

Solving multi-objective
production scheduling problems
with Tabu Search

by

Taïcir Loukil*, Jacques Teghem**
and Philippe Fortemps**

*Laboratoire de Recherche Opérationnelle
Faculté des Sciences Economiques et de Gestion de Sfax
Route de l'Aérodrome km 4, B.P. 1088, 3018 Sfax, Tunisie
E-mail: Taicir.Loukil@fsegs.rnu.tn

**Service de Mathématiques et de Recherche Opérationnelle
Faculté Polytechnique de Mons
9, rue de Houdain, 7000 Mons, Belgique
E-mail: Jacques.Teghem@fpms.ac.be
E-mail: Philippe.Fortemps@fpms.ac.be

Abstract: Most of multiple criteria scheduling problems are NP-hard, so that exact procedures can only solve small problems and with two criteria. The complexity and the diversity of multiple criteria scheduling problems resulted in many alternative approaches to solve them. Exact and approximate procedures proposed in the literature are mainly dedicated to the problem to be solved and their performance depends on the problem data, on the criteria optimized, and are generally difficult to implement. We propose in this paper a Tabu Search approach to multiple criteria scheduling problems. The proposed procedure is a general flexible method, able to solve hard multiple criteria scheduling problems, easy to implement, and providing a set of potential efficient schedules. The criteria are any combination chosen from $(C_{\max}, T_{\max}, \bar{L}, N_T$ and $\bar{F})$.

Keywords: scheduling, multiple criteria, Tabu Search.

1. Introduction

Most of research in production scheduling is concerned with minimization of a single criterion. However, scheduling problems often involve more than one as-

scant attention has been given to multiple criteria scheduling problems, especially in the case of multiple machines. This is due to the extreme complexity of these combinatorial problems. A comprehensive survey of multi-objective scheduling problems is in preparation (Loukil and Teghem, 1998).

Hoogeveen (1992) and Chen and Bulfin (1993) studied the complexity of the single machine bicriteria and multiple criteria problems. They concluded that only problems including flow time as primary criterion, in a hierarchical optimization, and problems minimizing flow time and maximum tardiness can be solved in polynomial time. All other problems are either shown to be NP hard or remain open as far as computational complexity is concerned. Obviously, the problems including more than one machine and two criteria are more difficult.

Metaheuristics, like Simulated Annealing, Tabu Search and Genetic Algorithms have demonstrated their ability to solve combinatorial problems such as vehicle routing, production scheduling, time tabling, etc. Recently, some authors suggested to adapt metaheuristics in order to solve multi-objective combinatorial (MOCO) problems. Fortemps et al. (1995) and Ulungu et al. (1999) present the MOSA algorithm for solving combinatorial optimization problems, and an interactive version was recently designed (Teghem et al., 2000). Of other authors: Ben Abdelaziz, Chaouachi and Krichen (1997) proposed a MOTS algorithm for the multi-objective knapsack problem; Hansen (1997) — a general adaptation of Tabu Search to the multiple criteria problems; Jaskiewicz (1997) — a metaheuristic approach to the multi-objective nurse scheduling; Gandibleux et al. (1998) — a MOTS procedure based on the preference modeling. Loukil et al. (1999) adapted the MOSA method to solve multi-objective scheduling problems.

Since scheduling problems are also combinatorial problems, applying the metaheuristics to production scheduling with multiple criteria is suitable (see, for instance, Neppali et al., 1996).

In this paper we propose a general heuristic method, based on multi-objective Tabu Search (MOTAS), able to tackle general multiobjective combinatorial problems.

Another aim of this paper is to show how such a method can be used to solve complex multiple criteria scheduling problems.

The suggested procedure is composed of two embedded heuristics:

- the first — called the scheduler — is a conventional one meant to build a feasible schedule;
- the second — called the optimizer — is the MOTAS method.

The aim of this general procedure is to generate a list of potentially efficient schedules.

