A very efficient Tabu search procedure for resource-constrained project scheduling problems

Michel Gagnon, Fayez Boctor and Gilles d'Avignon
Faculté des sciences de l'administration, Université Laval, Canada
fayez.boctor@fsa.ulaval.ca

Abstract
This paper presents a new and efficient Tabu search procedure to solve the Resource-Constrained, Project-Scheduling problem with either single or multiple execution modes and with variable resource availability. It investigates the efficiency of some neighborhood structures and of bounds used to reduce the computational time. Results show that the proposed Tabu search provides the best-known solutions to two benchmark problem sets.

1. Introduction

Resource-Constrained Project Scheduling (RCPS) consists of determining start times of a set of *J* activities. Generally, the objective function is to minimize the project duration without violating neither precedence nor resource constraints. Each activity may require one or more resources of different types. Under the non-preemption assumption, activities, once started, cannot be interrupted. In the Multiple-mode, Resource-Constrained Project Scheduling (MRCPS), each activity can be accomplished in one of several execution modes. Each execution mode is characterized by a known duration and given resource requirements. Hereafter, we consider the case where resource availability varies during the project duration. It is conceivable to use exact solution procedures only for small size projects (e.g., up to 50 activities) if it is necessary to find an optimal solution. For larger problems one needs to use heuristic procedures.

Recently, some researchers proposed local search methods to solve the RCPS and the MRCPS problems. For the RCPS problem, Pinson *et al.* (1994) proposed a first adaptation of the Tabu search. Boctor (1996) proposed a simulated annealing algorithm. Tsai and Gemmill (1998) proposed an adaptation of Tabu search based on the criticality of activities. Baar *et al*. (1997) proposed another adaptation of Tabu search and Hartman (1997) developed a genetic algorithm. For the MRCPS problem, Boctor (1996) proposed an adaptation of simulated annealing algorithm and Mori and Tseng (1997) proposed a genetic algorithm.

In the last decade, many researchers have suggested ways to further improve the performance of Tabu search procedure. Hereafter we will also introduce some new ideas to achieve the same goal.

The objectives of this presentation are to evaluate transformation functions (neighborhood definitions) embodied in Tabu search procedure and to adapt this procedure to the structure of the RCPS and MRCPS problems in order to provide a sufficiently quick and efficient solutions.

2. The proposed Tabu search procedure

Tabu search, proposed by Hansen (1986) and Glover (1989, 1990), is a local search improvement heuristic that progressively transforms a current solution into a new and hopefully better one. For a review of the latest developments in this area, we refer the reader to Glover and Laguna (1997). The Tabu search heuristic examines a predefined neighborhood of the current solution and tries to find a better solution. It may accept a less favorable solution to avoid getting trap in local optima. This is achieved by transforming a current solution $\boldsymbol{x}$ into a new one $\boldsymbol{x'}$, an operation often called a move and designated below as a transformation function $\boldsymbol{M}(\boldsymbol{x})$. The set of feasible solutions, which can be obtained from a current solution x by applying a transformation function $M(x)$, is called the neighborhood of x, noted as $\boldsymbol{V}(\boldsymbol{x})$. A solution in this set is called a neighbor of x. Note that some Tabu adaptations consider transformation functions that may yield unfeasible solutions.

In evaluating the neighborhood of a current solution x, one can systematically evaluate all the solutions belonging to $V(x)$ in order to improve the current solution x. But, an important amount of computation time is then needed. The strategy adopted here restricts the search to a small subset of the neighborhood, designated by $\boldsymbol{V}^*(\boldsymbol{x})$, where $V^*(x) \subseteq V(x)$. A second strategy is to stop the evaluation of a neighbor in $V^*(x)$ as soon as we know it will not yield a better objective function than that of the best solution obtained so far. This is achieved by applying bounds on the project duration.

In the proposed Tabu search procedure, a solution x is associated with a list of activities represented by a J-tuple $\{j_1,..., j_i, ..., j_J\}$ called a sequence where j_i indicates the number of the activity in position i in the sequence x. We also use $p(j)$ to indicate the position of activity j in the list x. Applying a list scheduling procedure, one can derive the schedule associated with a sequence x. To generate a neighbor, a transformation function selects one or more activities in the sequence and changes their positions. One can show that every sequence obtained by modifying activity positions results in a feasible solution if the precedence relations are respected. So, in our Tabu search adaptation and in order to generate a feasible sequence, the selected activity must be moved to a position that neither precedes any of its predecessors nor succeeds any of its successors. For a given sequence x, this is achieved by verifying the maximal position of all predecessors of activity j, denoted by Lj, and the minimal position of the successors of j, denoted by Hj (see Boctor 1996). In defining the transformation functions, every new sequence y obtained by $M(x)$ must be different of the sequence x.

3. Description of the transformation functions and neighborhood structure

We consider four transformation functions for the RCPS problem. Furthermore, they have been adapted to the MRCPS problem by eventually modifying the execution mode of the activities to be moved as well as the other activities. These functions modify a sequence x as follows:

1. The *Insert* transformation function: randomly selects an activity j in the sequence x and moves it from position $p(j)$ to a new position $p'(j)$ selected randomly in the interval (L_j+1, H_j-1).

2. The *Swap* transformation function: randomly selects an activity j and then randomly selects another activity j' among those occupying a position in the interval (L_j+1, H_j-1) and such that $L_{j'} < p(j) < H_{j'}$. Finally, swaps the positions of these two activities.

