

A Proposed Genetic Algorithm Coding for Flow-Shop Scheduling Problems

Hela Boukef, Mohamed Benrejeb, Pierre Borne

Abstract: A new genetic algorithm coding is proposed in this paper to solve flow-shop scheduling problems. To show the efficiency of the considered approach, two examples, in pharmaceutical and agro-food industries are considered with minimization of different costs related to each problem as a scope. Multi-objective optimization is thus, used and its performances proved.

Keywords: genetic algorithm, operations coding, flow-shop problems, multi-objective optimization, pharmaceutical industries, agro-food industries.

1 Introduction

In shop scheduling, there are three basic models classified according to the structure of a processing route:

- the job-shop scheduling problem is an operation sequencing problem on multiple machines subject to some precedence constraints among the operations,
- the flow-shop scheduling problem is a set of jobs that flow through multiple stages in the same order,
- the open-shop scheduling problem is a problem where the workshop has several resources and the routing of all the operations is free.

In pharmaceutical and agro-food industries, the tasks execution needs the use of several resources in a single order. It is thus about a flow-shop problem.

Flow-shop problems have received considerable attention from researchers during the last decades and the scheduling criterion most frequently used was the maximum completion time [3],[7], [4], [21], [15], [26]...

However, the analysis of the performance of a schedule often involves more than one aspect and therefore requires a multi-objective treatment [8], [17], [25]. The aim of multi-objective optimization is to deal with many criteria at the same time.

This paper is focused on minimizing the total costs related to manufacturing and delivery processes. To solve this kind of problems, the exact methods and the approached methods can be applied.

The exact methods, such as branch and bound [1] and linear programming methods [23], concern small size problems at the contrary of approached methods, such as tabu search [10], simulated annealing [16], genetic algorithms [13] and ants colony methods [6], that concern big size ones [14]. In this paper, we focus on the use of genetic algorithms method.

The principle scope of this method, based on natural selection mechanism, is the improvement of robustness and balance between cost and performance [11].

The genetic algorithms became famous due to their efficiency in solving combinatory optimization problems [19]. Their application fields are very important. They vary from complex real application such as pipelines flow control or robot planning path to theoretical combinatory problems.

The paper is organized as follows. First, the notations are introduced. Section 2 deals with the presentation and formulation of pharmaceutical and agro-food scheduling problems. Section 3 tackles the choice of the use of multi-objective evaluation. Section 4 handles with the presentation of genetic

algorithms and the proposed structured coding of list operations. In section 5, two examples, concerning agro-food and pharmaceutical industries scheduling, are treated by using this algorithm.

Notations :

- P_i : finished product after operation O_i
 p_{ik} : conditioning time of O_i on machine k
 $C_{P_{ik}}$: ending processing time of P_i on machine k
 $d_{P_i}^{liv}$: delivery date of P_i
 $C_{P_i}^{stk}$: storage cost by time unit of P_i

For pharmaceutical industries:

- $t_{p_{ik}}$: preparation time on machine k before O_i
 t_{ik}^{arr} : stoppage time during O_i on machine k
 C_{prod}^{tot} : total production cost
 C_k^u : production cost by time unit on machine k
 DO_{ik}^{nett} : cleaning operations time on machine k
 DO_{ik}^{chf} : size changing operations time on machine k
 Cr_{d_i} : costs of distribution delays of P_i
 C_{f_i} : manufacturing cost of P_i

For agro-food industries:

- t_i : effective start time of manufacturing O_i
 r_i : earliest start time of O_i
 γ_i : effective end time of O_i
 c_{ik} : k^{th} component from the whole components of O_i
 v_{ik} : limit validity date of a component c_{ik}
 DV_{P_i} : lifespan of P_i
 DR_{P_i} : back delay of P_i
 P_{ik}^{rev} : income of a component c_{ik}
 $P_{P_i}^{ven}$: unit sale price of P_i

2 Presentation and formulation of pharmaceutical and agro-food scheduling problem

2.1 Pharmaceutical industries scheduling case

In pharmaceutical industries, many problems can appear in production workshop. Several operations of cleaning and format changes have to be managed jointly with the manufacturing operations. The unproductive times generated by these operations are rather significant, taking into account the fact that the time launching of a product manufacturing depends on that which precede it. From a product to another, the change involves certain modifications on the level of each machine. These problems can brake the beginning of the production and delay its end.

