LOWER BOUNDS FOR THE SCHEDULING PROBLEM WITH UNCERTAIN DEMANDS

DJAMEL BERKOUNE*,**, KHALED MESGHOUNI*
BESOA RABENASOLO**

* LAGIS, UMR CNRS 8146, Ecole Centrale de Lille BP 48, 59651 Villeneuve d'Ascq Cedex, France
e-mail: khaled.mesghouni@ec-lille.fr

** GEMTEX EA 2461, Ecole Nationale Supérieure des Arts et Industries Textiles 9
rue de l'Ermitage, BP 30329, 59056 Roubaix Cedex 01, France
e-mail: {djamel.berkoune,besoa.rabenasolo}@ensait.fr

This paper proposes various lower bounds to the makespan of the flexible job shop scheduling problem (FJSP). The FJSP is known in the literature as one of the most difficult combinatorial optimisation problems (NP-hard). We will use genetic algorithms for the optimisation of this type of problems. The list of the demands is divided in two sets: the actual demand, which is considered as certain (a list of jobs with known characteristics), and the predicted demand, which is a list of uncertain jobs. The actual demand is scheduled in priority by the genetic algorithm. Then, the predicted demand is inserted using various methods in order to generate different scheduling solutions. Two lower bounds are given for the makespan before and after the insertion of the predicted demand. The performance of solutions is evaluated by comparing the real values obtained on many static and dynamic scheduling examples with the corresponding lower bounds.

Keywords: flexible job shop scheduling, insertion, makespan, predicted demands, lower bounds

1. Introduction

In order to solve a scheduling problem, one needs to find an adequacy between the tasks to be processed in the workshop, and the available means to carry out the production. The objective is to allocate resources to operations or tasks in order to optimize some criteria under various constraints of workshops: the duration of the operations, the availability of resources, priority constraints, and precedence constraints. There are two types of methods to solve this kind of problems (Alvarez-Valdes *et al.*, 1987; Brucker, 2003; Carlier and Chrétienne, 1988; Demeulemeester and Herrolein, 1990; Pinedo, 2002). The first group are exact methods such as the branch-and-bound algorithm, which are able to find optimal solutions but are not efficient for problems of a larger size because of their computation times. The other type of methods includes approximate methods, which are faster but the obtained solution is not guaranteed to be optimal. These methods are usually applied when the scheduling problem is known to be NP-hard.

Our previous work tackles the problem of job shop scheduling with partially and completely flexible devices (Mesghouni, 1999). In order to obtain solutions under a reasonable computation time, genetic algorithms can provide effective results for scheduling problems (Della Croce *et al.*, 1995; Kobayashi *et al.*, 1995; Mattfeld and Bierwirth, 2004; Mesghouni, 1999; Ponnambalam *et al.*, 2001; Sevaux and Dauzère-Pérès, 2003). As introduced by Holland (Goldberg, 1989; Holland, 1992), these genetic algorithms are able to find solutions very close to the global optimum. Their principle is to evolve an initial set of solutions to a final one while making a total improvement according to a criterion fixed as a preliminary. The use of these algorithms requires the definition of a problem coding and genetic operators such as crossover and mutation. The rate of convergence and the quality of the solution are improved with a coding specifically conceived for problems of scheduling as presented in (Mesghouni, 1999; Mesghouni *et al.*, 2004).

In our previous work, we set up a new approach to solve the problem of a flexible job shop with uncertain demands (Berkoune *et al.*, 2004; Mesghouni and Rabenasolo, 2002). Among various heuristic methods that were proposed to solve these problems, a certain number of heuristics of valid insertion of additional tasks on cumulative resources were presented in (Artigues, 1997; Artigues *et al.*, 2003). There exist several models of uncertainty: a machine breakdown and, more generally, uncertainty on the availability of resources, uncertainty on the processing time, on the release date, or on the delivery time. In our problem, we will study uncertainty on the demands: the possibility to sell or not to sell a product whenever it is manufactured in advance without a customer order. This situation arises, e.g., in production systems with a make-to-stock and make-to-order mixed strategy. The list of the demands is divided into the actual demands, which are scheduled in priority by the genetic algorithm, and the predicted demands, which are inserted in the remaining availability of the production resources, by using various insertion methods. These predicted demands are necessary for exploiting better the availabilities of the machines and increasing their productivity, and also to minimize the idle periods and cost of restarting the machines. The objective is to minimize the impact of insertion over the total duration of the production. In order to measure the efficiency and robustness of the approach, we should compare the results obtained by this method with the optimal one. But since this optimal solution is not always available, we should compare it with its lower bounds.

