament selection is taken as the selection operators for our algorithm.

Its principle consists in randomly selecting a set of n individuals. These individuals are then ranked according to their relative fitness and the fittest individual is selected for reproduction. The whole process is repeated N times for the entire population. The crossover selection

In this step, two chromosomes are randomly selected for the crossover. Several methods exist for the crossover; although Murata *et al.* reported that the two-point crossover gives successful results in scheduling problems [25]. Thus, our hybrid method applies the Two-point Crossover to mate chromosomes.

4.4 THE MUTATION OPERATOR

The mutation is a random change that can be applied to avoid premature convergence. In this study, we apply the Random Sequencing Mutation which consists on exchanging two random genes of a solution set.

4.5 THE INITIAL TEMPERATURE

The initial value of the temperature parameter is an important feature for the success of the algorithm.

A low initial temperature can restrict the search only in the region around the starting point. On the other hand, a too high starting temperature will keep the algorithm searching over the model space during a large number of iterations. Thus, we will waste a valuable computational time.

In this study, the starting temperature T_0 can be calculated by conducting an initial search in which all increases are accepted and calculating the average objective increase observed $\langle \Delta f \rangle$ [26]. T_0 is given by:

$$\tau = \exp\left(\frac{-\left| < \Delta f > \right|}{T_0} \right) \tag{5}$$

Where τ is the acceptance probability.

4.6 THERMAL EQUILIBRIUM ACHIEVEMENT

At each iteration of the thermal equilibrium loop a new perturbed model is computed according to the perturbation scheme. This model is then accepted or rejected according to the acceptance criterion and a new iteration begins. This process is repeated until it is considered that 'thermal equilibrium' is reached. For the developed method, we choose to decrease the temperature after a certain number of transitions at each temperature.

4.7 COOLING FUNCTION

It defines the way in which the temperature is going to be decreased. It is also an important parameter in the success of the search. A very low cooling schedule will waste a valuable computational time to reach the global minimum. On the other hand, a too fast cooling schedule can get the algorithm trapped in a local minimum. In this study, we choose the geometric schedule:

$$T_{K+1} = \alpha . T_K \tag{6}$$

Where α is constant number between 0 and 1.

4.8 THE STOP CRITERION

The stop criterion or threshold is to wait until a certain defined number of acceptances is not achieved for some number of successive temperature values.

5 COMPUTATIONAL EXPERIMENTS

To demonstrate the effectiveness of the SHGA/SA, we compared the performance of our approach with the performances of GAs and SA. In addition, in order to prove that the proposed algorithm works well, single-machine scheduling problems in agro-food industry with the objective to optimize the three criteria C1, C2, and C3 are taken from [27] for benchmark tests.

5.1 THE PARAMETERS CONFIGURATION OF SHGA/SA

The parameters taken in consideration in our algorithm for the optimization process are summarized in table 1. These parameters are the population size, the iteration number, the crossover rate, the mutation rate, the initial temperature, the number of transitions at each temperature, the cooling function factor α and the stopping criterion.

Table 1. SHGA/SA parameters

	Benchmark 1	Benchmark 2
Population size	10	30
Iteration number	100	200
Crossover rate	0.7	0.7
Mutation rate	0.01	0.01
Initial temperature	75	96
Number of transition at each temperature	10	20
Cooling function factor	0.9	0.25
Stopping criterion	0.9	0.25

5.2 RESULTS AND DISCUSSION

Taking into consideration the data from benchmark tests, the completion time, the expiration cost of product and the distribution discount are calculated using the expressions (1), (2) and (3) and the confidence coefficients α_1, α_2 and α_3 are respectively equal to 0.4, 0.1 and 0.5.

The figures 3 and 4 show respectively the fitness progress for Benchmark 1 and Benchmark 2 using GAs, SA and SHGA/SA. The x-axis is the number of iterations while the y-axis shows the current solution at that iteration. In addition, the arrows show the global minimum obtained in each method for each benchmark. E.g. for Benchmark 1, SHGA/SA converges at the 118th iteration and finds a global minimum equal to 17.

From fig 3 and 4, it can be seen that SHGA/SA performs more efficiently than GAs or SA. For example, in figure7, we see that the global minimum is obtained after 118th iteration with SHGA/SA. While for the same benchmark and under the same condition, we have to wait respectively until the 180th iteration and the 250th iteration with GAs and SA to find the global minimum.

