

A HYBRID ALGORITHM TO MINIMIZE THE NUMBER OF TARDY JOBS IN SINGLE MACHINE SCHEDULING

BANCILA, D.; BUZATU, C. & FOTA, A.

Abstract: Starting from the description and the classification of scheduling problems, this paper studies the problem of scheduling n jobs on a single machine in order to minimize the number of late jobs. Each job has a release time, a certain processing time and a due time. Taking into consideration all the aspects involved in scheduling, this specific case of problem is one of the most important because of its applicability on the engineering field, many solutions being proposed for its solving. By combining priority rules with genetic algorithms, this paper offers a hybrid solution for solving this problem.

Key words: scheduling, genetic, algorithm, optimization, priority



Authors' data: Ing. Dr. **Bancila**, D[aniel]; Univ. Prof. Ing. Dr. **Buzatu**, C[onstantin]; Univ. Assist. Prof. Ing. Dr. **Fota**, A[driana], Transilvania University, B-dul Eroilor 29, Brasov, Romania, d.bancila@unitbv.ro, cobuzatu@unitbv.ro, fota.a@unitbv.ro

This Publication has to be referred as: Bancila, D[aniel]; Buzatu, C[onstantin] & Fota, A[driana] (2010). A Hybrid Algorithm to Minimize the Number of Tardy Jobs in Single Machine Scheduling, Chapter 48 in DAAAM International Scientific Book 2010, pp. 549-558, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-901509-74-2, ISSN 1726-9687, Vienna, Austria
DOI: 10.2507/daaam.scibook.2010.48

1. Introduction

Scheduling is an important tool for manufacturing and engineering, where it can have a major impact on the productivity of a process. In manufacturing, the purpose of scheduling is to minimize the production time and costs, by telling a production facility what to make, when, with which staff, and on which equipment. Production scheduling aims to maximize the efficiency of the operation and reduce costs.

The approach of a scheduling problem supposes:

- input data description;
- output data description;
- optimization criteria description;
- disturbance events description.

For each operation, the following data have to be known:

- p_i – job processing time;
- r_i – earliest starting time;
- d_i – most late finishing time;
- res_i – allocated resources for the job.

As a result of the scheduling process the following output data must be obtained:

- T_i – real starting time of each operation;
- C_i – real completion time of each operation;
- L_i – the delay of each operation reported to its most late starting time.

The scheduling process takes into account some specific restrictions. In general the following restrictions are considered:

- precedence, shows the dependence of an operation to another;
- cumulative, shown that every resource is in limited quantity;
- disjunctive, shows that in the case of allocating the same resource to several operations these cannot be overlapped.

The optimization criteria define the scheduling objectives, thus the selection of scheduling scenarios. In most practical problems it is difficult to establish that one particular scheduling scenario is better or worse than other because of the large number of planning criteria. In most cases the efficiency of the scheduling problem solutions is appreciated after one of the following criteria:

- minimizing the production cycle;
- minimizing the waiting time;
- minimizing the delay time;
- minimizing the unfinished stocks;
- minimizing the costs.

In practice, beside all data and restrictions described above, certain unexpected disturbance events can appear, and thus the scheduling problems become more complicated. The main disturbance events affecting the production planning are:

- the priority in execution of certain jobs generally caused by clients demands;

- the absence of certain raw materials;
- the malfunctioning of certain machines or installations;
- repeating certain jobs because of scrap;
- variable transportation times between operations.

The description of such operations is done in the same way as describing the input data. An adequate solution can be obtained using the simulation as a method of evaluating production planning solutions.

The benefits of production scheduling include:

- process change-over reduction;
- inventory reduction, leveling;
- reduced scheduling effort;
- increased production efficiency;
- labor load leveling;
- accurate delivery date quotes;
- real time information.

Taking into account the number of jobs and the number of allocated machines the following scheduling cases can be described:

- n/1 job shop – scheduling n jobs on a single machine;
- n/m flow shop – scheduling n jobs that follow the same order on m machines;
- n/m job shop – scheduling n jobs on m machines (Bancila, 2008).