The paper is organized as follows: Section 2 gives a description of the MOTAS heuristic for a general MOCO problem. The multiple criteria production scheduling is considered in Section 3 and the procedure “scheduler-optimizer”

2. A multiobjective Tabu Search algorithm: MOTAS

2.1. The problem

We focus on the analysis of a MOCO problem (P) with the general form:

$$\text{“min”}_{x \in S} z_k(x), \quad k = 1, \dots, K, \quad (\text{P})$$

where S is a finite set of feasible solutions.

A solution $x^* \in S$ is efficient for (P) if there does not exist any other solution $x \in S$ such that: $z_k(x) \leq z_k(x^*)$, $k = 1, \dots, K$, with at least one strict inequality. Then, the vector $z(x^*)$ is said to be non-dominated in the space of objective functions.

Let $E(P)$ denote the set of all efficient solutions to a problem P . The aim of the MOTAS method is to determine a good approximation, denoted by PE , called the set of potentially efficient solutions, i.e. the generated solutions which are not dominated by any other generated solution. As in the MOSA method (see Ulungu et al., 1999), the method will require consideration of some weight vector $\lambda = (\lambda_k, k = 1, \dots, K; \lambda_k > 0 \forall k \text{ and } \sum_{k=1}^K \lambda_k = 1)$ in order to aggregate, in a way defined below, the different objective functions.

2.2. Basic concepts

Let x_n be a current solution at iteration n .

$N(x_n)$ is a neighborhood of x_n .

A subneighborhood $SN(x_n)$ is made by randomly selecting K_1 neighbors.

Tabu list length is K_2 with $K_2 < K_1$.

Let y_i and y_j be two solutions in $SN(x_n)$. $\Delta_k(y_i) = z_k(y_i) - z_k(x_n)$ (resp. $\Delta_k(y_j) = z_k(y_j) - z_k(x_n)$) is the variation of the objective function k . While moving from x_n to y_i (resp. y_j), three possibilities can be considered.

1. $\Delta_k(y_i) \leq \Delta_k(y_j) \forall k \in \{1, 2, \dots, K\}$ with at least one strict inequality: solution y_i dominates solution y_j ;
2. $\Delta_k(y_i) \geq \Delta_k(y_j) \forall k \in \{1, 2, \dots, K\}$ with at least one strict inequality: solution y_i is dominated by solution y_j ;
3. $k, k' \in \{1, 2, \dots, K\}$ such that: $\Delta_k(y_i) < \Delta_k(y_j)$, $\Delta_{k'}(y_i) > \Delta_{k'}(y_j)$; in this case, neither y_i dominates y_j , nor y_j dominates y_i ; this situation is specific to the multiple criteria framework.

Among the non-dominated solutions in $SN(x_n)$, it is necessary to define a method for selecting the “best” neighbor for x_n . Thus, solution y_i is “better” than solution y_j if its modification vector $\Delta(y_i)$ is smaller than the modification vector $\Delta(y_j)$ based on the infinite norm:

$$\text{Max}_{\lambda} \lambda, \frac{\Delta_k(y_i)}{\lambda} < \text{Max}_{\lambda} \lambda, \frac{\Delta_k(y_j)}{\lambda}$$

where R_k is the range of the k^{th} objective function for all non-dominated neighbors of x_n :

$$R_k = m_k - M_k$$

$$R_k = \max(\max_i(z_k(y_i)), z_k(x_n)) - \min(\min_i(z_k(y_i)), z_k(x_n)).$$

An aspiration value is defined by the equation:

$$A(y) = \max_k \lambda_k \frac{z_k(y) - z_k(x_n)}{m_k - M_k}$$

$$A^* = \min_{x \in PE(\lambda)} A(x)$$

2.3. Principles of the MOTAS algorithm

2.3.1. Determination of $PE(\lambda)$

The following procedure is applied to generate a set of potentially efficient solutions $PE(\lambda)$.

Initialization: Draw at random an initial solution x_0 , evaluate $z_k(x_0) \forall k$;

$PE(\lambda) = \{x_0\}$;

$M_k = 0$; $m_k = z_k(x_0) \forall k$;

Parameters $K_1 > K_2$; Parameter N (maximum number of iterations); $T = \emptyset$ (tabu list); $n = 0$.

