Solving Discrete-Continuous Scheduling Problems by Tabu Search

*Joanna Józefowska** *Rafal Różycki** *Grzegorz Waligóra** *Jan Węglarz**

* Institute of Computing Science, Poznań University of Technology
Piotrowo 3A, 60-965 Poznań, Poland
Email: {Joanna.Jozefowska, Rafal.Rozycki, Grzegorz.Waligora, Jan.Weglarz}@cs.put.poznan.pl

1 Introduction

In this paper we consider discrete-continuous scheduling problems defined in [7], where general results and methodology have been presented as well. These problems are characterized by the fact that each job simultaneously requires for its processing at a time discrete and continuous (i.e. continuously-divisible) resources. We deal with a class of these problems where there are: one discrete resource, which is a set of parallel, identical machines, and one continuous, renewable resource whose total amount available at a time is limited. Jobs are independent and nonpreemptable and their ready times are equal to 0. The processing rate of each job depends on the amount of the continuous resource assigned to this job at a time. The makespan, the mean flow time and the maximum lateness are considered as the scheduling criteria. The tabu search (TS) metaheuristic to solve the considered problems is discussed. A performance analysis of tabu search for discrete-continuous scheduling problems is presented. Results obtained by tabu search are compared to optimal solutions as well as to results produced by other two metaheuristics: simulated annealing (SA) and genetic algorithms (GA). Moreover, three tabu list management methods: the Tabu Navigation Method (TNM), the Cancellation Sequence Method (CSM) and the Reverse Elimination Methods (REM) are compared. Several computational experiments are described and some conclusions and final remarks are included.

2 Discrete-continuous scheduling problems

2.1 General model

We consider n independent, nonpreemptable jobs, each of them simultaneously requiring for its processing at time t a machine from a set of m parallel, identical machines (the discrete resource) and an amount (unknown in advance) $u_i(t) \in [0,1], i = 1, 2, \ldots, n$, of a continuous, renewable resource. The job model is given in the form [1]:

$$\dot{x}_i(t) = \frac{dx_i(t)}{dt} = f_i[u_i(t)], x_i(0) = 0, x_i(C_i) = \tilde{x}_i \tag{1}$$

where $x_i(t)$ is the state of job i at time t, f_i is an increasing, continuous function, $f_i(0) = 0$, C_i is (unknown in advance) completion time of job i, and $\tilde{x}_i$ is its processing demand (final state). We assume, without loss of generality, that $\sum_{i=1}^{n} u_i(t) = 1$ for every t. The problem is to find a sequence of jobs on machines and, simultaneously, a continuous resource allocation that minimize the given scheduling criterion.

2.2 Metaheuristic approach

The defined problem can be decomposed into two interrelated subproblems: (i) to find a feasible sequence of jobs on machines, and (ii) to allocate the continuous resource among jobs already sequenced. The notion of a *feasible sequence* is of crucial importance. Let us observe that a feasible schedule can be divided into $p \leq n$ intervals defined by completion times of consecutive jobs. Let Z_k denote the combination of jobs processed in parallel in the k-th interval. Thus, in general, a feasible sequence S of combinations $Z_k, k = 1, 2, \ldots, p$, can be associated with each feasible schedule. Feasibility of such a sequence requires that the number of elements in each combination does not exceed m and that each job appears exactly in one or in consecutive combinations in S (nonpreemptability). It has been shown in [7] that for concave job models and the makespan minimization problem, it is sufficient to consider feasible sequences of combinations $Z_k, k = 1, 2, \ldots, n - m + 1$, composed of exactly m jobs each. In the case of the mean flow time and the maximum lateness, more general feasible sequences have to be considered, i.e. sequences of n combinations composed of at most m jobs. For a given feasible sequence S of jobs on machines, we can find an optimal continuous resource allocation, i.e. an allocation that leads to a schedule minimizing the given criterion from among all feasible schedules generated by S. To this end, a convex mathematical programming problem has to be solved, in the general case (see [7]). An optimal schedule for a given feasible sequence (i.e. a schedule resulting from an optimal continuous resource allocation for this sequence) is called a *semi-optimal schedule*. In consequence, a globally optimal schedule can be found by solving the continuous resource allocation problem optimally for all feasible sequences. Unfortunately, in general, the number of feasible sequences grows exponentially with the number of jobs. Therefore it is justified to apply local search metaheuristics, such as tabu search, operating on the set of all feasible sequences (see [5]).