The general problem of scheduling on a single machine, denoted by $1|r_j|\sum U_j$, has, in the last years, gained a lot of interest because of its applicability on the engineering field. A set of n jobs needs to be scheduled on one machine, the processing cannot be interrupted, there are no precedence relations among jobs and only one job can be processed at a time. Each job J_j has a release time r_j , needs p_j time units to be processed on the machine and must be completed until the due time d_j . The starting time of a job is denoted by t_j ($t_j \geq r_j$) and the completion time by C_j ($C_j = t_j + p_j$). If the J_j job is completed after its due date ($C_j > d_j \Rightarrow U_j = 1$) it is said to be *late*. Otherwise, if $C_j \leq d_j$, the job is *on time*. Thus, our objective is to minimize the number of late jobs ($\sum U_j$).

In literature, this problem is considered to be NP-hard, series of solving methods being proposed over time.

In (Dauzere-Peres & Sevaux, 2004) the authors solve this problem by developing a branch-and-bound method. This algorithm defines a “master sequence” by using dominant rules.

In (Crauwels et al., 2005) the same problem is also treated by branch-and-bound techniques, but the jobs are grouped into families. A two alternative branching schemes is proposed.

The two methods presented above have a highly degree of efficiency in the case of small problems. However, in the case of more complex problems the computational time increases, making them impossible to be solved.

In (Gelinas & Soumis, 1997) is presented the scheduling problem on a single machine having as objective the minimization of completion time of all jobs. In this case the authors used the dynamic programming in order to solve the problem.

(Vakhania, 2004) solved the same problem by the means of an algorithm called “balancing procedure”, which used the GT heuristic.

In the last years, more and more interest gained the suboptimal methods, among these the genetic algorithms being frequently used.

In (Sevaux & Sorensen, 2004), is presented a genetic algorithm applied on a Just In Time scheduling, while in (Miller et al., 1999) the genetic algorithm is combined with an Wagner-Whitin algorithm and with TSP modeling.

2. Genetic Algorithms

A genetic algorithm (Fig.1) is an optimal search method motivated by natural selection and natural evolution. It maintains a population where each individual is characterized by its chromosome. Each chromosome consists of a sequence of genes.

In this paper, a gene corresponds to one of n jobs and a chromosome corresponds to a solution (a sequence of jobs). The quality of a chromosome is measured by the objective value of its sequence. A population is a collection of solutions or chromosomes. To mimic natural selection GA uses the quality of a chromosome to determine its fitness, which is then used to determine the probability that chromosome will survive to the next generation.