Iterative Procedure:

x_n — current solution;

$\delta = 0$.

Generate randomly K_1 neighbors: y_i ($i = 1, \dots, K_1$) is a neighbor of x_n .

For each $i < K_1$:

- If y_i is dominated by any $x \in PE(\lambda)$ do $i = i + 1$
- If y_i is non-dominated by all $x \in PE(\lambda)$, then update $PE(\lambda)$ by including y_i
 - If y_i is non tabu
 - If $\delta = 0$: $x_{n+1} \leftarrow y_i$ and $\delta = 1$;
 - If $\delta = 1$ and $A(y_i) < A(x_{n+1})$: $x_{n+1} \leftarrow y_i$.
 - If y_i is tabu
 - If $\delta = 0$ and $A(y_i) < A^*$: $x_{n+1} \leftarrow y_i$;
 - If $\delta = 1$ and $A(y_i) < \min(A^*, A(x_{n+1}))$: $x_{n+1} \leftarrow y_i$.

Do $i = i + 1$;

If $i = K_1$: update M_k , m_k , A^* and tabu list

while $n < N$ do $n = n + 1$;

End if $n = N$;

REMARK We use here sampling as neighborhood operator to generate the solutions y_i , $i = 1, \dots, K_1$. Clearly, other neighborhood operators can also be implemented in MOTAS.

2.3.2. Generation of PE

The weights have a significant influence on the generation of $PE(\lambda)$. As a matter of fact, because of the use of a scalarizing function, a given weight set λ induces a privileged search direction on the efficient frontier: the procedure generates only a subset of potentially efficient solutions in that direction.

Unfortunately, a lot of solutions of $PE(\lambda)$ appear to be dominated by some other solutions obtained with another weight set $\bar{\lambda}$ and belonging to $PE(\bar{\lambda})$ and vice versa.

Though the weight set induces a privileged direction, it is possible to obtain solutions not in this direction. These solutions are often dominated by some solutions generated with other weight sets.

The procedure which allows us to obtain a good approximation PE to $E(P)$ is as follows:

- A widely diversified set of weights is considered: different weight vectors $\lambda^{(l)}$, $l \in L$, are generated where $\lambda^{(l)} = (\lambda_k^{(l)}, k = 1, \dots, K)$ with $\lambda^{(l)} \geq 0 \forall k$ and $\sum_{k=1}^K \lambda_k^{(l)} = 1, \forall l \in L$. This set of weights is uniformly generated (see Ulungu et al., 1999). For each of them, the procedure described in Subsection 2.3.1 is applied to obtain $|L|$ lists $PE(\lambda^{(l)})$.
- The set $\bigcup_{l=1}^{|L|} PE(\lambda^{(l)})$ is filtered by pairwise comparisons in order to remove the dominated solutions. This filtering procedure is denoted by \wedge .

Finally,

$$\dot{P}E = \bigwedge_{l=1}^{|L|} PE(\lambda^{(l)}).$$

REMARK A data structure like quad tree (Habenicht, 1982) can be used to efficiently manage the filtering procedure.

3. Multicriteria production scheduling problems

It can be interesting to study the efficiency of the MOTAS method on different MOCO problems. For instance, we are currently working on the comparison between the MOSA method of Ulungu et al. (1999) and the MOTAS method on multiple objective assignment problems and multi objective knapsack problems, analysing the quality of the approximations PE obtained with these respective methods.

The aim of the present paper is different: we want to prove that the MOTAS method can be useful in the analysis of complex multi objective production

3.1. The scheduling problems

The problems considered can be classical flow job of job shop (see, for instance, French, 1982) or even more complex flexible job shop. An example of such a problem is given in El Maqrini and Teghem (1996). It consists of the schedule of n jobs with several operations on a network of machines in series and in parallel. Each machine can perform a set of possible operations and each job is characterized by a set of ordered operations; the processing times of these operations are known and can depend on the machine used. The configuration of the workshop gives several possible routings for each job.