Moreover, stoppage time due to machines break down, and production time are as many factors that breed important production costs [2].

Costs of distribution delays can also be calculated taking account of storage costs, date of production and end delivery date.

Let consider objective functions f_1 and f_2 . They represent the minimization of total cost of production, C_{prod}^{tot} , and the minimization of distribution delays cost of the product P_i , Cr_{d_i} .

The total cost of production has the following global expression:

$$C_{prod}^{tot} = \sum_{1 < j < N} C f_j$$

N is the number of manufactured products [22].

In pharmaceutical industries, this expression becomes:

$$f_1 = C_{prod}^{tot} = \sum_k C_k^u \sum_i w_{ik} (p_{ik} + t p_{ik}^{arr}) \tag{1}$$

with:

$$t p_{ik}^{arr} = DO_{ik}^{nett} + DO_{ik}^{chf}$$

w_{ik} : the coefficient of use of the machine k for the production of the product i , $w_{i,k} = \{0, 1\}$

The distribution delays penalties has the following expression:

$$f_2 = C_{rd_i} = \max(0, d_{P_i}^{liv} - C_{P_i}) C_{P_i}^{stk} \tag{2}$$

2.2 Agro-food industries scheduling case

Among the different problems occurring in agro-food industries let distinguish products perishability and distribution discount.

Products perishability is a major problem in agro-food industries because of products short expiry dates. The distribution discount contains penalties applied to sellers and storage costs of final products before their delivery. This criterion depends on the end-products storage's period of time before its expedition to the distribution areas [9].

Generally, expired product's costs, g_1 , and distribution discount costs, g_2 , are expressed by the following expressions:

$$g_1 = \sum_k P_{ik}^{rev} \left(\frac{\max(0, t_i - v_{ik})}{(t_i - v_{ik})} \right) \tag{3}$$

$$g_2 = \max(0, d_{P_i}^{liv} - C_{P_i}) \times \left(\frac{P_{P_i}^{ven}}{DV_{P_i} - DR_{P_i}} + C_{P_i}^{stk} \right) \tag{4}$$

Usually, resolution approaches take account of one criterion at once. In this paper, we want to optimize a trade off between several criteria at the same time and to find a mean of global evaluation of these criteria.

3 Multi-objective evaluation

Weighting sum function method, one of the different multi-objective methods, consists in combining linearly the different functions applying a weighted coefficient to each of them and summing them [5]. We can also say that it is a weighted linear combination of the objectives and it is used to aggregate the considered objectives in a single one.

Pharmaceutical industries case [3]

The weighted sum of functions f_1 and f_2 gives the following function f_{eq} :

$$f_{eq} = \alpha_1 f_1 + \alpha_2 f_2 \tag{5}$$

then:

$$f_{eq} = \alpha_1 \left(\sum_k C_k^u \sum_i w_{ik} (p_{ik} + t p_{ik} + t_{ik}^{arr}) \right) + \alpha_2 \left(\max(0, d_{P_i}^{liv} - C_{P_i}) C_{P_i}^{stk} \right) \tag{6}$$

where α_i are the coefficient that privilege one function instead of another, $\alpha_1 + \alpha_2 = 1$.

Agro-food industries case [24]

The weighted sum g_{eq} of functions g_1 and g_2 is expressed as following:

$$g_{eq} = \beta_1 g_1 + \beta_2 g_2 \quad (7)$$

then:

$$g_{eq} = \beta_1 \sum_k P_{ik}^{rev} \left(\frac{\max(0, t_i - v_{ik})}{(t_i - v_{ik})} \right) + \beta_2 \max(0, d_{P_i}^{liv} - C_{P_i}) \times \left(\frac{P_{P_i}^{ven}}{DV_{P_i} - DR_{P_i}} + C_{P_i}^{stk} \right) \quad (8)$$

where β_i are the coefficients that privilege one function instead of another, $\beta_1 + \beta_2 = 1$.