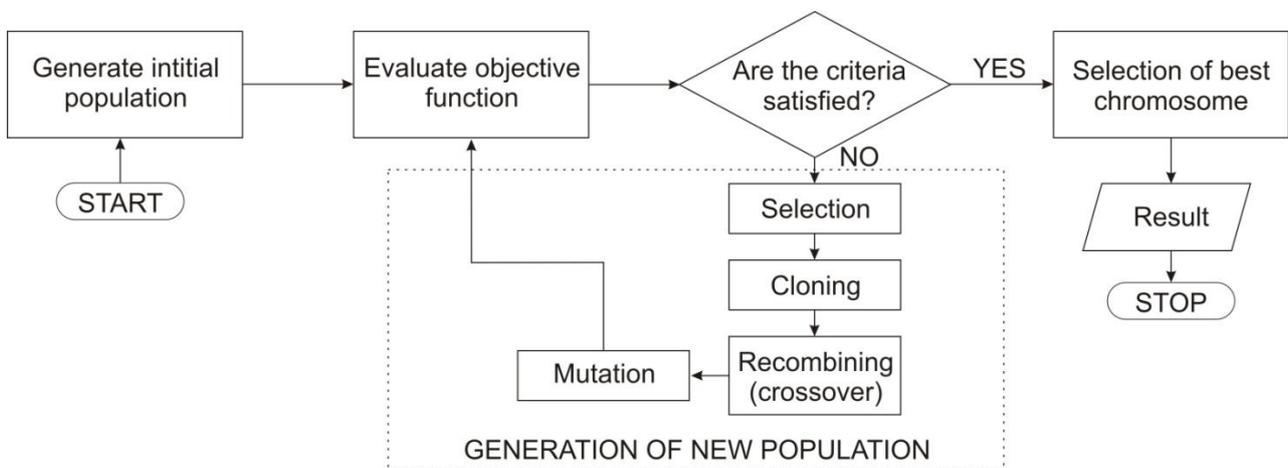


Fig. 1. The fundamental structure of a genetic algorithm

GA generates offspring from a pair of chromosomes in the population. Chromosomes with high fitness values will survive and those with low fitness values will die out. The process is designed to produce a new generation that tends to have chromosomes that are better than those in the previous generation. In the standard GA, a finite length of string represents each chromosome. A string may consist of binary bits, integers, or characters. A problem solution must be encoded in strings. The encoding scheme is important because it has a significant effect on the accuracy of the GA. In our problem, we represent a solution (or chromosome) by an integer

string stored in an array S . The length of string is equal to the total number of jobs needed to be scheduled.

Basically, the GA procedure includes chromosome reproduction, chromosome crossover, gene mutation, chromosome fitness, and natural selection. The reproduction operator will reproduce the next generation based on their fitness value. The crossover operator, the most important step in a GA, exchanges a pair of substrings of their parents to generate offspring chromosomes. The mutation operator randomly selects some of the genes of each chromosome and changes their values (Miller et al., 1999). In this problem, the GA is implemented in the following way.

- *Reproduction scheme & evaluation.*

The reproduction scheme adopted here is an elitist ranking in which the individuals are sorted by their fitness value. Then, the next generation is obtained by using the reproduce operator called roulette wheel. Here, each individual in a population has a roulette wheel slot sized in proportion to its fitness. To reproduce, we simply spin the weighted roulette wheel and obtain a reproduction candidate with probability proportional to its fitness. Each time we require another offspring, a simple spin of the weighted roulette wheel yields the reproduction candidate. In this study, the cost function consisted in minimizing the number of tardy jobs $\min(L_j)$. This function is used for the evaluation of each chromosome, and therefore assigning to each chromosome a fitness value.

- *GA operators.*

The two-point ring-like crossover operator is used because it offers a less positional bias than the one-point standard crossover without introducing any distribution bias. Other ordered crossover operators, such as PPX or GOX, have also been tested, but their performance is almost the same as the two-point ring-like operator. The mutation operator employed here is a position based mutation operator (PBM). The mutated position is randomly chosen.

- *GA parameters.*

Another important feature of the GA procedure is the operator rates. It has become widely recognized that variable operator rates are useful for maintaining diversity in the population and, hence, for alleviating the premature convergence problem. A crossover rate of 50% and a mutation rate of 10% are used in this case.

The most attractive feature of the GA optimization lies in its attempt to explore the search space in a global fashion by studying the entire population of the chromosomes. However, the implementation of a standard GA often encounters the problem of either premature convergence to the local optima, or of an inefficient search requiring a long time to reach an optimal solution. To overcome these difficulties, we propose a hybrid algorithm. This algorithm uses the modifications to the standard GA as described above. Other aspects of the procedure are discussed in the following section.

3. The Hybrid Algorithm

The proposed hybrid algorithm (HA) (Fig.2) is based on the fundamentals of genetic algorithms, being modified according to the following steps:

1. Generate an initial population.
2. Produce a new generation P.
3. Invert the new obtained P generation to produce InvP generation.
4. Evaluate the objective function for both P and InvP generations.
5. Choose from P and InvP the best n individuals to form the next generation (n – number of the individuals from a generation).
6. Repeat Steps 2–5 until the prescribed stop criterion is satisfied.