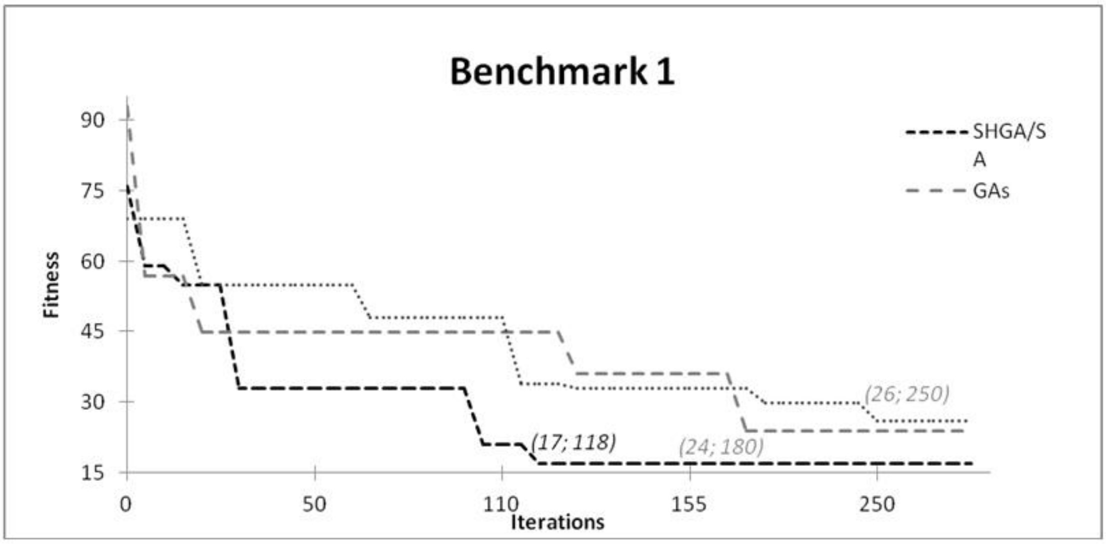

Fig. 4. Evolution process for the benchmark 1

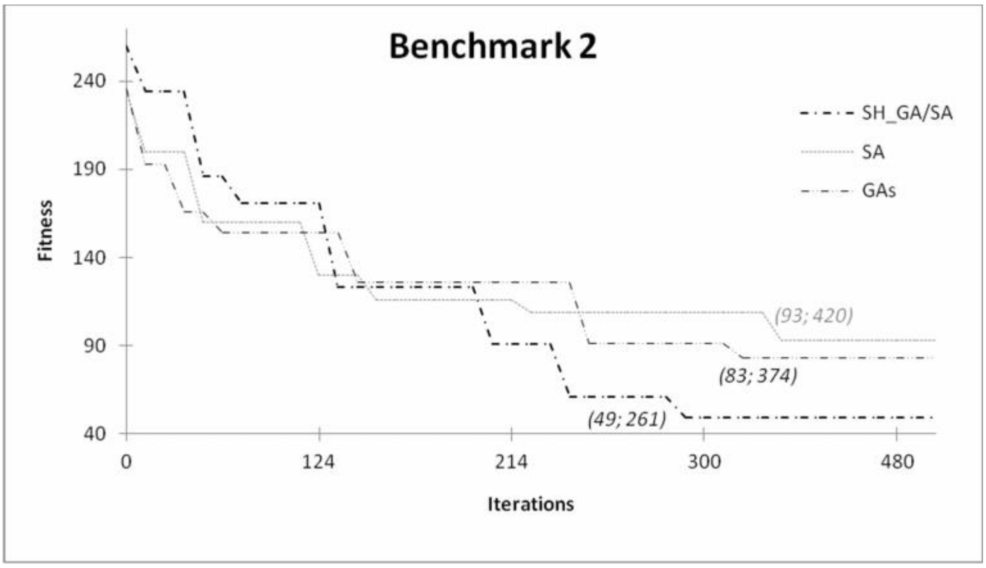

Fig. 5. Evolution process for the benchmark 2

The computational results for the Benchmark 1 and Benchmark 2 using SHGA/SA are detailed in table 2 below. We show the objective function, convergence point and the CPU computation time. A low objective function indicates a more optimal solution. The convergence column is the number of iterations till a final solution is reached. The CPU time is expressed in seconds.