Genetic algorithms are adapted from natural systems and are successfully used in artificial systems. They proof themselves in optimization fields and in multiple other application fields [17]. That's why they are adopted and then proposed for our scheduling problems resolution.

4 Proposed operations coding used in genetic algorithms application

Genetic algorithms are parts of evolutionary algorithms that are composed of three essential elements:

- a population made up of several individuals representing the potential solutions (configurations) of a given problem,
- an evaluation mechanism of each individual adaptation regard to his external environment,
- evolution operators allowing the elimination of certain individuals and the creation of new ones [12].

4.1 Genetic algorithms

Genetic algorithms are iterative algorithms whose aim is to optimize a function called fitness [13]. They are exploration algorithms based on natural selection mechanisms and genetics and use at the same time survival of the best adapted structures principles and pseudo-random exchanges information, to form an exploration algorithm which has certain characteristics of human exploration. With each generation, a new set of individuals is created by using best elements parts of the precedent generation as well as innovating parts.

The genetic algorithms are not purely random. They effectively exploit information obtained previously to speculate in the position of new points to explore, with the hope to improve the performance [11].

To realize this scope, they use a set of points, called individual population, where each individual represents a possible solution of the given problem and contains different elements, called gene, which can take multiple values [20]. The different operators used in genetic algorithms are selection, crossover and mutation.

The aim of selection operator is to pick individuals that can survive and/or reproduce themselves to transmit their characteristics to the next generation. The selection operator is based on conservation principle of the most adapted individuals and elimination of the less ones [12]. No selection operator is absolutely perfect; a risk of favouring a number of individuals always exists and it could be a real drawback.

The crossover operator ensures the combination of parental genes to form new descendants with new potentials. This operator works randomly according to a probability fixed by the user keeping count of the optimization problem [20].

The mutation operator consists in a random changing for an individual certain gene's values [12]. Without this operator, a uniform population is produced.

A suitable coding choice is an important task that can guarantee the success of the genetic algorithms application.

The classic coding introduced by Holland [13] corresponds to the binary alphabet (0/1) where the chromosome is simply represented by a finite 0-1 array. More generally, genes can be real, characters or any expressions or elementary entities [14].

4.2 Structured list operations coding

In the case related to pharmaceutical industries scheduling, cleaning and size changing operations as well as stoppage and preparation times generate very high production costs. Distribution delays can also occur and generate significant costs.

In the case of agro-food industries scheduling, products expiry dates and storage time before delivery have to be taken into account for perishability costs and distribution discount costs evaluation.

Coding to be implemented must thus be able to deal with management time problems, precedence constraints, assignment resources, and the different costs that result from these problems.

SLOC's presentation

Inspired by Parallel Machine Coding CPM [20] and List Operations Coding CLO [14], the proposed Structured List Operations Coding SLOC, offers to solve the previously quoted problems.

The *one point crossover* operator for the SLOC, selects two individual parents (according to the technique of the caster) and a crossover point. First child receives from first parent the chromosomes preceding the crossover point (same thing for second child and second parent) and it receives from the second parent chromosomes following the crossover point (same thing for second child 2 and first parent). Updates are then carried out.

The *two points crossover* operator for the SLOC chooses two individual parents and two crossover points. First child receives from first parent the chromosomes preceding the first crossover point and following the second crossover point (same thing for the second child and second parent) then, it receives from the second parent chromosomes located between the two crossover points without redundancy (same thing for second child and first parent). Updates are then carried out.

When it is about only one machine, the *mutation operator* chooses one individual parent and two mutation points, the permutation of the two chromosomes produces a new individual.

When they are two (or more) machines, this operator chooses one individual parent and a mutation point, the operation thus consists in changing for the individual concerned the number of the machine and making the necessary updates.

The use of these operators has as principle concern the improvement of robustness and the balance between costs and performances.

Algorithm

The genetic algorithm steps, applied to pharmaceutical and agro-food scheduling, is given in figure 1.

