A NEW BRANCH–AND–BOUND–BASED HEURISTIC FOR RESOURCE–CONSTRAINED PROJECT SCHEDULING WITH MINIMAL AND MAXIMAL TIME LAGS

Christoph Schwindt, Klaus Neumann

Institut für Wirtschaftstheorie und Operations Research
University of Karlsruhe
Kaiserstraße 12, D–76128 Karlsruhe, Germany
tel.: +49 (721) 608 3226, fax: +49 (721) 608 3082
e–mail: schwindt@wior.uni-karlsruhe.de

1 Introduction

In the Resource–Constrained Project Scheduling Problem with minimal and maximal time lags (RCPSP/max), both minimal and maximal time lags between the start of successive activities have to be observed. Minimal time lags may differ from the respective activity durations. Throughout its duration, each activity takes up a constant amount of scarce renewable resources. A constant amount of each resource is assumed to be available. The objective is to schedule activities subject to time and resource constraints such that the project duration is minimized. Activities and time constraints are represented by an activity–on–node network in which the introduction of maximal time lags generates strong components (cycle structures). Although maximal time lags are essential for modelling important practical applications (deadlines, overlapping operations, time–varying resource requirements of activities and time–varying resource availabilities, cf. Neumann & Schwindt 1995), little attention has been given to RCPSP/max thus far. The only exact algorithm for RCPSP/max is the branch–and–bound procedure offered by Bartusch et al. (1988). Brinkmann & Neumann (1995) propose two different heuristic approaches for RCPSP/max, both being based on truncated branch–and–bound algorithms for RCPSP. The direct method processes activities successively. The contraction method first constructs a feasible subschedule for each cycle structure in the underlying project network separately. After that, the cycle structures are replaced by single nodes corresponding to the execution of all activities of the respective cycle structure (contraction of cycle structures). The resulting instance of RCPSP then can be approximately solved by truncated versions of branch–and–bound algorithms for RCPSP (scheduling of the acyclic contracted network). Computational experience has shown that the approach of the contraction method clearly outperforms the direct method. In particular, the contraction method finds a feasible schedule markedly more often than the direct method. Since an instance of RCPSP/max is feasible if and only if there is a feasible subschedule for each cycle structure, the bottom–up approach of the contraction method which emphasizes on scheduling individual cycle structures seems to be well–suited to solve hard problem instances.

In the following, we propose a new contraction method for RCPSP/max based on (a truncated version of) the branch–and–bound algorithm provided by Demeulemeester & Herroelen (1992) which currently has to be considered as the most advanced exact method for RCPSP. In contrast to the heuristics of Brinkmann & Neumann (1995), in our contraction method the resource requirements of a contracted cycle structure are not supposed to be constant in time. We introduce additional decision points based on the resource requirement profiles which we have obtained by the separate scheduling of cycle structures. Furthermore, in the course of scheduling the contracted network the subschedules which have been constructed for the cycle structures are appropriately stretched by right–shifting activities whose resource requirements would prevent the execution of the subschedule at the current decision point. During the scheduling of the cycle structures and of the contracted network good lower bounds on the minimum project duration are determined using a dualization recently introduced by Mingozzi et al. (1994).

2 Scheduling cycle structures

The scheduling of each cycle structure represents a problem of type RCPSP/max for which we have adapted a truncated version of the branch–and–bound method of Demeulemeester and Herroelen (DH). The DH procedure is based on the extension of partial schedules corresponding to the nodes of the search tree. Resource conflicts are solved by delaying activities introducing additional precedence constraints. The set of minimal delaying alternatives implies the branching into new nodes. Two efficient dominance rules are used to avoid the generation of dominated nodes. We sketch two essential modifications of the DH procedure which we made for the scheduling of cycle structures: the computation of decision points t and the fathoming of nodes in the search tree.

<u>Decision points</u>
Let S_t be the set of activities in progress at time t in the partial schedule PS_t under consideration. With ST_i we denote the start time of activity $i \in S_t$ in PS_t. The non–preemptable duration of activity i is given by D_i. $\mathcal{S}(i)$ stands for the set of activities j for which a minimal time lag between the starts of activities i and j has to be observed. Since generally, the minimal time lags T_{ij}^{min} between the starts of activities i and j do not correspond to duration D_i of activity i the computation of the next decision point t after the current decision point t' has to be adjusted as follows:

$$t := \min\left\{\min_{i \in S_{t'}}(ST_i + D_i), \min_{i \in S_{t'}, j \in \mathcal{S}(i)}(ST_i + T_{ij}^{min})\right\}.$$

The next activity cannot be scheduled until the first activity i in progress at time t' has been finished or an additional activity j becomes eligible.

<u>Fathoming of nodes</u>
In the DH algorithm, nodes p of the search tree are fathomed if the corresponding lower bound $LB(p)$ is greater than or equal to the current upper bound UB or if the partial schedule belonging to node p is dominated by other partial schedules according to the left–shift dominance rule or the cutset dominance rule. If we have to cope with maximal time lags, the search tree additionally can be pruned every time a lower bound on the time lag between the start of a scheduled activity i and an unscheduled activity j exceeds the corresponding maximal time lag T_{ij}^{max}.