Thus, this paper will propose a method of calculating lower bounds for the makespan by extending the procedure of (Carlier, 1987) for the problem of job shop scheduling in the case of predicted scheduling. The calculation of lower bounds was used in the literature for several scheduling problems, in particular, the single machine problem (Carlier, 1982), parallel machines (Carlier, 1987), hybrid problems of the flow shop (Billaut *et al.*, 2002) and job shops scheduling (Carlier and Pinson, 1989). Generally, these methods are based on the relaxation of a set of constraints (the preemption of tasks, the disjunctive constraint) to underestimate the makespan of optimal scheduling.

This paper is organized as follows: In the second section we present the specifity of the flexible job shop, and the notation we use throughout the paper. Some solution methods used to deal with such a problem are given in Section 3. In the fourth section we propose a new method for calculating the lower bounds of the predicted demands. Finally, the last sections will be devoted to the presentation of the results on illustrative examples and the conclusions concerning this research work.

2. Problem Statement and Notation

The flexible job shop scheduling problem consists in the organization of the execution of N jobs on M machines. Each job J_j represents a number n_j of operations. Each operation i of a job J_j denoted by O_{ij} can be carried out on a machine k with a duration p_{ijk}, which depends on this machine. The actual processing time for each operation will only be fixed after the final assignment on a given machine.

Notation:

N	the total number of jobs;
n	the number of firm jobs, and $N - n$ the number of predicted jobs, $n \leq N$;
p_{ijk}	the processing time for the operation O_{ij} on the machine k;
γ_{ij}	the shortest execution time of the operation $O_{ij} : \gamma_{ij} = \min_{1 \leq k \leq M}(p_{ijk})$;
$C_{f\max}$	the makespan of the firm jobs;
$C_{p\max}$	the makespan of the predictive jobs;
$C_{f\max^*}$	the lower bound to the makespan of the firm jobs;
$C_{p\max^*}$	the lower bound to the makespan of the predictive jobs;
r_{ij}	the earliest beginning date/release date of the operation O_{ij};
r_j	the earliest beginning date of the job j, $r_j = r_{1,j}$, $\forall 1 \leq j \leq N$;
r'_k	the earliest date of the availability of the machine k;
$[A, B]_k$	the availability interval of the machine k;
$\mathcal{R}_k$	the set of possible starting dates on the machine k (heads of the availability interval);
TP_{sk}	the s-th element in $\mathcal{R}_k$.

In this problem, we make the following hypotheses:

- All the machines are available at r'_k, $\forall 1 \leq k \leq M$;
- The firm jobs $J = \{J_1, \ldots, J_n\}$ will be scheduled in priority, and the predicted jobs $J' = \{J_{n+1}, \ldots, J_N\}$ will be inserted in the remaining availability of production resources.
- One machine can process only one operation at a given moment (resource constraints);
- A started operation runs to completion (no preemption condition);
- The following constraint of precedence exists between two successive operations of the job J_j: $r_{i+1,j} \geq r_{ij} + p_{ijk}$;
- All jobs are independent of each other.

The solution of such a problem thus requires two decisions: the assignment of each operation O_{ij} to one machine k, and the calculation of the date when to start each operation.

3. Solution Methods

The problem considered presents two main difficulties. The first one is the scheduling of the firm demands. The other difficulty is the insertion of the predicted operations in the existing solutions. Figure 1 shows the use of the different procedures for solving the scheduling problem.

3.1. Principal Scheduling. The list of demands is divided into the firm demands, which will be scheduled in accordance with their priorities by genetic algorithms, and the predicted demands, which will be inserted in previous solutions using various heuristic methods (Berkoune *et al.*, 2004). This approach is normally simpler to implement, and it can be justified by the priority consideration of the firm jobs and the optimal use of the machines by filling their available production capacities with the predicted jobs. These different procedures are applied to each change in the problem data, e.g., the change in the status of the predicted jobs into the firm jobs.