SLOC's structure

In the case of only one machine, an individual is composed of a list of products, and for each product a cost function is associated in table 1 where List i is a set of products placed in a well defined ordered

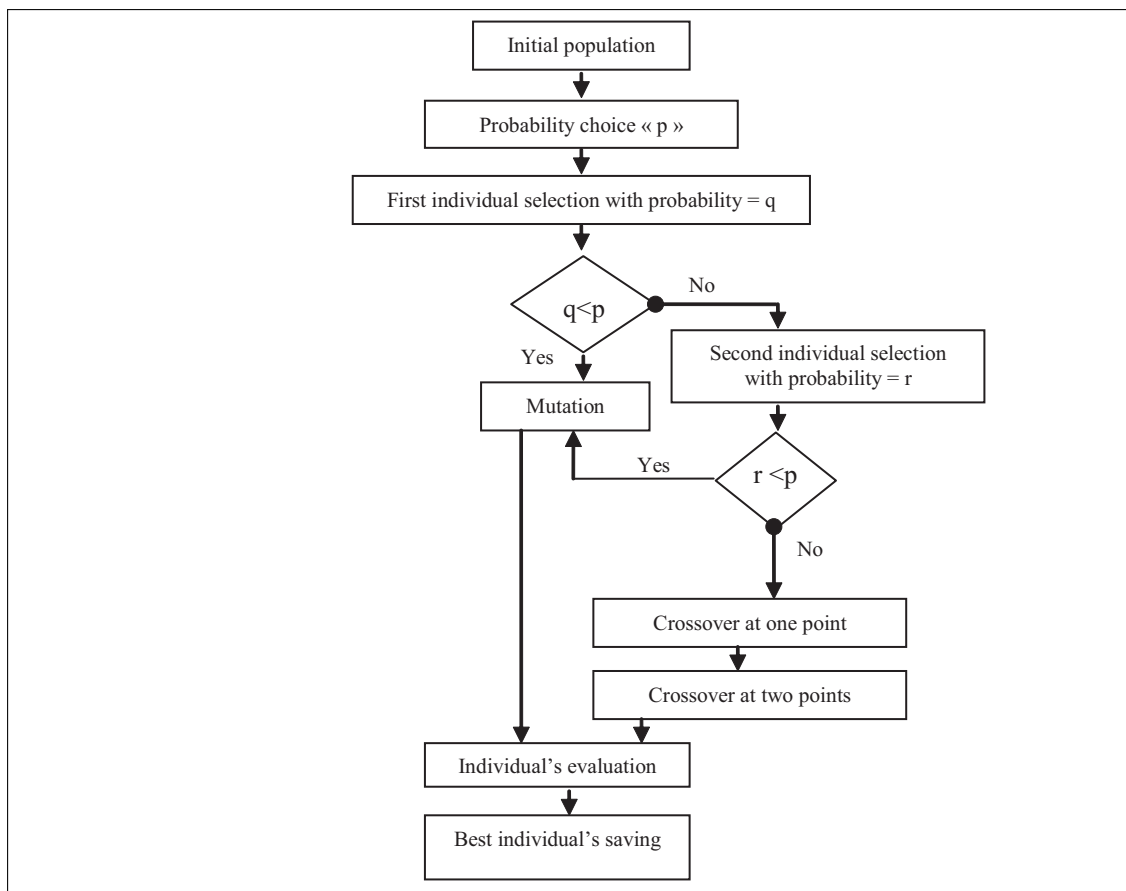


Figure 1: Genetic algorithm steps

passage, $i = 1, 2, \dots, n$ and $f_{eq}(List_i)$ is an equivalent function corresponding to the weighted sum of the two considered cost functions, $i = 1, 2, \dots, n$.

Table 1: : Scheduling data in agro-food industries

List of products	f_{eq}
$List_1$	$f_{eq}(List_1)$
$List_2$	$f_{eq}(List_2)$
...	...
$List_n$	$f_{eq}(List_n)$

In the case of two or several machines, an individual is also composed of a list of products but, in this case, to each product corresponds a data structure taking into account the number of the machines on which the product is manufactured, of its beginning execution time and of its end time.