3 Tabu search

Tabu search is a metastrategy based on neighbourhood search with overcoming local optimality. It works in a deterministic way, trying to model human memory processes. Memory is implemented by the implicit recording of previously seen solutions, using simple but effective data structures. Tabu search was originally developed by Glover and a comprehensive report of the basic concepts and recent developments is given in [2].

3.1 Solution representation

A feasible solution for TS is represented by a feasible sequence defined in the previous section.
E.g. if $n = 6$ and $m = 3$, then in the makespan minimization problem a feasible sequence may have the form:

$$S = \{1, 2, 3\}, \{2, 3, 4\}, \{2, 3, 5\}, \{3, 5, 6\}$$

which means that jobs 1, 2 and 3 are processed in parallel in the first interval, jobs 2, 3 and 4 - in the second interval, jobs 2, 3 and 5 - in the third one and jobs 3, 5 and 6 in the last interval. In the case of the mean flow time or the maximum lateness, the form of a feasible sequence may be the following:

$$S = \{1, 2, 3\}, \{2, 3, 4\}, \{2, 3, 5\}, \{3, 5, 6\}, \{3, 5\}, \{5\}.$$

3.2 Starting solution

A starting solution is generated in two steps. In the first step jobs in particular combinations are generated randomly. In the second step the obtained solution is transformed according to the vector of processing demands (or duedates) in order to construct a solution with a better value of the considered scheduling criterion. More precisely, the following transformation is made:

- for the makespan minimization problem, the job which occurs the largest number of times in the random solution is replaced by the job with the largest processing demand and so on; in consequence, a job with a larger processing demand appears in a larger number of combinations which, intuitively, should lead to better schedules;
- for the mean flow time minimization problem, the job which ends in the first combination is replaced by the job with the smallest processing demand and so on; in consequence, a job with a smaller processing demand is finished earlier which should be advantageous in this case;
- for the maximum lateness minimization problem, the job which ends in the first combination is replaced by the job with the earliest duedate and so on; in consequence, a job with an earlier duedate is finished earlier which is, obviously, profitable in the case of this scheduling criterion.

3.3 Objective function

The value of the objective function for a feasible solution is defined as the value of the considered scheduling criterion in the semi-optimal schedule for the corresponding feasible sequence.

3.4 Stop criterion

The stop criterion is defined as an assumed number of visited solutions.

3.5 Tabu list management

A neighbour of a current solution is obtained by replacing a job in a chosen combination by another job. A job may be replaced only in either the first or the last combination in the sequence of combinations it occurs (nonpreemptability), provided that these combinations are different (each job must be executed). A move leading from a solution to a neighbouring one is described by 3 attributes:

(*number of a combination, job replaced, job introduced*).

E.g. if we move from a solution {1,2,3}, {2,3,4} to a solution {1,2,4}, {2,3,4}, then the move representing this transition has the following attributes: (1,3,4).
The tabu list is managed according to the Reverse Elimination Method [2]. In the further research also another two methods have been tested: the Tabu Navigation Method and the Cancellation Sequence Method.

4 Computational experiments

Several computational experiments have been performed to evaluate the efficiency of the proposed tabu search algorithms for the considered classes of discrete-continuous scheduling problems. The algorithms

(n,m)	TS	SA	GA
(10,2)	46/0.008%	24/0.011%	26/0.017%
(10,3)	53/0.044%	5/0.153%	13/0.126%
(10,4)	65/0.077%	15/0.219%	16/0.242%
(15,2)	47/0.030%	4/0.094%	5/0.036%
(20,2)	54/0.028%	1/0.098%	0/0.078%

Table 1: Computational results for the makespan

(n,m)	TS	SA	GA
(10,2)	94/0.492%	1/2.536%	6/1.582%
(10,3)	95/2.207%	1/3.528%	4/1.990%
(10,4)	77/1.672%	3/3.567%	43/1.910%
(15,2)	99/0.0004%	0/4.801%	1/4.134%
(20,2)	100/0.000%	0/7.640%	0/7.152%

Table 2: Computational results for the mean flow time

have been implemented in C++ and have run on SGI PowerChallenge XL with 12 RISC 8000 processors in the Poznań Supercomputing and Networking Center. The instances have been generated randomly for the number of jobs $n = 10, 15, 20$ and the number of machines $m = 2, 3, 4$. In the case of the makespan the results are compared to optimal solutions (found by the full enumeration procedure), in the cases of the mean flow time and the maximum lateness the results are compared to the best solutions found by any of the metaheuristics tested. In the latter case a truncated experiment has been performed because of the difficulty in solving the continuous part of the problem.