In order to evaluate better the quality of the solutions, we set up a whole range of lower bounds, and compare them with the results of total scheduling of the whole jobs $(J \cup J')$ in one global step without considering the priority of the firm jobs.

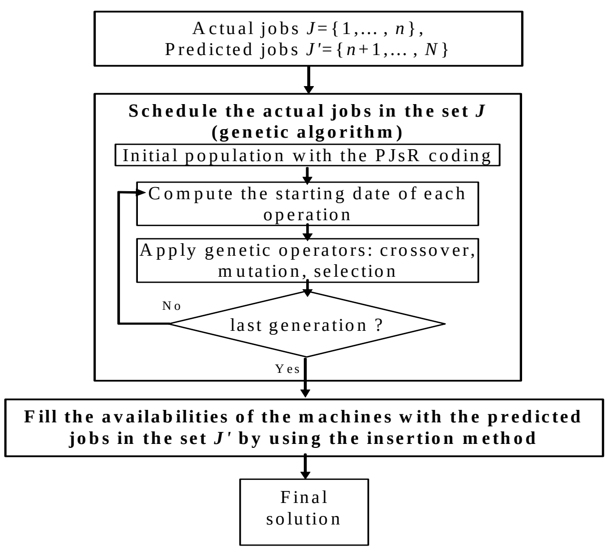

Fig. 1. Flow chart of the proposed scheduling algorithm.

3.2. Genetic Algorithm for the Scheduling Problem. Genetic algorithms (GAs) are algorithms which belong to the group of evolutionary algorithms. They allow us to make an initial set of solutions evolve to a final set of solutions while making a total improvement according to a fixed criterion. These algorithms were developed at the Michigan University by Holland and his co-workers in the 1960s (Goldberg, 1989; Holland, 1992). They are based on the natural selection mechanism in biological systems. GAs are iterative search algorithms where the goal is to optimize a preset function called the criterion or function "fitness", and work in parallel on a whole set of candidate solutions, called the "population" of individuals or chromosomes. The latter consist of a set of elements called "genes" which can take several values called "alleles" (Renders, 1995). A chromosome is a representation or a coding of a solution to the given problem in the form of a chain. The first population is selected either randomly, by some heuristics or methods that are specific to the problem, or still by a mixture of random and heuristic solutions. GAs generate new individuals so that they are more suitable or promising than their predecessors. The process of the improvement of individuals is carried out by genetic operators, which are: selection, crossover and mutation (Ponnambalam *et al.*, 2001; Renders, 1995; Syswerda, 1990).

3.2.1. Problem Coding. We use a representation of the chromosome called the Parallel Jobs Representation (PJsR), cf. (Mesghouni, 1999). In the genetic algorithm, the chromosome is represented by a matrix where each row is an ordered series of the operation sequence of this job, and each element of this row contains two terms. The first on is the machine which performs this operation and the other one is the starting time of this operation if the assignment of this operation to this machine is definitive. This starting time is calculated taking into account resource constraints (Goldberg, 1989; Syswerda, 1990). Genetic operators such as the crossover operator and the mutation operator have been adapted to the PJsR chromosome representation.

3.2.2. Genetic Operators. We define the following operators:

- *Crossover operators.* The goal of the crossover is to obtain, by a mixture of solutions, other chromosomes likely to improve the results. In our case, there are two operators of crossover (Mesghouni, 1999): the operator of the line crossover handles the jobs, and the operator of the column crossover handles a set of operations.
- *Mutation operators.* The essential role of the mutation is to introduce some diversification into the population which the crossover operator cannot bring.
- *Selection operators.* This operator consists in choosing the individuals from which one will create the next generation. In our case we used the principle of the roulette wheel, which reinforces the probability of retaining the most promising individuals in terms of the fitness function.

3.3. Algorithms of Insertion of the Predicted Jobs: This procedure uses the *Earliest Completion Time* rule (*ECT*). The *ECT* rule schedules in priority the operation with the earliest possible completion time among all unscheduled operations. This algorithm finds the position of insertion corresponding to the execution time of the operation i on the machine k by minimizing the total execution time under precedence constraints.

