# A Branch-and-Bound Procedure for the Resource-Constrained Project Scheduling Problem with Partially Renewable Resources and Time Windows

Kai Watermeyer and Jürgen Zimmermann

Clausthal University of Technology, Germany
kai.watermeyer, juergen.zimmermann@tu-clausthal.de

**Keywords:** Project scheduling, Partially renewable resources, Branch-and-bound.

## 1 Introduction

In this paper we present a branch-and-bound procedure for the resource-constrained project scheduling problem with partially renewable resources and time windows (RCPSP/max,$\pi$). For the first time the concept of partially renewable resources is embedded in the context of projects with general temporal constraints.

Partially renewable resources were introduced by Böttcher *et al.* (1996) and have just been considered for projects restricted to precedence constraints (RCPSP/$\pi$). For each partially renewable resource a resource capacity for a subset of time periods of the planning horizon is given. In this way timetabling and complex labor regulation problems can be modeled as project scheduling problems (Álvarez-Valdés *et al.* 2006). For the RCPSP/$\pi$ a branch-and-bound procedure has been developed in Böttcher *et al.* (1999) and also approximation procedures in Schirmer (1999) and Álvarez-Valdés *et al.* (2006, 2008) have been investigated.

In Section 2 the RCPSP/max,$\pi$ is described formally. Section 3 presents the enumeration scheme the developed branch-and-bound procedure is based on and in Section 4 the branch-and-bound procedure is outlined. Finally, in Section 5 the results of a computational study are presented where we compared the performance of our branch-and-bound procedure with the outcome of the mixed-integer linear programming solver *IBM CPLEX*.

## 2 Problem description

The resource-constrained project scheduling problem with time windows and partially renewable resources (RCPSP/max,$\pi$) can be modeled as an activity-on-node network where the nodes correspond to all activities of the project $V = \{0, 1, \dots, n+1\}$ with $n$ real activities and the fictitious activities 0 and $n+1$ representing the start and end of the project, respectively. Each activity $i \in V$ is assigned a non-interuptible processing time $p_i \in \mathbb{Z}_{\geq 0}$ and a resource demand $r^d_{ik} \in \mathbb{Z}_{\geq 0}$ for each partially renewable resource $k \in \mathcal{R}$ considered in the project. The arcs of the network given by the set $E \subseteq V \times V$ represent the temporal constraints between the activities where the arc weight $\delta_{ij} \in \mathbb{Z}$ for arc $\langle i, j \rangle \in E$ implicates a minimal time lag between the start times of activity $i$ and activity $j$ which has to be fulfilled. For each resource $k \in \mathcal{R}$ a resource capacity $R_k$ and a subset of time periods of the whole planning horizon $\Pi_k \subseteq \{1, 2, \dots, \bar{d}\}$ is given with $\bar{d}$ as a given maximal project duration. It is assumed that an activity $i$ just consumes a resource $k$ with $r^d_{ik}$ units in each time period of $\Pi_k$ activity $i$ is in execution where the start times of all activities are restricted to integer values. The number of time periods an activity $i$ with start time point $S_i$ is in execution during the defined time periods of resource $k$ is given by the so called resource usage $r^u_{ik}(S_i) := |\{S_i+1, S_i+2, \ldots, S_i+p_i\} \cap \Pi_k|$ so that the corresponding resource consumption can be determined by $r^c_{ik}(S_i) := r^u_{ik}(S_i) \cdot r^d_{ik}$.

The objective of the problem is to assign each activity $i \in V$ a start time $S_i$ so that all time and resource constraints are fulfilled and the project duration is minimized. In the following a sequence of start times of all activities $S = (S_0, S_1, \ldots, S_{n+1})$ with $S_0 := 0$ is called a schedule where it is said to be time-feasible, resource-feasible or feasible if it fulfills all temporal constraints, all resource constraints or all constraints, respectively. The problem RCPSP/max,$\pi$ can be stated as follows:

$$\begin{aligned}
\text{Minimize} \quad & f(S) = S_{n+1} & \\
\text{subject to} \quad & S_j - S_i \geq \delta_{ij} & (\langle i,j\rangle \in E) \\
& S_0 = 0 & \\
& \sum\nolimits_{i \in V} r^c_{ik}(S_i) \leq R_k & (k \in \mathcal{R}) \\
& S_i \in \mathbb{Z}_{\geq 0} & (i \in V)
\end{aligned}$$

