PROACTIVE RESOURCE ALLOCATION HEURISTICS FOR ROBUST PROJECT SCHEDULING

Kristof Braeckmans, Erik Demeulemeester and Willy Herroelen
Department of Applied Economics, K.U.Leuven, Belgium
<first name>.<family name>@econ.kuleuven.ac.be

Abstract The well-known deterministic resource-constrained project scheduling problem (RCPSP) involves the determination of a *predictive schedule (baseline schedule or pre-schedule)* of the project activities that satisfies the finish-start precedence relations and the renewable resource constraints under the objective of minimizing the project duration. This pre-schedule serves as a baseline for the execution of the project. During execution, however, the project can be subject to several types of disruptions that may disturb the baseline schedule. Management must then rely on a *reactive scheduling* procedure for revising or reoptimizing the pre-schedule.

The objective of our research is to develop proactive heuristics for allocating resources to the activities of a given baseline schedule in order to maximize its stability. We develop several integer programming based models and heuristics such that the stability of the baseline schedule is maximized. We report on computational results on a set of benchmark problems.

Keywords: RCPSP, resource allocation, MIP

1. Stable resource allocation

We assume that the uncertainty during project execution resides in the activity durations that we assume to be beta distributed. Using average activity durations, a minimum duration baseline schedule is computed by the branch-and-bound algorithm of Demeulemeester and Herroelen, 1992. During project execution, these stochastic durations may take on a realization different from their expected values so that the initial schedule needs to be repaired.

The way in which resources are passed on between the various activities in the baseline schedule can be described by a resource flow network (Artigues and Roubellat, 2000). During project execution, these resource flows affect the propagation of the schedule disruptions throughout the schedule. We assume that the resource flows are kept unchanged during project execution. Although this is a very strict policy, we can easily defend this assumption with the example of a scarce resource with a high set-up cost such as a crane.

Given a baseline schedule, our objective is to generate the resource flows such that the stability of the baseline schedule is maximized. As a measure for stability, we use $\sum_{i \in N} w_i |E(S_i) - s_i|$, the sum of the weighted absolute deviations of the start times s_i in the baseline schedule from the expected actual start times S_i realized during project execution.

2. Heuristics based on IP programming

To compute our feasible resource flow networks, we make use of linear and integer programming models. As the stability measure contains stochastic variables, which we cannot use as an input for an IP solver, we developed heuristic linear integer programming models with alternative linear objective functions.

For example, we can minimize the number of extra flow carrying arcs that are added to the model. Alternatively, the weighted sum of the total floats over all activities can be maximized.

3. Computational Experiment

We compare the results obtained by the IP models with the results obtained by previously developed resource flow network heuristics (Artigues and Roubellat, 2000 and Policella et al., 2004). All heuristics are encoded in C++ and we use ILOG CPLEX 8.1 to solve our IP models. We report on computational results obtained on a set of benchmark problems.

References

Artigues, C. and Roubellat, F. (2000). A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *European Journal of Operational Research*, 127:294–316.

Demeulemeester, E. and Herroelen, W. (1992). A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, 38:1803–1818.

Policella, N., Oddi, A., S.F.Smith, and Cesta, A. (2004). Generating robust partial order schedules. In *Proceedings CP '04*.