3. The *Left Shift* transformation function: randomly selects an activity j in the sequence x and inserts it in a new position $p'(j)$ chosen randomly in the interval (L_j+1, $p(j)$-1).

4. The *Exchange* transformation function: randomly selects a critical activity j and a non-critical activity j' among those occupying a position in the interval (L_j+1, $p(j)$-1) and such that $L_{j'} < p(j) < H_{j'}$. Then, it exchanges their positions.

To deal with the MRCPS (multi-mode, resource-constrained project scheduling) problem, we need to decide whether or not to change the execution modes for the neighbor sequence. Two strategies can be used in this case:

- Either to use a heuristic rule to select, for each activity, one of its possible execution modes;
- Or to keep the execution modes without any changes.

Based on the above, in our Tabu search procedure for solving the MRCPS problem we use the following transformation functions:

5. This transformation function is identical to the transformation function #1 but the choice of the execution mode to be used is made according to the selection rule SFM (shortest feasible execution mode, see Boctor 1993) which, for each activity *j*, selects the shortest feasible one.

6. This transformation function is identical to the transformation function #2. The execution modes of all activities are kept without any changes.

7. This transformation function is identical to the transformation function #2 but the choice of the execution modes to be used in the new sequence is made according to the selection rule SFM.

4. Computational results and performance evaluation

The proposed Tabu search algorithms were coded using Visual Basic on an IBM ThinkPad 385-XD with Intel Pentium 233 MHz processor running under the Windows 95 operating system. Two stopping criteria were tested: (1) stops when the number of evaluated solutions (iterations) reaches a given predetermine value, or (2) stops if no better solution is found during a predetermined number of iterations.

For the single-mode problems, the set of 110 test problems assembled by Patterson (1984), for which we know the optimal duration, was used and the results were compared to those obtained by previous research works. The main performance criteria are the average, the minimal and the maximal percentage increase of project duration over the optimal duration, the number of times our Tabu procedure gave the optimal duration and the average CPU time.

The best results were obtained using the "*Insert"* transformation function (transformation function #1). When the initial schedule is generated by a parallel heuristic procedure using the MINSLK rule and the maximum number of visited solutions without improvement is set to 100, the average percentage increase in project duration was 0.5584 % with a mean execution time of 0.038 seconds. In this case the average number of evaluated (visited) solutions was 451.

If we increase the maximum number of visited solutions without improvement to 1100 instead of 100, we reduce the average percentage increase over the minimal duration to zero and we get the optimal solution for each of the 110 test problems. In this case, the average percentage CPU time increases to 0.335 seconds.

For the multi-mode problem, the set of 120 test problems with 50 activities generated by Boctor (1993) was used. The results were compared to those obtained by Boctor (1993, 1996a, 1996b). The main performance criteria are the average, the minimal and the maximal percentage increase of project duration above the critical path length (calculated using the shortest duration modes), the number of times the best solution was obtained and the average CPU time.

Again, the "*Insertion*" transformation function combined with the SFM mode selection rule (transformation function #5) improves the duration of all the 120 test problems. The maximum number of iterations without finding a better solution was set to 100. The average percentage increase in project duration over the critical path length is 25.6 % with a mean execution time of 0.589 s. The average number of solutions evaluated was 1 222.

References

Baar, T., Brucker, P. and Knust, S., 1997, Tabu-search algorithms for the resource-constrained project scheduling problem, Working Paper, Universität Osnabrück.

Boctor, F. F., 1993, Heuristics for Scheduling Projects with Resource Restrictions and Several Resource-Duration Modes, *International Journal of Production Research*, 31(11): 2547-2558.

Boctor, F. F., 1996a, A New and Efficient Heuristic for Scheduling Projects with Resource Restrictions and Multiple Execution Modes, *European Journal of Operational Research*, 90: 349-361.

Boctor, F. F., 1996b, Resource-Constrained Project Scheduling by Simulated Annealing, *International Journal of Production Research*, 34(8): 2335-2351.

Glover, F., 1989, Tabu Search - Part I, *ORSA Journal of Computing*, 1: 190-206.

Glover, F., 1990, Tabu Search - Part II, *ORSA Journal of Computing*, 2: 4-32.

Glover, F. and Laguna, M., 1997, *Tabu Search*, Kluwer Academic Publishers.

Hansen, P., 1986, The steepest Ascent Mildest Descent Heuristic for Combinatorial Programming, *Numerical methods in combinatorial programming conference*, Capri, Italy.

Hartman, S., 1997, Project scheduling with Multiple Modes: A Genetic Algorithm, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, no. 435

Mori, M. and Tseng, C. C., 1997, A Genetic Algorithm for Multi-mode Resource Constrained Project Scheduling Problem, *European Journal of Operational Research*, 100: 134-141.

Patterson, J. H., 1984, A Comparison of Exact Approaches for Solving the Multiple Constrained Resource, Project Scheduling Problem, *Management Science*, 30(7): 854-867.

Pinson, E., Prins, C. and Rullier, F. 1994, Using Tabu Search for Solving the Resource-Constrained Project Scheduling Problem, Proceedings of the 4th international workshop on project management and scheduling, Leuven, Belgium, pp-102-106

Tsai, Y. and Gemmil, D. D., 1998, A Tabu Search to Schedule Activities of Stochastic Resource-Constrained Projects, *European Journal of Operational Research*, 111: 129-141.