## 3 Enumeration scheme

The enumeration scheme of the developed branch-and-bound procedure is based on a stepwise restriction of the allowed resource usages of the activities of the project. The procedure starts with the determination of the earliest possible start times $ES_i$ of all activities $i \in V$ for the resource-relaxation of RCPSP/max,$\pi$. If this schedule is resource-feasible the optimal solution is already found. Otherwise there is at least one resource $k$ whose resource capacity $R_k$ is exceeded so that the resource usage of at least one activity consuming resource $k$ have to be decreased to get a feasible schedule. The enumeration scheme makes use of the start time dependency of the resource usage $r^u_{ik}(\cdot)$ of all activities $i \in V$ for resource $k$. It is easy to see that for a feasible schedule $S$ the resource usage of at least one activity $i \in V$ has to be lower than the resource usage of the resource-infeasible schedule $ES$, i.e., $r^u_{ik}(S_i) \leq r^u_{ik}(ES_i) - 1$. So we preserve all feasible schedules by branching the resource-relaxation in subproblems where each subproblem restricts the resource usage of an activity $i$ with $r^u_{ik}(ES_i) > 0$ to $r^u_{ik}(ES_i) - 1$. The resource usage restriction of activity $i$ for resource $k$ is achieved by permitting only start time points $t$ with $r^u_{ik}(t) \leq r^u_{ik}(ES_i) - 1$. In order to save these permitted start time points for all activities in the enumeration process a so called start time restriction $W_i$ for each activity is introduced. This is set to $W_i := \{ES_i, ES_i+1, \ldots, LS_i\}$ for each activity at the beginning of the process with $LS_i$ as the latest possible start time point of activity $i$ for the resource-relaxation of RCPSP/max,$\pi$. For the subproblem in which the resource usage of activity $i$ is restricted the start time restriction is set to $W_i := W_i \cap \{t \in \{0, 1, \ldots, \bar{d}\} \mid r^u_{ik}(t) \leq r^u_{ik}(ES_i) - 1\}$ so that the resource usage of activity $i$ of resource $k$ is lower or equal to $r^u_{ik}(ES_i) - 1$ if activity $i$ starts at time point $t \in W_i$. For each achieved subproblem the earliest possible start time points of all activities have to be determined so that all temporal constraints of the RCPSP/max,$\pi$ are fulfilled and also $S_i \in W_i$ for each $i \in V$ is satisfied. This can be done by a modified label correcting algorithm which determines the earliest possible start time points denoted by $ES_i(W)$ of all activities $i \in V$ with a worst-case time complexity of $\mathcal{O}(|V||E|\,(1 + \mathcal{B}))$ with $\mathcal{B}$ as the number of interruptions of consecutive time points in $W_i$ over all activities $i \in V$. If all determined and all following subproblems are tackled like described for the resource-relaxation of the RCPSP/max,$\pi$ it can be shown that the procedure determines after a finite number of iterations an optimal schedule or shows the infeasibility if there is no optimal schedule.

## 4 Branch-and-bound procedure

The enumeration scheme describes the decomposition of the currently considered part of the solution space in one or more components for a chosen conflict resource, i.e., a resource whose capacity is exceeded. The strategy to decide which of the conflict resources is used next to decompose the solution space is called branching strategy. The way to determine which node in the enumeration tree is considered next is called search strategy. For both strategies different approaches have been investigated on benchmark test sets.

Before the branch-and-bound procedure is started a preprocessing phase is conducted. In this step start time points of activities are eliminated for which it can be shown that they cannot be part of any of the feasible schedules. For this a start time point of an activity is eliminated if the resource consumption of the activity started at this time point and the sum of the minimal resource consumptions of all other activities over all start time points satisfying the temporal constraints to the considered activity exceeds the capacity of at least one resource.

Furthermore, for each node in the search tree two lower bounds for the project duration are determined to be able to prune this node and the following parts of the enumeration tree if one of these lower bounds is greater or equal to the project duration of the best found solution so far. The first lower bound is given by the minimal possible project duration taking the start time restrictions of all activities into consideration. The second lower bound is equal to the minimal project duration for which at least one resource-feasible schedule in the currently considered part of the search tree exists so that all temporal constraints to the start and the end of the project are satisfied.

To reduce the search tree even further a dominance rule is used in addition. For this an unexplored node is called dominated by another node if the restrictions of the resource usages over all activities and resources are lower or equal to the resource usage restrictions of the other node. In this case the unexplored node is pruned from the search tree.

## 5 Performance analysis

In order to evaluate the performance of our branch-and-bound (*BnB*) procedure we have compared the obtained results with the outcome of the mixed-integer linear programming (MILP) solver *IBM CPLEX* in the latest version 12.7.1. The computational study was conducted on a PC with Intel Core i7-3820 CPU with 3.6 GHz and 32 GB RAM under Windows 7. The *BnB* procedure was coded in C++ and compiled with the 64-bit Visual Studio 2015 C++-Compiler. The instance sets we have used are adaptions of the well-known benchmark test set UBO (Schwindt 1998) where we replaced the included renewable resources by 30 partially renewable resources using the generation procedure described in Schirmer (1999). Note that there is no instance with a project network containing a cycle of positive length. In this manner we have generated 729 instances with 10, 20, 50, 100, and 200 activities, respectively. For the computational study we set the runtime limit to 60 seconds and used an adaption of the MILP given in Böttcher *et al.* (1999) for the *IBM CPLEX* solver. The mathematical program is given as follows:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{t \in \mathcal{T}_{n+1}} t \cdot x_{n+1,t} & \\
\text{subject to} \quad & \sum_{t \in \mathcal{T}_i} x_{it} = 1 & (i \in V) \\
& \sum_{t \in \mathcal{T}_j} t \cdot x_{jt} \geq \sum_{t \in \mathcal{T}_i} t \cdot x_{it} + \delta_{ij} & (\langle i,j \rangle \in E) \\
& \sum_{i \in V} r^d_{ik} \sum_{v \in \Pi_k} \sum_{\tau \in Q_{i,(v-1)} \cap \mathcal{T}_i} x_{i\tau} \leq R_k & (k \in \mathcal{R}) \\
& x_{it} \in \{0,1\} & (i \in V, t \in \mathcal{T}_i)
\end{aligned}
$$