In this table, we also show the computational results for GAs and SA as a comparison.

Table 2. Computational results

	GAs			SA			SHGA/SA		
	F	Convergence	CPU	F	Convergence	CPU	F	Convergence	CPU
Benchmark 1	24	180	8.669	26	250	2.116	17	118	2.374
Benchmark 2	83	374	68.356	93	420	28.210	49	261	22.022

The results from table show that, for Benchmark 1, SA has the fastest CPU time but finds the worst solution between the 3 algorithms. SHGA/SA runs in a comparable time to SA (in fact slightly slower) and finds the best solution. For Benchmark 2, SHGA/SA is faster than the other 2 algorithms in terms of CPU time and finds the best solution. The results indicate that the performances of SA were significantly improved when the sequential hybridization technique was applied.

Using table 2, we can also have a closer look at the speed difference between the different methods used. Methods SA and SHGA/SA run in comparable times. For Benchmark 1, the difference between SA and SHGA/SA is less than 1 second while for Benchmark 2, it’s less than 7 seconds .GAs is slower than the other 2 methods. For Benchmark 1, the difference is 6.5 seconds compared to SA and 6.5 second compared to SHGA/SA wile for benchmark 2, the difference is about 40.2 seconds compared to SA and 46.3 seconds compared to SHGA/SA. The results indicate that the performances time of GAs were significantly improved when the sequential hybridization technique was applied.

In conclusion, the hybrid method is more efficient than the classics GAs or SA. It finds a better result with an acceptable time convergence and a fast time completion. This efficiency can be explained by the fact that the hybrid method keeps the good characteristics of GAs and SA and overcomes their weaknesses when we combine them.

6 CONCLUSION

In this paper, is proposed a hybrid sequential GA/SA algorithm for the single-machine scheduling problem in agro-food industry. The hybridization follows a high level approach in which GAs followed by SA is applied. The objective function considered is that of three objectives in which the distribution discount costs and the makespan are primary objectives and the expiration cost of product is a secondary one.

To test the performance of the proposed hybrid algorithm, experiments were carried out on a single machine scheduling problem in agro-food industry. Computational results show that the combined GAs and SA outperform the GAs and SA applied separately in solution quality.

The experimental evaluation also shows that combining GAs and SA give better computational effort than GAs or SA alone.

Consequently, following the experiments in the single-machine scheduling problems in agro-food industry, the results are very satisfactory and convincing and we except to apply the SHGA/SA to other complex machine environments like job shop and other combinatorial problems in near future.

REFERENCES

[1] M. Garey, D. Johnson, *Computer and Intractability: a Guide to the Theory of NP- Completeness,* Freeman W-H, San Francisco, 1979.

[2] J. Carlier, “Scheduling jobs with release dates and tails on identical machines to minimize the makespan”, *European Journal of Operational Research,* Vol. 29, pp. 298-306, 1987.

[3] K. C. Ying, S. W. Lin, C. Y. Huang, “Sequencing single-machine tardiness problems with sequence dependent setup times using an iterated greedy heuristic”, *Expert Systems with Applications*, Vol. 36, pp. 7087–7092, 2009.

[4] F. Jin, S. Song, C. Wu, “A simulated annealing algorithm for single machine scheduling problems with family setups”, *Computers & Operations Research*, Vol. 36, pp. 2133- 2138, 2009.

[5] F. F. Choobineh, E. Mohebbi and H. Khoo, “A multiobjectivetabu search for a single-machine scheduling problem with sequence-dependent setup times”, *European Journal of Operational Research*, Vol. 175, pp. 318–337, 2006.

[6] M. Koksalan, A. B. Keha, “Using genetic algorithms for single machine bicriteria scheduling problems”, *European Journal of Operational Research*, Vol. 145, pp. 543–556, 2003.

[7] C. J. Liao, H. C. Juan, “An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups”, *Computers & Operations Research*, Vol. 34, Issue 7, pp. 1899-1909, 2007.

[8] 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”, Vol. 193, pp. 73–85, 2009.

[9] F. Xhafa, J. A. Gonzalez, K. P. Dahal and A. Abraham, “A GA (TS) Hybrid Algorithm for Scheduling in Computational Grids”, *Hybrid Artificial Intelligence Systems, Lecture Notes in Computer Science*, pp. : 285-292, 2009.