Many additional constraints are considered in El Maqrini and Teghem (1996):

- non null ready dates of jobs,
- variable delay between two successive operations of a same job,
- dependent set-up times between two operations on a machine,
- waiting stocks of finite capacity in front of each machine,
- maintenance periods of the machines.

Our aim is to consider such models with several regular criteria such as the makespan (C_{\max}), the mean flow time (\bar{F}), the maximum and mean tardiness (T_{\max}, \bar{T}), the number of tardy jobs (N_T) (for a definition of these criteria, see French, 1982), and to generate an approximation PE of the set of efficient schedules.

3.2. Particularization of MOTAS method

In this specific framework, several elements of the MOTAS method described in Section 2 must be particularized:

- a solution x is described as an order of the n jobs,
- a neighbor solution is obtained by the permutation of two successive jobs,
- the cardinality K_1 of the subneighborhood is equal to $\lfloor n/k_1 \rfloor$ with $k_1 < n$,
- the length K_2 of the tabu list is defined by $\lfloor n/k_2 \rfloor$ with $k_1 < k_2 < n$.

3.3. The scheduler

To obtain the evaluations $z_k(x)$, $k = 1, \dots, K$, of a solution on the different criteria, it is necessary to associate to the order of the n jobs corresponding to x , a real feasible schedule of the different operations on the machine configuration, in respect of the different constraints. This is the role of the scheduler. So, the scheduler is a heuristic dedicated to build a feasible solution from an ordered list of jobs. Each job is considered successively in this order and, by applying logical rules, its operations are assigned to the possible machines respecting the different constraints.

An example of such scheduler is described in details in El Maqrini and Teghem (1996): the scheduler provides a feasible schedule of all the jobs, satis-

corresponding to an ordered list of jobs i.e. to a solution x — is evaluated and the objective function values $z_k(x)$, $k = 1, \dots, K$, are determined.

3.4. The couple Optimizer-Scheduler

This scheduler is coupled with the MOTAS method, playing the role of optimizer, as described in Fig. 1.

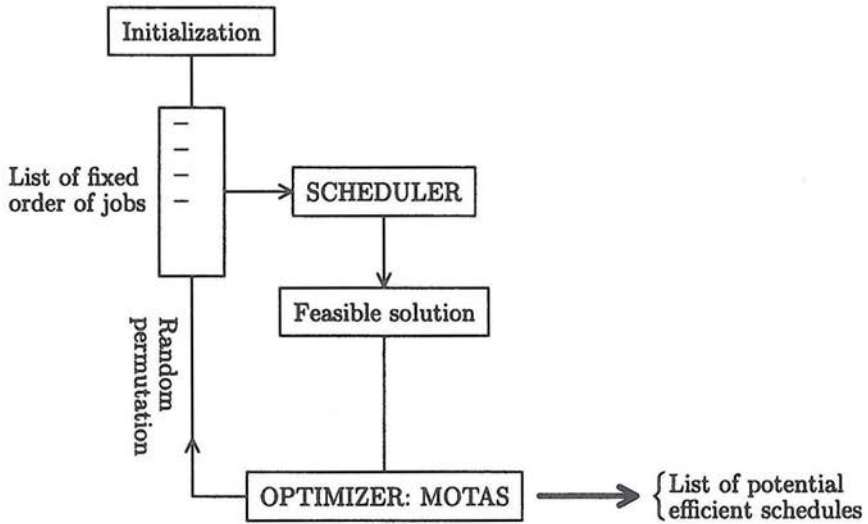


Figure 1. The proposed model

At each iteration of MOTAS, two jobs are chosen randomly and their positions in the ordered list are exchanged; the new ordered list of jobs is proposed to the scheduler. The general rules of MOTAS are applied to generate a set of potentially efficient schedules.

4. Computational results

The proposed method has been tested on many problems taken from the literature dealing with multi-objective scheduling problems (problems on a single machine, problems of open shop, flow shop and job shop). Almost all the efficient sequences already obtained in the literature are generated by the MOTAS software. Sometimes, better solutions than those found by other proposed heuristics are generated. The number of efficient sequences ranges from one to eight for the tested problems for a number of jobs limited by 15 (Table 1).