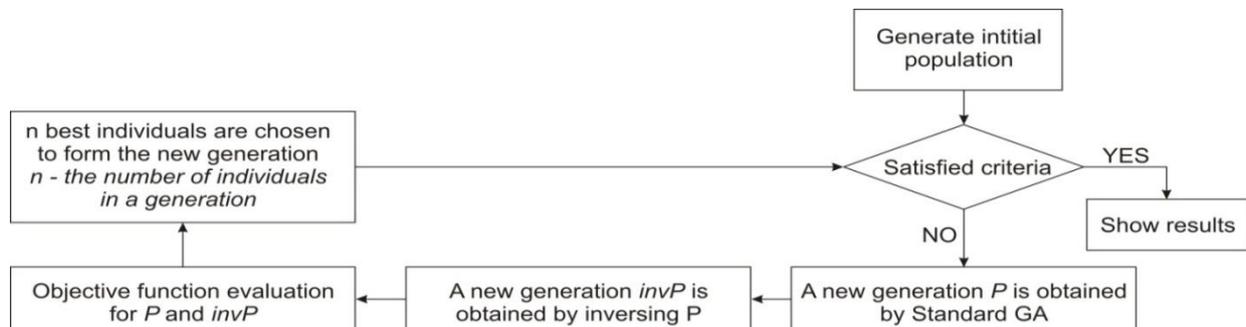


Fig. 2. The hybrid algorithm description

Each step will be explained in the following sections.

- *Generate the initial population.*
 In this step, initial chromosomes (or individuals), that are the initial solutions, are generated under some controls using different rules of heuristics. For example one initial solution is generated based on a SRPT rule (jobs with shortest remaining processing time are scheduled first). Then, the HA randomly generates other initial solutions until the population size is full. Other heuristic rules used to generate the initial solution can be: LRPT (longest remaining processing time), MS (minimal slack), SPT (shortest processing time), LPT (longest processing time), or even complex heuristic rules which combine different simple rules. The number of individuals in the population is given as a parameter of the GA, depending on the complexity of the problem.
- *Produce a new generation P.*
 Here, the HA uses the standard GA procedure to generate offspring and conduct the global search. Two constraints are taken into consideration here: sequence dependent feature and the number of tardy jobs. There are three GA operators, reproduction, crossover, mutation, used in the HA. They are implemented in the same way as the standard GA described in the last section.
- *Invert the new obtained generation.*
 In this step, the generation produced by the standard GA is inverted. The inverted generation (InvP) is produced by mirroring each chromosome's genes (Fig.3).

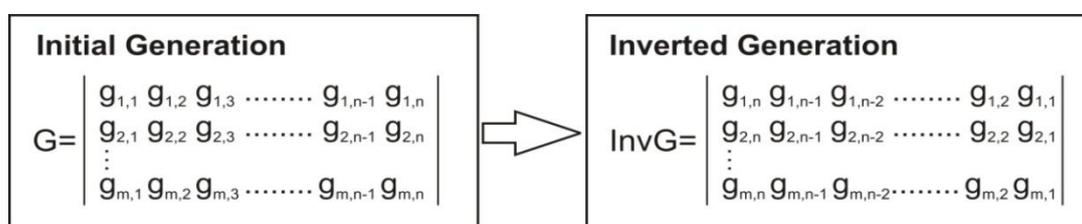


Fig. 3. Inverting procedure

- *Evaluate objective function.*
In this step, we evaluate both previously obtained generations by the objective function. Each individual on both generations will be then assigned his objective function value.
- *Choose individuals to form the new generation.*
The next generation is obtained by selecting from the two P and InvP generations, the individuals which better satisfy the objective function. The selection continues until the number n of population size is reached.
- *Repeat Steps 2–5 until the prescribed stop criterion is satisfied.*
Now, the fitness value of each chromosome is obtained in this step. In this paper, the objective function is used for the fitness of each chromosome. Then the HA checks the stop criterion. If it is satisfied, HA outputs the best solution in the final population and exits. Otherwise, HA repeats steps 2–5.