As in the case of only one machine, to each individual (list of products) is associated the function corresponding to the weighted sum of both or several cost functions, showed in table 2 and using the following notations:

- nM_i : machines number in which the product P_i is manufactured
- t_i : beginning time of product P_i manufacturing
- C_i : time ending of product P_i manufacturing
- f_{eq} : equivalent function to optimize
- ind_i : given individual of a given generation

Table 2: SLOC coding for n machines and m products for a given individual and a given generation

<i>Product</i> ₁	<i>nM</i> ₁	<i>t</i> ₁	<i>C</i> ₁	<i>f_{eq}(ind_i)</i>
...				
<i>Product</i> _{<i>j</i>}	<i>nM</i> _{<i>j</i>}	<i>t</i> _{<i>j</i>}	<i>C</i> _{<i>j</i>}	
...				
<i>Product</i> _{<i>n</i>}	<i>nM</i> _{<i>n</i>}	<i>t</i> _{<i>n</i>}	<i>C</i> _{<i>n</i>}	

5 Agro-food and pharmaceutical scheduling resolution

In this section, we propose to solve a single machine flow-shop problem in agro-food industries and a two-machines flow-shop problem in pharmaceutical industries. The scope is the minimization of different costs, appropriate to each scheduling problem.

5.1 First example: Resolution of a flow-shop scheduling problem in agro-food industries

The flow-shop scheduling problem data in agro-food industries in the case of one machine is presented in the following table 3. Taking into account the data presented in this table, products perishability and distribution discount costs can be calculated following expressions 3 and 4.

Starting from g_{eq} function, the genetic algorithms are applied to solve the scheduling problem in order to obtain the products best list which minimizes this function, by using structured lists operation coding. Indeed, for 200 generations and a 20 individuals population per generation, mutation and crossover operations are carried out, thus generating, new individuals whose new costs are calculated. Then, the best individual is progressively generated and saved.

Table 3: Scheduling data in agro-food industries

Operation name	Component name	r_i	p_i	v_{ik}	P_{ik}^{rev}	P_{Pi}^{ven}	DV_{Pi}	DR_{Pi}	d_{Pi}^{liv}	C_{Pi}^{stk}
Op1	C11	1	3	3	8	4	5	11	10	6
Op1	C12	1	3	8	10	4	5	11	10	6
Op1	C13	1	3	2	7	4	5	11	10	6
Op2	C21	2	5	5	13	5	12	10	19	2
Op2	C22	2	5	12	11	5	12	10	19	2
Op2	C23	2	5	11	4	5	12	10	19	2
Op3	C31	2	3	5	6	6	15	12	22	7
Op3	C32	2	3	20	2	6	15	12	22	7
Op3	C33	2	3	3	9	6	15	12	22	7
Op4	C41	3	2	17	11	8	11	15	11	14
Op4	C42	3	2	10	2	8	11	15	11	14
Op4	C43	3	2	4	5	8	11	15	11	14
Op5	C51	3	2	11	14	7	8	10	15	18
Op5	C52	3	2	15	12	7	8	10	15	18
Op5	C53	3	2	4	23	7	8	10	15	18