[10] A. Karray, M. Benrejeb, P. Borne, “A hybrid method based on genetic algorithms and tabu search for the single-machine scheduling problem in agro-food industry”, *International Review Of Automatic Control*, Vol. 4, N. 2, 2011.

[11] K. Baker, *Introduction to sequencing and scheduling*, John Wiley & Sons; 1943

[12] R. Treillon, C. Lecomte, *Gestion Industrielle des entreprises alimentaires,* Lavoisier TEC&DOC, Paris, 1996.

[13] A. Karray, M. Benrejeb, P. Borne, “New Parallel Genetic Algorithms for the single-machine scheduling problem in agro-food industry”, *2011 International Conference on Communications, Computing and Control Applications, CCCA’11*, March 3-5, Hammamet, Tunisia.

[14] H. Chen, N. Flann, “Parallel Simulated Annealing and Genetic Algorithms: A Space of Hybrid Methods,” In *Proc. Int’l Conf. Evolutionary computation − PPSN III, Lecture Notes in Computer Science*, vol. 866, Berlin: Springer-Verlag, 1994, pp. 428-438.

[15] S.W. Mahfoud, and D.E. Goldberg, “Parallel recombinative simulated annealing: A genetic algorithm,” *Parallel Computing*, vol. 21, no. 1, pp.1-28, Jan. 1995.

[16] T. Hiroyasu, M. Miki, and M. Ogura, “Parallel Simulated Annealing using Genetic Crossover,” in *Proc. IASTED Int’l Conf. on Parallel and Distributed Computing Systems*, Las Vegas, 2000, pp. 145-150.

[17] C. Baydar, “A hybrid parallel simulated annealing algorithm to optimize store performance,” in *Workshop on GECCO 2002*, New York, 2002.

[18] Z. G. Wang, M. Rahman, Y. S. Wong and K. S. Neo, “Development of Heterogeneous Parallel Genetic Simulated Annealing Using Multi-Niche Crowding”, *International Journal of Computer, Control, Quantum and Information Engineering* Vol. 1, No. 10, 2007.

[19] G. I. Zobolas, C.D. Tarantilis, G. Ioannou, “Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristics algorithm”, *Computers & Operations Research* ,Vol. 36, pp. 1249-1267, 2009.

[20] S. Bagchi, S. Uckum, Y. Miyabe, K. Kawamura, “Exploring Problem-Specifc Recombination Operators for Job Shop Scheduling". *International Conf. Genetic Algorithms* (ICGA-91), 1991.

[21] R. Nakano, T. Yamada. "Conventional Genetic Algorithms for Job Shop Problems". *International Conf. Genetic Algorithms* (ICGA-91), 1991.

[22] I. Kacem, S. Hammadi, P. Borne, "Direct chromosomes representation and advanced genetic operators for flexible job-shop problem", *Proceedings of CIMCA International Conference*, July 9-11, Las Vegas, Nevada, USA.

[23] C. Caux, H. I. Pierreval, M. C. Portman, « Les algorithmes génétiques et leurs applications aux problèmes d'ordonnancement ». *APII*, vol. 29 (4), pp. 409-443, 1995.

[24] D. E. Goldberg, K. Deb, "A comparison of selection schemes used in genetic algorithms", *in: G. J. E Rawlins (Ed), Foundation of Genetic Algorithms, Morgan Kaufman Publishers*, San Mateo, CA, USA, 1991, pp. 69-93.

[25] T. Murata, H. Ishibuchi, H. Tanaka, "Genetic algorithms for flowshop scheduling problems", *Comput. Ind. Eng.,* Vol. 30 pp. 1061-1071, 1996.

[26] A. Karray, *contribution à l'ordonnancement d'ateliers agroalimentaire utiliant des methodes d'hybridation hybride*, thèse de Doctorat, Ecole National d'Ingénieurs de Tunis, 2011.

[27] A. Karray, M. Benrejeb, P. Borne, « Algorithmes génétiques séquentiel pour la résolution de problèmes d'ordonnancement en industries agro-alimentaires », *Revue électronique Sciences et Technologies de l'Automatique, e-STA*, Vol. 8, N. 1, pp. 15-22, 2011.