4. H.A. vs. S.G.A.

The advantages of using the hybrid algorithm instead of a standard genetic algorithm relies themselves from the construction of the algorithm:

- By generating a controlled initial population, we eliminate from the start a certain number of unfeasible solution, therefore minimizing the search space;
- By generating the inverted InvP, evaluating both new, P and InvP formed generations by the objective function, and choosing the next generation as a resultant of the best individuals in each generation, we assure a higher degree of avoidance the premature convergence to a local optima, a smaller computational time than in a standard genetic algorithm, and a reduced number of necessary generations for obtaining the optimum solution (Buzatu & Bancila, 2008).

5. Computational Results and Comparison

In this section we will discuss the computational results of the hybrid algorithm in the case of single machine scheduling. In order to properly compare the performances of a standard GA and the HA, we applied both algorithms on several scheduling problems, a simple case being outlined in the following section.

The problem consists of seven jobs which have to be scheduled on a single machine. The release, processing and due time for each job are given in Tab. 1.

J	1	2	3	4	5	6	7
r_j	0	2	4	5	7	11	18
p_j	3	5	2	1	7	6	3
d_j	4	7	9	8	18	22	27

Tab. 1. Release, processing and due time for each job

When comparing their performance, the HA and GA have the same population size, consisting of six individuals, the genetic operators are the same in both methods, and the number of generations is set to 60. The results of one sample are illustrated in Fig. 4 and Fig.5.

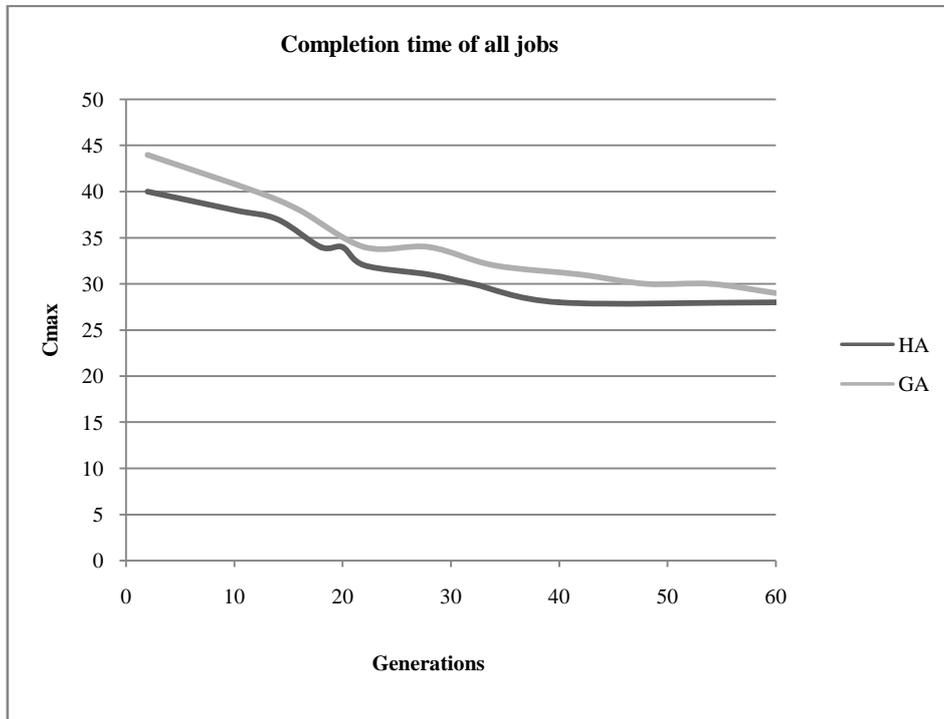


Fig. 4. Comparison between HA and GA – completion time of all jobs

From Fig. 4 it is shown that the HA improves much faster than the GA, which is an advantageous feature of the HA. In addition, the execution time for the HA is almost the same as the GA. The time spent on inverting procedure is negligible.