4. Lower Bounds

To solve the FJSP, we choose to use an approximate method, a genetic algorithm, which is likely to give a near-optimal solution. In order to measure the efficiency and robustness of our approach, we should compare the results obtained by our method with the optimal one. But since this optimal solution is not available, we should compare it with its lower bounds. We propose to calculate two lower bounds for the makespan, before and after the insertion of the predicted demand. These predicted lower bounds will permit us to choose the best solution for the insertion of the predicted jobs and to measure the performance of the proposed approach. Our proposition generalizes certain bounds offered in the literature for the problems of parallel machines. We consider non-preemption tasks, take into account precedence constraints and include the estimated cost of each predicted job. We associate two lower bounds $C_{f\max^*}$ and $C_{p\max^*}$ with each subset of the jobs J and J' which is not empty. In our problem, the lower bounds are calculated, on the one hand, for the firm jobs J by the method of calculation of the bounds which generalizes those proposed in (Carlier, 1987). On the other hand, we developed a new method of calculation of the lower bounds for the predicted jobs J', which is a function of the availability of the machines.

4.1. Lower Bounds for the Makespan of the Firm Jobs

Proposition 1. *The parameter*

$$C^1_{f\max^*} = \max_{1\le j\le n}\left(r_j + \sum_{i=1}^{n_j}\gamma_{ij}\right), \tag{1}$$

where $\gamma_{ij} = \min_{1\le k\le M}(p_{ijk})$*, constitutes a lower bound to the makespan of the firm jobs in the FSJP a problem.*

Proof. For any assignment of the operations to the machines, the makespan is the completion time of all operations, $C_{f\max} = \max\{C_j \mid j = 1,\dots,n\}$. However, for each operation, $p_{ijk} \ge \gamma_{ij} = \min_{1\le k\le M}(p_{ijk})$ by definition and, supposing that there is no waiting interval between two successive operations, we get the minorization (1) from

$$C_{f\max} \ge C_j \ge r_j + \sum_{i=1}^{n_j} p_{ijk} \ge r_j + \sum_{i=1}^{n_j}\gamma_{ij},$$

where k labels the machine assigned to O_{ij}. ■

Remark 1. If the cardinality of a set of jobs J is higher than the number of machines, or in the case of the relaxation of constraints such as preemption of tasks or a disjunctive constraint on resources, there is no solution which reaches the lower the bound in Proposition 1. In this case, we use another method for the calculation of a possible lower bound. This method is defined below.

Proposition 2. *The parameter*

$$C^2_{f\max^*} = \frac{1}{M}\left(\sum_{k=1}^{M} r'_k + \sum_{j=1}^{n}\sum_{i=1}^{n_j}\gamma_{ij}\right), \tag{2}$$

constitutes a lower bound to the makespan of the firm jobs in the FSJP.

Proof. Equation (2) gives a lower bound to the makespan of firm jobs for the FJSP in the case where the number of jobs N is higher than the number of machines M, knowing that each machine k has an availability time denoted by r'_k (in our case, $r'_k = 0,\ \forall 1 \le k \le M$). Using the same reasoning as in the classical calculation of the lower limits (Carlier, 1987), and by relaxing the non-preemption constraint of the tasks, we obtain the following minorization:

$$r'_1 + r'_2 + r'_3 \cdots + r'_k + \cdots + \sum_{j=1}^{n}\sum_{i=1}^{n_j}\gamma_{ij} \le MC_{f\max}.$$

Consequently,

$$C_{f\max} \ge \frac{1}{M}\left(r'_1 + r'_2 + \cdots + r'_k + \sum_{j=1}^{n}\sum_{i=1}^{n_j}\gamma_{i,j}\right).$$

The lower bound is the sum of the availability times of all the machines and the smallest processing time of all the operations divided by the number of the machines. ■

Concluding, the lower bounds (1) and (2) allow us to deduce limits for the criterion considered. These limits are defined by the following relationship:

$$\begin{aligned} C_{f\max^*} &= \max(C^1_{f\max^*}, C^2_{f\max^*}) \\ &= \max\left(\frac{1}{M}\Big(\sum_{k=1}^{M} r'_k + \sum_{j=1}^{n}\sum_{i=1}^{n_j}\gamma_{ij}\Big), \max_{1\le j\le n}\Big(r_j + \sum_{i=1}^{n_j}\gamma_{ij}\Big)\right). \end{aligned} \tag{3}$$

Equation (3) is a simple combination of Propositions 1 and 2.