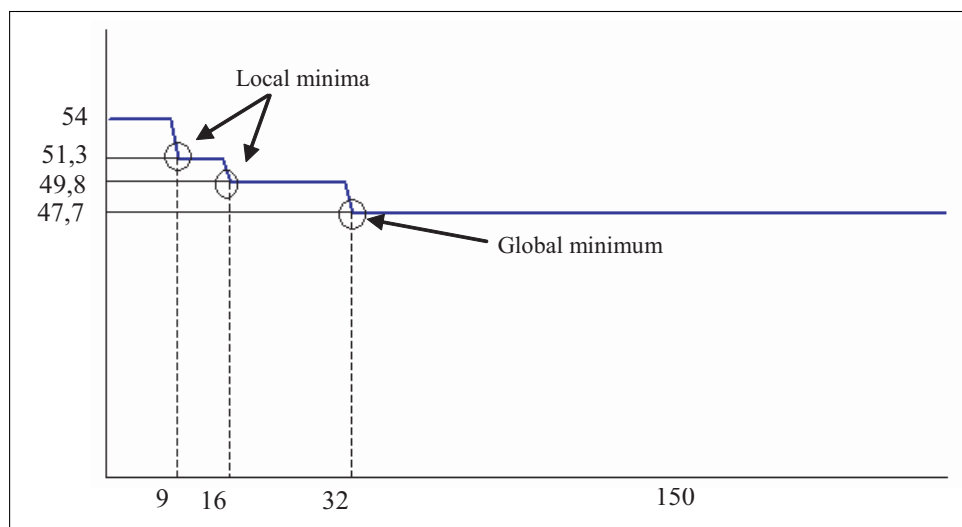


Figure 2: Total cost evolution through the generations in agro-food industries

The figure 2 shows that the cost of the best individual of the initial population is 54, then, the first local minimum observed shows that the cost of the best individual passed to 51,3 and this in the ninth generation. For the second local minimum, the total cost passes to 49,8 in the sixteenth generation to stabilize itself in the thirty second generation with 47,7 which is the observed global minimum.

5.2 Second example: Resolution of a flow-shop scheduling problem in pharmaceutical industries

The flow-shop scheduling problem data in pharmaceutical industries, in the case of two machines M1 and M2, is presented in the following table 4.

Table 4: Scheduling data in pharmaceutical industries

Product name	p_{i1}	t_{i1}^{arr}	tp_{i1}	p_{i2}	t_{i2}^{arr}	tp_{i2}	$d_{p_i}^{liv}$
P01	1800	240	120	2880	100	100	3
P02	2400	310	150	—	—	—	20
P03	2250	240	135	2100	115	100	14
P04	1950	175	130	2300	135	115	11
P05	2800	300	120	—	—	—	10
P06	1750	150	100	1500	115	110	25
P07	—	—	—	2400	120	110	15
P08	3500	315	180	3000	250	110	25
P09	2200	270	120	2500	110	150	45
P10	1000	115	165	—	—	—	32
P11	1850	155	180	1650	155	130	12
P12	2120	180	200	2400	160	120	56
P13	3300	210	140	—	—	—	15
P14	4500	190	175	3800	150	190	60
P15	—	—	—	1600	100	100	35
P16	6750	180	120	7000	220	100	45

Times of manufacturing, stoppage and preparation are indicated in minutes and the various dates are calculated compared to an initial time t_0 .

Times and dates comprising the symbol " — " indicate the impossibility of manufacturing the concerning product on the corresponding machine.

Taking into account the data presented in the table 4, the production cost and the distribution delays penalties can be calculated while following expressions 1 and 2.

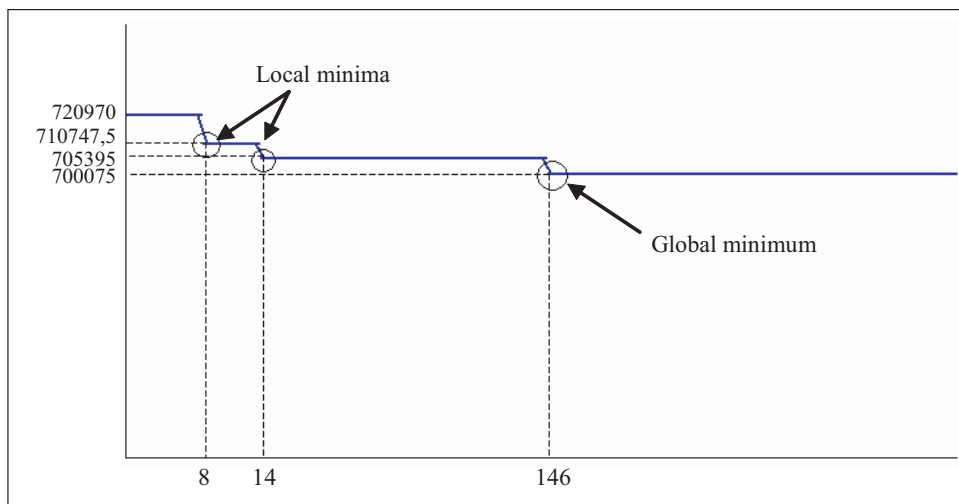


Figure 3: Total cost evolution through the generations in pharmaceutical industries

Starting from function f_{eq} , the genetic algorithms are applied to solve the scheduling problem in order to obtain the products best list which minimizes this function, by using proposed structured lists operation coding. Indeed, for 200 generations and 20 individuals population per generation, mutation and

crossover operations are carried out, thus generating, new individuals whose new costs are calculated. Then, the best individual is progressively generated and saved.