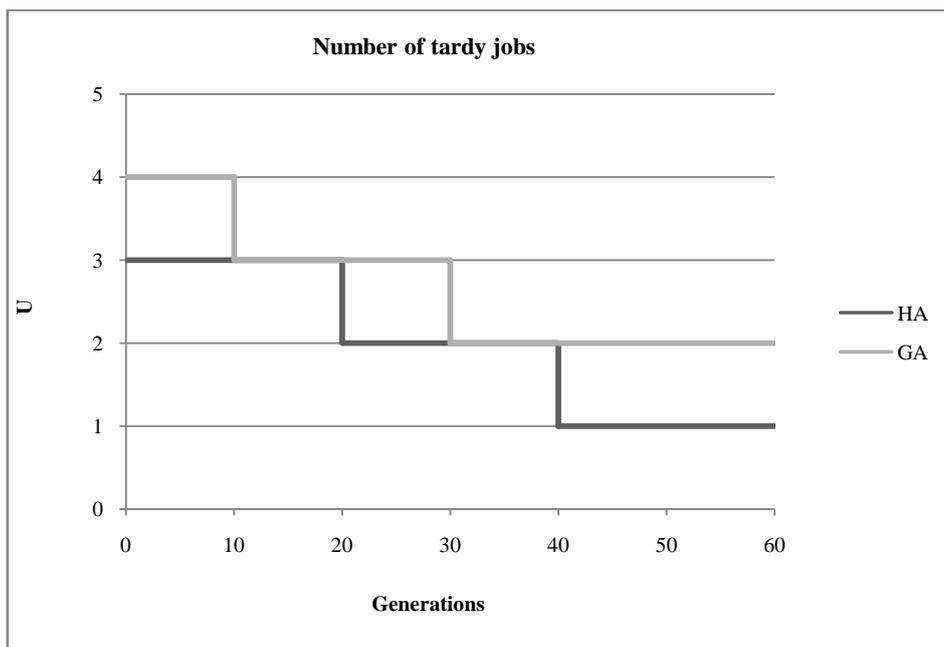


Fig. 5. Comparison between HA and GA – number of tardy jobs

As seen in Fig. 5 the number of tardy jobs obtained by applying HA to our problem is smaller than in the case of GA.

The two optimal solutions generated by applying HA and GA on our problem become:

1→3→4→5→6→7→2 (generated with HA);

1→3→5→6→7→2→4 (generated with GA).

By applying active-decoding procedure on these two solutions we obtained the two Gantt diagrams in Fig. 6 and Fig. 7.