4.2. Lower Bounds for the Makespan of the Predicted Jobs.

The lower bounds of the predicted jobs are a function of the remaining availability of the machines after scheduling the firm jobs.

Proposition 3. *The parameter*

$$C_{p\max^*} = \max_{n+1\le j\le N}\left(\sum_{i=1}^{n_j}\min_{1\le k\le M}\right.$$
$$\left.\times\left(p_{ijk}+\left(\min(\mathcal{R}_k)-\sum_{x=1}^{i-1}\gamma'_{x-1,j}\right)\right)\right) \qquad (4)$$

constitutes a lower bound to the makespan of predicted jobs in the FJSP with uncertain demands inserted, where $\mathcal{R}_k$ *is the set of the possibilities of the insertion times on the machine* k, $\gamma'_{0,j}=0$ *and* $\gamma'_{ij}=\min_k p'_{ijk}$, $p'_{ijk}=p_{ijk}+\min_k\mathcal{R}_k-\sum_{x=1}^{i-1}\gamma'_{x-1,j}$.

Proof. Assume that the time of the availability of the machine k is $D'_k\ge r'_k$. The equivalent processing time p'_{ijk} of the operation O_{ij} on the machine k is calculated below. First, define A representing the earliest possible time of starting O_{ij} for a specific machine availability time D'_k:

$$A=\begin{cases} r_{ij} & \text{if } D'_k\le r_{ij},\\ D'_k & \text{otherwise,}\end{cases} \qquad (5)$$

with r'_k being the release time of O_{ij}. We have two methods for calculating the lower limit of $C_{p\max}$.

If the interval of availability of the machine satisfies $[A,B]\ge p_{ijk}$, then a possible time of the insertion of O_{ij} on the machine k will be $TP_{gk}=A$, with g standing for the rank of insertion on the machine k. We obtain the final set of possible insertion times for each machine $\mathcal{R}_k=\{TP_{1k},TP_{2k}\ldots TP_{gk}\}$. Let $TP_{sk}=\min(\mathcal{R}_k)$, which we will use for all the operations inserted on this machine. In this case,

$$p'_{ijk}=p_{ijk}+TP_{sk}-\sum_{x=1}^{i-1}\gamma'_{x-1,j}, \qquad (6)$$

with $\gamma'_{0j}=0$ and $\gamma'_{ij}=\min_k p'_{ijk}$.

The definition of equivalent processing times including the times of waiting for the availability of the first machine is then

$$p'_{ijk}=p_{ijk}+\left(\min(\mathcal{R}_k)-\sum_{x=1}^{i-1}\gamma'_{x-1,j}\right). \qquad (7)$$

The available times $\mathcal{R}_k$ depend on the starting times (5). Each predicted job is processed during the interval of time $[0,C_{p\max}]$. We have

$$C_{p\max}\ge\max_{j\in J'}\left(\sum_{i=1}^{nj}\gamma'_{ij}\right), \qquad (8)$$

with

$$\gamma'_{ij}=\min_{1\le k\le M}\left(p'_{ijk}\right)$$
$$=\min_{1\le k\le M}\left(p_{ijk}+\left(\min(\mathcal{R}_k)-\sum_{x=1}^{i-1}\gamma'_{x-1,j}\right)\right).$$

■

Finally, a lower bound to the makespan of the FJSP with uncertain demands is

$$C_{\max^*}=\max\left(C_{f\max^*},C_{P\max^*}\right). \qquad (9)$$

5. Illustrative Examples

Let us illustrate the above results with examples of various sizes. Part of these tests with firm jobs comes from the literature (Berkoune *et al.*, 2004; Kacem, 2003; Mesghouni, 1999), while others were generated randomly. The example comprises a number of firm jobs n, between 3 and 22, and a number of predicted jobs n', between 2 and 10, with a number of machines between 3 and 10 with total flexibility. For each solution, we present the value of the lower bounds and the value of the criterion obtained.

Example 1. The first columns of Table 1 concerns the result produced by the genetic algorithm for the firm jobs while the last three columns show the actual $C_{\max}$ and its lower bounds after the insertion of the predicted jobs. The columns labelled as “Rate (%)” give the percentage of busy periods of machines over the interval $[0,C_{\max}]$. The other columns are expressed in time units (ut).