We present here the results for two of these problems ($n = 10$, $m = 5$)

Criteria	$n = 10, m = 5$		$n = 20, m = 5$	
	CPU (sec) ²	PE	CPU (sec)	PE
C_{\max}, \bar{F}	59	5	115	3
C_{\max}, \bar{T}	44	4	230	6
C_{\max}, T_{\max}	50	2	150	5
C_{\max}, N_T	53	2	177	2
\bar{F}, \bar{T}	45	2	177	2
\bar{F}, T_{\max}	55	3	265	2
\bar{F}, N_T	39	1	170	2
\bar{T}, T_{\max}	48	1	110	1
\bar{T}, N_T	49	3	141	2
T_{\max}, N_T	44	4	180	1

¹ |PE| is the number of potentially efficient solutions

² On a Pentium 166 Mhz

Table 1. Test of the software (CPU and |PE|¹) for different criteria. Number of weights' sets = 5. Max number of iterations = 50. Neighborhood size = 5. Tabu list length = 2.

Criteria	$n = 10, m = 5$			$n = 20, m = 5$		
	L = 5	γ	L = 10	L = 5	γ	L = 10
C_{\max}, \bar{F}	5/4	0	5/2	3/0	0	5/5
C_{\max}, \bar{T}	4/0	0	8/4	3/1	0	6/5
C_{\max}, T_{\max}	2/1	1	6/6	5/1	0	4/4
C_{\max}, N_T	2/2	0	3/2	2/1	1	2/2
\bar{F}, \bar{T}	2/0	0	7/7	2/2	0	3/3
\bar{F}, T_{\max}	3/1	0	3/3	2/1	0	3/3
\bar{F}, N_T	1/1	0	2/1	2/0	0	2/2
\bar{T}, T_{\max}	1/0	0	2/2	1/1	1	1/1
\bar{T}, N_T	3/3	0	2/0	2/0	0	2/2
T_{\max}, N_T	4/2	0	3/2	1/1	1	1/1
Total	27/14	1	41/29	23/8	3	29/28

$\alpha_1 = |\hat{E}_1(P)|$ is the number of potentially efficient solutions for $|L| = 5$;

$\alpha_2 = |\hat{E}_2(P)|$ is the number of potentially efficient solutions for $|L| = 10$;

$\gamma =$ is the number of solutions included in $\hat{E}_1(P) \cap \hat{E}_2(P)$;

$\beta_1 =$ is the number of solutions of $\hat{E}_1(P)$ included in $\hat{E}_1(P) \wedge \hat{E}_2(P)$;

$\beta_2 =$ is the number of solutions of $\hat{E}_2(P)$ included in $\hat{E}_1(P) \wedge \hat{E}_2(P)$.

Table 2. |PE|/N = 10 versus N = 20/|L| = 5 versus |L| = 10. Max number of

$[P(1 - T - R/2), P(1 - T + R/2)]$, where T is the tardiness factor, R is the control factor of the range of due dates (we used $R = 0.4$ and $T = 0.6$), and $P = (n + m - 1)p'$ with p' being the mean execution time of a job operation.

Logically, the CPU time increases with the number of jobs. It also increases with the maximum number of iterations, the number of weight sets and the tabu list length.

The number of potentially efficient solutions does not always increase with the number of jobs. It increases with the maximum number of iterations, the neighborhood size, the number of weights' sets, and tabu list length.

We made many other experiments for different values of parameters: number of the weight sets, the maximum number of iterations, the neighborhood size and the tabu list length. We just present in Table 2 the results for two values of $|L|$, 5 and 10.

The results confirm that the quality of the obtained solutions depends largely on the number of weight sets. The user has therefore to find a compromise between the quality of solutions and the CPU times.

5. Conclusion

In this paper we described the principles of the MOTAS method and its implementation for the multi-objective scheduling problems. The proposed method can deal with all the general problems and gives a set of potentially efficient solutions. More experiments are still necessary to obtain strong conclusions on the performance of the MOTAS method. Nevertheless, it appears already that MOTAS constitutes a useful approach for approximating the efficient set of scheduling problems in reasonable CPU times.