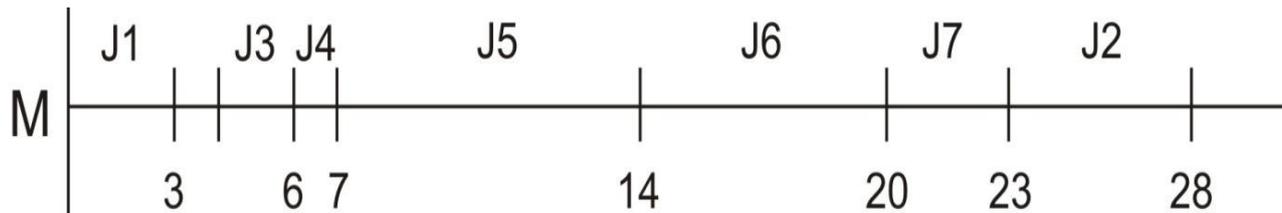


Fig. 6. Gantt diagram obtained by active-decoding the solution generated with HA

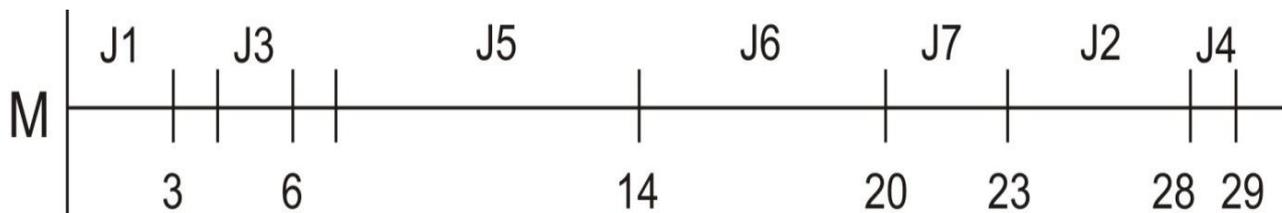


Fig. 7. Gantt diagram obtained by active-decoding the solution generated with GA

As seen in Fig. 6 and Fig.7 by solving the problem with both methods, our original HA, and the standard genetic algorithm, the completion time in the case of HA is of 28 time units, as opposed to the 29 time units obtained in the case of standard GA.

6. Conclusions

A hybrid algorithm based on an integration of a genetic algorithm and heuristic rules is proposed in this paper in order to solve the problem of minimizing the number of tardy jobs in single machine scheduling.

The most prominent feature of our algorithm is a combination of heuristic rules (used to generate the initial individuals) with genetic operators to generate new individuals, which can effectively guide and speed up the genetic search.

Another characteristic of the algorithm is consisted in the way the next generation is formed, by using the inverted generation, greatly increasing the flexibility of the algorithm.

We have compared the hybrid algorithm HA to a standard genetic algorithm GA and the computational results showed that our algorithm is superior to the GA.

As a future research, by combining the presented algorithm with other scheduling methods like: linear programming, simple and complex heuristic rules,

branch & bound techniques, other space search methods or even artificial intelligence, new algorithms for solving different problems like scheduling on parallel machines or the generalized job shop scheduling, can be obtained.

Furthermore, as an application of the theoretical research, a scheduling software solution based on this hybrid algorithm scheduling method, is in development.

The paper is developed with the aid of the research made in the Scientifically Research Contract type CNCSIS – IDEI, code PCE_756/2008.

7. References

- Bancila, D. (2008). Scheduling Algorithms and Recommendations on Their Field of Application. *Academic Journal Of Manufacturing Engineering*, vol. 6, no. Supplement Issue 1, (2008) pp. 18-23, ISSN 1583-7904
- Buzatu, C.; Bancila, D. (2008). A Hybrid Algorithm for Job Shop Scheduling, *Proceedings of the 6th International Conference of DAAAM Baltic Industrial Engineering*, Kyttner, R., pp. 309-314, ISBN 978-9985-59-783-5, Tallinn, April 2008, Tallinn University of Technology, Tallinn
- Crauwels, H.A.J.; Potts, C.N.; Van Oudheusden, D.; Van Wassenhove, L.N. (2005). Branch and Bound Algorithms for Single Machine Scheduling with Batching to Minimize the Number of Late Jobs. *Journal of Scheduling*, Vol. 8, (2005) pp. 161-177, ISSN 1094-6136
- Dauzere-Peres, S.; Sevaux, M. (2004). An Exact Method to Minimize the Number of Tardy Jobs in Single Machine Scheduling. *Journal of Scheduling*, Vol. 7, (2004) pp. 405-420, ISSN 1094-6136
- Gelinas, S.; Soumis, F. (1997). A Dynamic Programming Algorithm for Single Machine Scheduling with Ready Times. *Annals of Operations Research*, Vol. 69, (1997) pp. 135-156, ISSN 0254-5330
- Miller, D.; Chen, H.; Matson, J.; Liu, Q. (1999). A Hybrid Genetic Algorithm for the Single Machine Scheduling Problem. *Journal of Heuristics*, Vol. 5, (1999) pp. 437-454, ISSN 1381-1231
- Sevaux, M.; Sorensen, K. (2004). A Genetic Algorithm for Robust Schedules in a One-Machine Environment with Ready Times and Due Dates. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, Vol. 2, (2004) pp. 129-147, ISSN 1619-4500
- Vakhania, N. (2004). Single-Machine Scheduling with Release Times and Tails. *Annals of Operations Research*, Vol. 129, (2004) pp. 253-271, ISSN 0254-5330