The solutions with only firm jobs show inactivity times which are evaluated between one fifth and one third of the time allowed for the manufacturing process. Accordingly, in order to increase the productivity rate and to reduce the inactivity period of the machines, we should use this period to add extra jobs.

The scheduling of all the jobs $(J\cup J')$ without considering the priority to the firm jobs gives the results in Table 2. Comparing the lower bounds of the scheduling methods (insertion and rescheduling) and those found with total scheduling by considering all the tasks on the same level of priority, we find that the differences between these bounds are small. These indicate the suitability of the obtained solutions.

Moreover, the small variations found between the values of the criteria and those of the corresponding lower bounds are due to the difficulty of the optimization problem which involves several non-homogeneous constraints such as the precedence constraints and the availabilities of machines. ♦

Table 1. Scheduling of firm jobs with GAs and the insertion of predicted jobs.

	$n \times n' \times M$	Scheduling of the firm jobs			After insertion of the predicted jobs		
		$C_{\max^*}$ (ut)	$C_{\max}$ (ut)	Rate (%)	$C_{\max^*}$ (ut)	C_{max} (ut)	Rate (%)
Ex01	3×3×3	6	7	71.42	10	11	87.88
Ex02	4×2×5	16	16	47.50	17	20	62.00
Ex03	5×3×5	17	17	71.76	20	22	78.18
Ex04	5×3×7	6	6	61.90	11	12	61.38
Ex05	6×6×6	8	9	83.33	15	16	80.13
Ex06	7×2×9	5	6	68.52	9	11	70.00
Ex07	7×6×5	10	11	83.63	17	19	88.42
Ex08	7×6×8	6	7	67.85	12	12	68.75
Ex09	10×5×10	7	7	67.15	8	9	83.33
Ex10	10×2×7	15	16	66.07	17	17	72.26
Ex11	10×10×8	7	9	76.39	11	14	85.72
Ex12	12×7×10	7	7	74.28	8	10	83.00
Ex13	20×7×10	8	9	92.22	12	13	93.84
Ex14	22×5×10	9	11	90.00	12	14	90.00

Table 2. Scheduling of all jobs with genetic algorithms in one global step.

	$N \times M$	Scheduling of all jobs with GAs		
		$C_{\max^*}$ (ut)	$C_{\max}$ (ut)	Rate (%)
Ex01	6×3	9	10	96.67
Ex02	6×5	16	16	72.94
Ex03	8×5	17	17	96.47
Ex04	8×7	7	9	73.01
Ex05	12×6	11	13	93.59
Ex06	9×9	7	7	82.53
Ex07	13×5	15	17	96.47
Ex08	13×8	10	12	97.91
Ex09	15×10	7	8	87.50
Ex10	12×7	15	17	67.22
Ex11	20×8	8	10	81.25
Ex12	19×10	8	10	87.00
Ex13	27×10	10	13	93.84
Ex14	27×10	10	13	93.84

6. Conclusion

In this study, we analyzed a method of scheduling which makes it possible to determine optimized execution times of tasks on parallel machines in the case of a flexible job shop scheduling problem with the insertion of predicted tasks. A genetic algorithm is applied to build a set of good solutions for the jobs known with certainty in the first step. In the second step, a method of insertion is applied to increase the productivity of the initial solutions by inserting jobs which may be uncertain but likely to be ordered. The method takes into account all the constraints and specifications of the problem. The quality and robustness of this heuristic method can be verified through the computation of lower bounds to the makespan. The bounds for the firm list of jobs generalize those in the literature on parallel machines. We also propose new lower bounds to the makespan of the predicted jobs. Simulations show that a small distance between the lower bounds and the values of the criteria obtained for the solutions generated by the proposed approach is generally satisfactory and promising. These results demonstrate good quality of the different lower bounds and the adequacy of the proposed method.

References

Alvarez-Valdes R. and Tamarit J.M. (1987): *Project scheduling with resource constraints: A branch & bound approach*. — Europ. J. Oper. Res., Vol. 29, No. 3, pp. 262–273.

Artigues C. (1997): *Ordonnancement en temps réel d'ateliers avec temps de préparation des ressources*. — Ph.D. thesis, University of Paul Sabatier, Toulouse, France.