3 Scheduling the contracted network

Let $\mathcal{C}$ be the set of cycle structures C of the project network N. $V(C)$ represents the set of all activities belonging to cycle structure C. With n_C we denote the cardinal number of $C \in \mathcal{C}$. Once feasible subschedules S_C have been determined for all cycle structures $C \in \mathcal{C}$, the contracted network $N_\mathcal{C}$ is obtained by shrinking cycle structures C to single nodes c (contraction nodes) corresponding to the execution of S_C. Let $V^\mathcal{C}$ be the set of contraction nodes c which correspond to cycle structures $C \in \mathcal{C}$. For an appropriate linking of nodes $c \in V^\mathcal{C}$ in $N_\mathcal{C}$ we refer to Brinkmann & Neumann (1995). As for the scheduling of cycle structures we briefly sketch the adjustments of the DH procedure which are necessary for the scheduling of the contracted network $N_\mathcal{C}$.

Decision points

Let ST_i^C be the start time of activity i in subschedule S_C. Then, $V_C(\tau) := \{i \in V(C)| \tau - D_i < ST_i^C \leq \tau\}$ represents the set of activities belonging to cycle structure C which are in progress in S_C at time τ. Brinkmann & Neumann (1995) calculate durations $D_c := \max_{i \in V(C)}(ST_i^C + D_i)$ and resource requirements $r_{c\kappa} := max_{\tau=0,\ldots,D_c}(\sum_{i \in V_C(\tau)} r_{i\kappa})$ for each node $c \in V^\mathcal{C}$. Hence, resource requirements of contracted cycle structures are set to be constant in time, and the contracted network can be scheduled by any algorithm for RCPSP. This assumption, however, may prevent the scheduling of an eligible node $c \in V^\mathcal{C}$ although the remaining resource availabilities would have allowed the processing of subschedule S_C. In order to avoid schedules which are not maximal w.r.t. the resource constraints we modify the computation of the next decision point t at time t' as follows: Let $FT_i^C = ST_i^C + D_i$ be the finish time of activity i w.r.t. subschedule S_C. Then,

$$t_1 := \min\left\{ \min_{i \in S_{t'} \setminus V^\mathcal{C}} (ST_i + D_i), \min_{c \in S_{t'} \cap V^\mathcal{C}} \left(ST_c + \min_{i \in V(C)} \{FT_i^C | ST_c + FT_i^C \geq t'\} \right) \right\}$$

represents the first time after t' for which an activity i in progress at time t' is finished (including activities of cycle structures C whose contraction node c has been scheduled). With

$$t_2 := \min_{i \in S_{t'}, j \in \mathcal{S}(i)} (ST_i + T_{ij}^{min})$$

we denote the first time after t' for which a node j of the contracted network $N_\mathcal{C}$ becomes eligible. Then, we set the next decision point t to be $t := \min\{t_1, t_2\}$.

Stretching of cycle structures

Let $c \in V^\mathcal{C}$ be an unscheduled contraction node eligible at time t. Since the resource requirements $r_{c\kappa}(\tau) := \sum_{i \in V_C(\tau)} r_{i\kappa}$ of c at time τ depend on the start times ST_i^C of activities $i \in V(C)$, it may be favourable to delay the processing of some activity $j \in V(C)$ if $r_{c\kappa}(ST_j^C)$ exceeds the remaining availability of resource κ at time $t + ST_j^C$. The problem of right–shifting activities $i \in V(C)$ such that the completion time of c is minimized, the sequence of activities given by schedule S_C remains unchanged, and all maximal time lags T_{ij}^{max} are met, can be approximately solved by a label–correcting algorithm.

4 The contraction method

The outline of the contraction method is as follows:

Determine the set $\mathcal{C}$ of cycle structures in project network N.
FOR $C \in \mathcal{C}$ DO
    Find a feasible subschedule $S_C = (ST^C_{i_1}, \ldots, ST^C_{i_{n_C}})$ for cycle structure C.
END (* FOR *).
Construct contracted network $N_{\mathcal{C}}$.
Determine a schedule $S = (ST_{j_1}, \ldots, ST_{j_s})$ for contracted network $N_{\mathcal{C}}$.
Update subschedules S_C if cycle structures C are stretched.
FOR $C \in \mathcal{C}$ DO
    Let $ST_i := ST_c + ST^C_i \forall i \in V(C)$.
END (* FOR *).

References

[1] Bartusch, M., Möhring, R.H., Radermacher, F.J. (1988): Scheduling project networks with resource constraints and time windows. *Ann. Oper. Res. 16*, 201–240

[2] Brinkmann, K., Neumann, K. (1995): Sequential and Contraction B&B–Based Heuristics for RLP/max and RCPSP/max. *Report WIOR–471*, Institut für Wirtschaftstheorie und Operations Research, University of Karlsruhe

[3] Demeulemeester, E., Herroelen, W. (1992): A branch–and–bound procedure for the multiple resource–constrained project scheduling problem. *Mgmt Sci. 38*, 1803–1818

[4] Mingozzi, A., Maniezzo, V., Ricciardelli, S., Bianco, L. (1994): An exact algorithm for project scheduling with resource constraints based on a new mathematical formulation. *Working Paper*, University of Bologna

[5] Neumann, K., Schwindt, C. (1995): Projects with minimal and maximal time lags – Construction of activity–on–node networks and applications. *Report WIOR–447*, Institut für Wirtschaftstheorie und Operations Research, University of Karlsruhe