This method is general, flexible, easy to implement and it allows the decision maker to modify easily the parameters of the method. Furthermore, for real applications, an interactive version of the algorithm — similar as in Teghem et al. (2000) — can be used so as to take into account the decision maker's preferences.

References

- BEN ABDELAZIZ, F., CHAOUACHI, J. and KRICHEN, S. (1997) Métaheuristiques de résolution des problèmes du sac à dos multiobjectif. MOSIM 97, actes de la 1^{ère} conférence francophone "Modélisation des systèmes de production et de logistiques".
- CHEN, C.L. and BULFIN, R.L. (1993) Complexity of a single machine, multi-criteria scheduling problems. *EJOR*, **70**, 115–125.
- EL MAQRINI, H. and TEGHEM, J. (1996) Efficiency of metaheuristics to schedule general flexible job shop. Technical report. F.P.Ms. IMAGE. O3, presented at the Ninth International Working Seminar on Production Eco-

- FORTEMPS, PH., ULUNGU, E.L. and TEGHEM, J. (1995) Heuristics for Multi-Objective combinatorial optimization problem by simulated annealing. *Proceedings of the 6th National Conference on Multiple Criteria Decision Making*, Beijing.
- FRENCH, S. (1982) Sequencing and Scheduling: an introduction to the Mathematics of Job Shop.
- GANDIBLEUX, X. and FREVILLE, A. (1998) The multi-objective tabu search customized on 0-1 multiobjective knapsack problem: the two objectives case. *Deuxièmes rencontres francophones de recherche opérationnelle*, Sousse, Avril 1998.
- HABENICHT, W. (1982) Quad trees, a data structure for discrete vector optimization problems. *Lecture Notes in Economics and Mathematical Systems*, **209**, 136-145.
- HANSEN, M.P. (1997) Tabu search for multi-objective optimization. Working paper. Institute for Mathematical Modeling, Technical University of Denmark. Presented at M.C.D.M.'97, Cape Town, January 1997.
- HOOGVEEN, J.A. (1992) *Single machine bicriteria scheduling*. Doctoral Thesis, C.W.I. Amsterdam.
- JASZKIEWICZ, A. (1997) A metaheuristic approach to multiple objective nurse scheduling. *Foundations of Computing and Decision Sciences*, **22**, 3, 169-183.
- LOUKIL, T. (2000) *Ordonnancement de production multicritère*. Ph.D. thesis. Faculté des Sciences Economiques, Université de Sfax, Tunisie.
- LOUKIL, T. et TEGHEM, J. (1998) Ordonnancement multicritère de la production : Panorama de la littérature. Presented to the Deuxièmes Rencontres Francophones de Recherche Opérationnelle, Sousse, Avril 1998. Paper in preparation.
- LOUKIL, T., TEGHEM, J. and TUYTTENS, D. (1998) Solving multi-objective scheduling problems with the MOSA method. Presented at the 5th International Conference of "Decision Science Institute", Integrating Technology & Human Decisions: Global Bridges Into The 21th Century, Athens 99.
- LOUKIL, T., TEGHEM, J. and TUYTTENS, D. (1999) Solving scheduling problems with the MOSA method. *Proceedings of ACIDCA'2000. International Conference on Artificial and Computational Intelligence for Decision, Control, and Automation for Engineering and Industrial Applications*, 22-24 March 2000, Monastir, Tunisia, pp. 88-93.
- NEPPALLI, V.R., CHEN, CH.L. and GUPTA, J.N.D. (1996) Genetic algorithms for the two-stage bicriteria flow shop problem. *EJOR*, **95**, 356-373.
- TEGHEM, J., TUYTTENS, D. and ULUNGU, E.L. (2000) An interactive heuristic method for multi-objective combinatorial optimization. *Computers and Operations Research*, **27**, 621-634.
- ULUNGU, E.L., TEGHEM, J., FORTEMPS, PH. and TUYTTENS, D. (1999) MOSA method: a tool for solving multi-objective combinatorial optimization