In Table 1 the results obtained for the makespan minimization problem are presented [5]. For each problem size 100 instances have been generated. In this table the number of optimal solutions found/the average relative deviation from optimum in percent are given for each metaheuristic.

In Table 2 the results obtained for the mean flow time minimization problem are shown in the same form as above, but optimum is replaced by the best solution known [3].

In Table 3 the results for the maximum lateness minimization problem are presented [4]. In this case only the number of best solutions found is given, where the number of instances generated for $m = 2, 3$ and 4 machines is equal to, respectively: 21, 14 and 7.

Table 4 presents the results for the makespan minimization problem solved using the tabu list management methods tested [6]. The results are given in the same form as in Table 1.

The results show that the tabu search metaheuristic is a very efficient algorithm for the considered problems. The results obtained by tabu search are significantly better than the ones obtained by the other two metaheuristic, both in terms of the number of optimal (best known) solutions found and in

(n,m)	TS	SA	GA
(10,2)	16	2	7
(10,3)	9	0	5
(10,4)	7	0	0

Table 3: Computational results for the maximum lateness

(n,m)	TNM	CSM	REM
(10,2)	59/0.002%	52/0.006%	46/0.008%
(10,3)	61/0.027%	53/0.066%	53/0.044%
(10,4)	73/0.048%	66/0.103%	65/0.077%
(15,2)	70/0.006%	56/0.025%	47/0.030%
(20,2)	67/0.030%	57/0.041%	54/0.028%

Table 4: Computational results for tabu list management methods

terms of the average relative deviation. For the makespan minimization problem the results produced by tabu search are very close to optimum, allowing to regard the proposed TS algorithm as an almost optimal heuristic for this problem. As far as the tabu list management methods are concerned, the TNM performs best but the differences are not very significant making the algorithm very effective regardless of the applied method.

5 Conclusions

In this paper discrete-continuous problems of scheduling nonpreemptable, independent jobs on parallel, identical machines are considered. The makespan, the mean flow time and the maximum lateness are considered as the scheduling criteria. A tabu search approach to the defined problems is discussed. Three tabu list management methods are tested on the makespan minimization problem. Several computational experiments are described in the paper. The results confirm a very good efficiency of the proposed tabu search algorithms for the discrete-continuous scheduling problems. The results obtained by TS are superior to the results produced by both simulated annealing and genetic algorithms.

References

[1] Burkow W.N. Raspriedielenije riesursow kak zadacza optimalnogo bystrodiejstwia. *Avtomat. I Tielemieh.*, 27, 7, 1966.

[2] Glover F., Laguna M. *Tabu Search.* Kluwer Academic Publishers, Norwell, 1997.

[3] Józefowska J., Mika M., Różycki R., Waligóra G., Węglarz J. Discrete-continuous scheduling to minimize the mean flow time - computational experiments. *Computational Methods in Science and Technology*, 3, pages 25–37, 1997.

[4] Józefowska J., Mika M., Różycki R., Waligóra G., Węglarz J. Discrete-continuous scheduling to minimize maximum lateness. *Proc. of the Fourth International Symposium on Methods and Models in Automation and Robotics MMAR'97*, Międzyzdroje 26-29.08.1997, pages 947–952, 1997.

[5] Józefowska J., Mika M., Różycki R., Waligóra G., Węglarz J. Local search metaheuristics for some discrete-continuous scheduling problems. *European J. Oper. Res.*, 107, 2, pages 354–370, 1998.

[6] Józefowska J., Waligóra G., Węglarz J. Tabu list management methods for a discrete-continuous scheduling problem. *European J. Oper. Res.*, special issue on the 12th European Chapter on Combinatorial Optimization ECCO XII, Bendor, France, 27-29.05.1999, to appear.

[7] Józefowska J., Węglarz J. On a methodology for discrete-continuous scheduling. *European J. Oper. Res.*, 107, 2, pages 338–353, 1998.