Artigues C., Michelon P. and Reusser S. (2003): *Insertion techniques for static and dynamic resource constrained project scheduling*. — Europ. J. Oper. Res., Vol. 149, No. 2, pp. 249–267.

Berkoune D., Mesghouni K. and Rabenasolo B. (2004): *Insertion methods of uncertain demands in workshop scheduling*. — Proc. 4-th Conf. AUTEX, Roubaix, France, (on CD-ROM).

Billaut J.C., Carlier J. and Néron A. (2002): *Ordonnancement d'ateliers à ressources multiples. Ordonnancement de la production*. — Paris: Hermès.

Brucker P. (2003): *Scheduling Algorithms, 4-th Ed.* — New York: Springer.

Carlier J. (1982): *The one machine sequencing problem*. — Europ. J. Oper. Res., Vol. 11, No. 1, pp. 42–47.

Carlier J. (1987): *Scheduling jobs with release dates and tails on identical machines to minimize makespan*. — Europ. J. Oper. Res., Vol. 29, No. 3, pp. 298–306.

Carlier J. and Chrétienne P. (1988): *Problème d'ordonnancement modélisation/complexité/algorithmes*. — Paris: Masson.

Carlier J. and Pinson E. (1989): *An algorithm for solving the job shop problem*. — Manag. Sci., Vol. 35, No. 2, pp. 164–176.

Della Croce F., Tadei R. and Volta G. (1995): *A genetic algorithm for job shop problem*. — Comput. Oper. Res., Vol. 22, No. 1, pp. 15–24.

Demeulemeester E. and Herroleln W. (1990): *A branch and bound procedure for the multiple constrained resource project scheduling problem*. — Proc. 2-nd Int. Workshop *Project Management and Scheduling*, Compiègne, France, pp. 8–25.

Goldberg D.E. (1989): *Genetic Algorithms in Search, Optimization and Machine Learning*. — Reading, MA: Addison-Wesley.

Holland J.H. (1992): *Adaptation in Natural and Artificial Systems, 2-nd Ed.* — Michigan: University Michigan MIT Press.

Kacem I. (2003): *Ordonnancement multicritères des job shops flexibles: Formulation, bornes inférieures et approche évolutionniste coopérative*. — Ph.D. thesis, University of Lille 1, Lille, France.

Kobayashi S., Ono I. and Yamamura M. (1995): *An efficient genetic algorithm for job shop scheduling problem*. — Proc. *ICGA'95*, San Francisco, CA, USA, pp. 506–511.

Mattfeld D.C. and Bierwirth C. (2004): *An efficient genetic algorithm for job shop scheduling with tardiness objectives*. — Europ. J. Oper. Res., Vol. 155, No. 3, pp. 616–630.

Mesghouni K. (1999): *Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de la production*. — Ph.D. thesis, University of Lille 1, Lille, France.

Mesghouni K. and Rabenasolo B. (2002): *Multi-period predictive production scheduling with uncertain demands*. — Proc. IEEE Int. Conf. *Systems, Man and Cybernetics, SMC'02*, Hammamet, Tunisia, Vol. 6, Paper WA2K2, p. 6.

Mesghouni K., Hammadi S. and Borne P. (2004): *Evolutionary algorithm for job shop scheduling*. — Int. J. Appl. Math. Comput. Sci., Vol. 14, No. 1, pp. 91–103.

Pinedo M. (2002): *Scheduling: Theory, Algorithm, and Systems, 2-nd Ed*. — Upper Saddle River, NJ: Prentice Hall.

Ponnambalam S.G., Aravindan P. and Sreenivasa Rao P. (2001): *Comparative evaluation of genetic algorithms for job shop scheduling*. — Prod. Plann. Contr., Vol. 12, No. 6, pp. 560–74.

Renders J.M. (1995): *Algorithmes Génétiques et Réseaux de Neurones*. — Paris: Hermès.

Sevaux M. and Dauzère-Pérès S. (2003): *Genetic algorithms to minimize the weighted number of late jobs on a single machine*. — Europ. J. Oper. Res., Vol. 151, No. 2, pp. 296–306.

Syswerda G. (1990): *Schedule optimization using genetic algorithm*, In: Handbook of Genetic Algorithms (L. Davis, Ed.). — New York: Van Nostrand Reinhold.

Received: 17 March 2005
Revised: 2 February 2006
Re-revised: 10 April 2006