The figure 3 shows that the cost of the best individual of the initial population is 720970, then, the first local minimum observed shows that the cost of the best individual passed to 710747,5 and this in the eighth generation. For the second local minimum, the total cost passes to 705395 in the fourteenth generation to stabilize itself in the hundred forty sixth generation with 700075 which is the observed global minimum.

6 Conclusion

In this paper, a multi-objective approach was adopted for the flow-shop scheduling problems resolution in agro-food and pharmaceutical industries. The proposed structured list operations coding and its use to implement genetic algorithms showed the capacity of this type of algorithms to find the global minimum aimed. It is interesting to make a comparison of the effectiveness of the choice of this suggested coding with other codings.

Bibliography

- [1] P. Baptiste and C. Le Pape “ A constraint-Based Branch and Bound Algorithm for Preemptive Job-Shop Scheduling ”. 5th IEE, International Symposium on Assembly and Task Planning, Besançon, 1996.
- [2] H. Boukef, F. Tangour et M. Benrejeb “ Sur la formulation d’un problème d’ordonnancement de type flow-shop d’ateliers de production en industries pharmaceutiques ”. Journées Tunisiennes d’Electrotechnique et d’Automatique, JTEA’06, Hammamet, 2006.
- [3] J.L. Bouquard, J.C. Billaut, M.A. Kubzin and V.A. Strusevich “ Two-machine flow-shop scheduling problems with no-wait jobs ”. Operations Research Letters, vol.33, pp. 255-262, 2004.
- [4] J. Breït “ A polynomial-time approximation scheme for the two-machine flow shop scheduling problem with an availability constraint ”. Computers and Operations Research, vol.33, pp. 2143–2153, 2005.
- [5] Y. Collette et P. Siarry “ Optimisation Multiobjectif ”. Editions Eyrolles, Paris, 2002.
- [6] A. Colorni, M. Dorigo, V. Maniezzo and M. Trubian “ Distributed optimization by ant colonies ”. First European Conference on Artificial Life, Paris, pp. 134-142, 1991.
- [7] C. Esswein, J.C. Billau, and V.A. Strusevich “ Two-machine shop scheduling: Compromise between flexibility and makespan value ”. European Journal of Operational Research, vol.167, pp. 796–809, 2004.
- [8] F.G. Filip, G. Neagu, and D. A. Donciulescu “ Job-shop scheduling optimization in real-time production control ”. Computers in Industry, vol. 4, pp. 395-403, 1983.
- [9] E. Gargouri, S. Hammadi and P. Borne “ A distributed scheduling for agro food manufacturing problems ”. Journal of IEEE Transactions on Systems, Man and Cybernetics, vol.33, 2003.
- [10] F. Glover “ Tabu search, part II ”. ORSA, Journal of Computing, vol.2, pp. 24-32, 1989.
- [11] G.E. Goldberg “ Algorithmes génétiques ”. Editions Addison-wesley, France, 1994.