The MILP is a time-indexed formulation with binary variables $x_{it}$ for each activity $i \in V$ and each start time point $t$ of the activity in the set $\mathcal{T}_i := \{ES_i, ES_i + 1, \ldots, LS_i\}$. The binary variable $x_{it}$ takes the value 1 exactly if activity $i$ starts at time point $t$, i.e., $t = S_i$. The set $Q_{it}$ contains all time points activity $i$ could be started so that activity $i$ would be in execution at time point $t$, i.e., $Q_{it} := \{t - p_i + 1, \ldots, t\}$.

**Table 1.** Results of the computational study

| | $UBO10^\pi$ | | $UBO20^\pi$ | | $UBO50^\pi$ | | $UBO100^\pi$ | | $UBO200^\pi$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *BnB* | *CPLEX* | *BnB* | *CPLEX* | *BnB* | *CPLEX* | *BnB* | *CPLEX* | *BnB* | *CPLEX* |
| #opt | 511 | 565 | 288 | 391 | 116 | 113 | 58 | 34 | 53 | 5 |
| #feas | 55 | 1 | 259 | 160 | 352 | 65 | 333 | 6 | 312 | 1 |
| #infeas | 129 | 132 | 30 | 57 | 0 | 19 | 0 | 3 | 0 | 0 |
| #noSol | 3 | 0 | 34 | 3 | 59 | 330 | 93 | 441 | 101 | 460 |
| #trivial | 31 | 31 | 118 | 118 | 202 | 202 | 245 | 245 | 263 | 263 |
| $\varnothing^{\text{CPU}}_{\text{opt}}$ | 1.60 | 0.56 | 2.51 | 6.78 | 1.72 | 4.89 | 1.76 | 15.41 | 6.21 | 40.51 |
| $\varnothing^{\text{CPU}}_{\text{infeas}}$ | 0.44 | 0.03 | 2.31 | 0.45 | – | 2.44 | – | 14.90 | – | – |

The results of the computational study are given in Tab. 1 where for each test set, for instance $UBO10^\pi$ with 10 activities, the results of the *BnB* procedure and the *IBM CPLEX* solver (*CPLEX*) are listed. The term #opt stands for the number of optimal solved instances for which the schedule *ES* is not optimal, term #feas describes the number of instances the solution procedure was able to find a solution which could not be proofed to be optimal and #infeas gives the number of instances the procedure could proof the infeasibility for. In the following two rows, the number of instances the solution procedure was not able to find any feasible solution (#noSol) and the number of so called trivial instances for which the schedule *ES* is already optimal (#trivial) are given. Finally, the last rows show the average used CPU time in seconds over all optimal solved ($\varnothing^{\text{CPU}}_{\text{opt}}$) and over all instances for which the infeasibility could be proofed ($\varnothing^{\text{CPU}}_{\text{infeas}}$).

In Tab. 1 it can be seen that the *IBM CPLEX* solver dominates the developed BnB procedure for the instance sets $UBO10^\pi$ and $UBO20^\pi$. In contrast, the BnB procedure is able to obtain optimal and feasible solutions for more instances of the test sets $UBO50^\pi$, $UBO100^\pi$ and $UBO200^\pi$.

## References


Álvarez-Valdés R., E. Crespo, J.M. Tamarit and F. Villa, 2006, "A scatter search algorithm for project scheduling under partially renewable resources", *Journal of Heuristics*, Vol. 12, pp. 95-113.

Álvarez-Valdés R., E. Crespo, J.M. Tamarit and F. Villa, 2008, "GRASP and path relinking for project scheduling under partially renewable resources", *European Journal of Operational Research*, Vol. 189, pp. 1153-1170.

Böttcher J., A. Drexl, R. Kolisch, F. Salewski, 1996, "Project scheduling under partially renewable resource constraints", Technical Report, *Manuskripte aus den Instituten für Betriebswirtschaftslehre 398*, University of Kiel.

Böttcher J., A. Drexl, R. Kolisch and F. Salewski, 1999, "Project scheduling under partially renewable resource constraints", *Management Science*, Vol. 45, pp. 544-559.

Schirmer A., 1999, "Project scheduling with scarce resources: models, methods and applications", Dr. Kovač, Hamburg.

Schwindt C., "Generation of resource-constrained project scheduling problems subject to temporal constraints", *Technical Report WIOR-543*, University of Karlsruhe.