- [12] J.K. Hao, P. Galinier et M. Habib “ Métaheuristiques pour l’optimisation combinatoire et l’affectation sous contraintes ”. *Revue d’Intelligence Artificielle*, pp. 2-39, 1999.
- [13] J.H. Holland “ *Adaptation in natural and artificial systems* ”. PhD, Michigan Press Univ., Ann Arbor, MI, 1975.
- [14] I. Kacem, S. Hammadi and P. Borne “ Flexible job-shop scheduling problems: formulation, lower-bounds, encodings, and controlled evolutionary approach ”. *Computational Intelligence in Control*, Idea Group Publishing, 2003.
- [15] P.J. Kalczynski and J. Kamburowski “ On no-wait and no-idle flow shops with makespan criterion ”. *European Journal of Operational Research*, vol.178, pp. 677–685, 2006.
- [16] S. Kirkpatrick and M.P. Vecchi “ Optimization by simulated annealing ”. *Science*, vol.220, pp. 671-680, 1983.
- [17] T. Loukil, J. Teghem and D. Tuytens “ Solving multi-objective production scheduling problems using metaheuristics ”. *European Journal of Operational Research*, vol.161, pp. 42–61, 2003.
- [18] E. Lutton “ *Etat de l’art des algorithmes génétiques* ”, INRIA, pp.137-140, 1994.
- [19] M.H. Mabed, M. Rahoual, E.G. Talbi and C. Dhaenens “ Algorithmes génétiques multicritères pour les problèmes de flow-shop ”. 3ème Conférence Francophone de MOdélisation et de SIMulation, MOSIM’01, Troyes, pp. 843-849, 2001.
- [20] K. Mesghouni, S. Hammadi and P. Borne “ On modeling genetic algorithm for flexible job-shop scheduling problem ”. *Studies in Informatics and Control Journal*, vol.7, pp. 37-47, 1998.
- [21] A. Oulamara “ Makespan minimization in a no-wait flow shop problem with two batching machines ”. *Computers and Operations Research*, vol.34, pp. 1033–1050, 2005.
- [22] I. Saad, S. Hammadi, P. Borne and M. Benrejeb “ Aggregative Approach for the Multiobjective Optimization Flexible Job-Shop Scheduling Problems ”. *The IEEE International Conference on Service Systems and Service Management, IEEE-ICSSSM’06*, pp. 889-894, Troyes, 2006.
- [23] M. Sakarovitch “ *Graphes et Programmation Linéaire* ”. Edition Hermann, Paris, 1984.
- [24] F. Tangour, S. Hammadi, P. Borne and M. Benrejeb “ Ordonnancement dynamique dans un atelier de production agroalimentaire ”. Séminaire d’Automatique-Industrie, SAI’06, Matmata, 2006.
- [25] F. Tangour and I. Saad “ Multiobjective optimization scheduling problems by Pareto-optimality in agro-alimentary workshop ”. *International Journal of Computers, Communications & Control, IJCCC*, vol. I, n°. 3, pp. 71-83, 2006.
- [26] J. Yuan, A. Soukhal, Y. Chen and L. Lu “ A note on the complexity of flow shop scheduling with transportation constraints ”. *European Journal of Operational Research*, vol. 178, pp. 918–925, 2006.

Hela BOUKEF^{1,2}, Mohamed BENREJEB¹, Pierre BORNE²

¹Ecole Nationale d’Ingénieurs de Tunis
Unité de recherche LARA-Automatique
BP 37, Le Belvédère, 1002 Tunis, Tunisie

²Ecole Centrale de Lille, Cité scientifique
Laboratoire d’Automatique, Genie Informatique et Signal

BP 48, 59651 Villeneuve d'Ascq Cedex, France

E-mail: hela.boukef@yahoo.fr, mohamed.benrejeb@enit.rnu.tn, pierre.borne@ec-lille.fr



Hela Boukef was born in Carthage, Tunisia in 1981. She graduated from "Institut Supérieur de Gestion de Tunis" in 2003 and obtained the Master of automatic and signal treatment in 2006 at the "Ecole Nationale d'Ingénieur de Tunis". She is currently preparing the Ph.D. degree in automatic and computer science within the framework of LAGIS-EC-Lille and LARA-ENIT cooperation. Her research is related to optimization methods for discrete events systems, computer science and operational research.



Mohamed Benrejeb was born in Tunisia in 1950. He obtained the Diploma of "Ingénieur IDN" (French "Grande Ecole") in 1973, The Master degree of Automatic Control in 1974, the PhD in Automatic Control of the University of Lille in 1976 and the DSc of the same University in 1980. Full Professor at "Ecole Nationale d'Ingénieurs de Tunis" since 1985 and at "Ecole Centrale de Lille" since 2003, his research interests are in the area of analysis and synthesis of complex systems based on classical and non conventional approaches.



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, of the University of Waterloo (Canada) in 2006 and of the polytechnic University of Bucarest (Romania). He is author or co-author of about 200 Journal articles and book chapters, and of 34 plenary lectures and of more than 250 communications in international conferences. He has been the supervisor of 66 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.