A Comparison of two MILP formulations for the resource renting problem

Max Reinke and Jürgen Zimmermann

Clausthal University of Technology, Germany
max.reinke@tu-clausthal.de, juergen.zimmermann@tu-clausthal.de

Keywords: Project scheduling, resource renting, mixed-integer linear programming.

1 Introduction

Typically each activity in a project uses resources during its execution. In many cases those resources have to be rented, for example heavy machinery on construction sites. The renting of resources induces two types of costs namely time-independent procurement costs and time-dependent renting costs. In this paper we present two MILP formulations for the resource renting problem with general temporal constraints (RRP/max). The RRP/max was introduced by Nübel (2001) and extends the well-known resource availability problem by taking time-dependent renting costs into account. The problem aims at minimizing the total resource costs to execute a project while taking general temporal constraints into account. In Section 2 the RRP/max is described formally. Section 3 presents two different time-indexed MILP formulations. Finally, in Section 4 the results of a preliminary computational study are presented where we compare the performance of our two models.

2 Problem description

With given renting and procurement costs for each resource type the resource renting problem (RRP/max) can be modeled as an activity-on-node network, where the nodes represent activities $V = \{0, \ldots, n+1\}$ with n real activities and the fictitious project start 0 and project end $n+1$. Each activity $i \in V$ is assigned a duration $p_i \in \mathbb{Z}_{\geq 0}$ and has to be performed without preemption. The arcs of the network given by set $E \subseteq V \times V$ represent temporal constraints between activities. Arc weights $\delta_{ij} \geq 0$ indicate a minimal time lag for arc $\langle i,j \rangle \in E$ while $\delta_{ij} < 0$ gives a maximal time lag between the start times of activities i and j. Here a project has a given deadline $\bar{d}$ introduced by $\langle n+1, 0 \rangle$ with $\delta_{n+1,0} = -\bar{d}$. A sequence of start times S_i for all activities $i \in V$ is called a schedule $S = (S_0, S_1, \ldots, S_{n+1})$, which is termed time-feasible if $S_j - S_i \geq \delta_{ij}$ for all $\langle i,j \rangle \in E$. Each activity i in a project has an earliest start time ES_i and a latest start time LS_i, so that the possible start times can be limited to the set $W_i = \{ES_i, ES_{i+1}, ..., LS_i\}$. For the execution of activity i an amount $r_{ik} \in \mathbb{Z}_{\geq 0}$ of resource $k \in \mathcal{R}$ is needed. The RRP/max considers renewable resources which have to be rented. The usage of one unit of resource k for t periods incurs procurement costs c_k^p and time-dependent renting costs of $t \cdot c_k^r$. To carry out a project according to a given time-feasible schedule S, the number of available (rented) resources has to be equal to or exceed the resource demand for all $k \in \mathcal{R}$ and $t \in T$. Thus, a renting policy $\varphi_k(S,t)$ is needed, which specifies when resources are procured or released, so that $\varphi_k(S,t) \geq \sum_{i \in V | S_i \leq t < S_i + p_i} r_{ik}$ for all $k \in \mathcal{R}$ and $t \in T$. For the case $c_k^p < c_k^r$ an optimal renting policy is to procure additional resources when the resource demand has a positive step discontinuity at t and to release idle resources immediately, since procuring new resources is less expensive than keeping unused resources. In general $c_k^p > c_k^r$ holds, here procurement costs of a resource k are higher than renting costs for one period. In this case, we can define $span = \lfloor c_k^p / c_k^r \rfloor$ as the maximum number of time periods for which it is beneficial to hold unused resources if they are used later in the project. The objective of the RRP/max is to find a schedule which minimizes the total costs incurred by renting resources. The problem can be stated as follows.

$$\begin{aligned}
&\text{Min} \quad F(S) = \sum\nolimits_{k\in\mathcal{R}} \left[c_k^r \int_0^{\bar{d}} \varphi_k(S,t)dt + c_k^p \sum\nolimits_{t\in J_k} \Delta^+\varphi_k(S,t) \right] \\
&\text{s.t.} \quad S_j - S_i \geq \delta_{ij} && (\langle i,j\rangle \in E) \\
&\qquad \varphi_k(S,t) \geq \sum\nolimits_{i\in V | S_i \leq t < S_i + p_i} r_{ik} && (t\in T, k\in\mathcal{R}) \\
&\qquad S_0 = 0 \\
&\qquad S_i \in \mathbb{Z}_{\geq 0} && (i\in V)
\end{aligned}$$

3 Mixed-integer linear programs

Both MILP formulations presented in this paper are based on well-known time-indexed formulations for the RCPSP. The first formulation based on the formulation by Pritsker *et al.* (1969) uses binary variables x_{it} for all $i \in V$ and $t \in T$ where $x_{it} = 1$ if activity i starts at point in time t. To model the resource demand, we use the variable z_{kt} representing the amount of resource $k \in \mathcal{R}$ needed at $t \in T$. The renting policy $\varphi_k(S,t)$ is modeled using positive variables a_{kt} and w_{kt}, which represent the number of units of resource k added or withdrawn at t. With these variables the model (RRP-SP) can be formulated.

$$\begin{aligned}
&\text{Min} \quad \sum\nolimits_{k\in\mathcal{R}} c_k^p \sum\nolimits_{t\in T} a_{kt} + \sum\nolimits_{k\in\mathcal{R}} c_k^r \sum\nolimits_{t\in T} t\cdot(w_{kt} - a_{kt}) && && (1.1) \\
&\text{s.t.} \quad \sum\nolimits_{t\in W_i} x_{it} = 1 && (i\in V) && (1.2) \\
&\qquad \sum\nolimits_{t\in W_j} t\cdot x_{jt} - \sum\nolimits_{t\in W_i} t\cdot x_{it} \geq \delta_{ij} && (\langle i,j\rangle\in E) && (1.3) \\
&\qquad \sum\nolimits_{i\in V} r_{ik} \sum\nolimits_{\tau=\max\{ES_i, t-p_i+1\}}^{\min\{t, LS_i\}} x_{i\tau} \leq z_{kt} && (k\in\mathcal{R}, t\in T) && (1.4) \\
&\qquad \sum\nolimits_{\tau=0}^{t} (a_{k\tau} - w_{k\tau}) \geq z_{kt} && (k\in\mathcal{R}, t\in T) && (1.5) \\
&\qquad \sum\nolimits_{t\in T} a_{kt} - \sum\nolimits_{t\in T} w_{kt} = 0 && (k\in\mathcal{R}) && (1.6) \\
&\qquad a_{kt}, w_{kt}, z_{kt} \in \mathbb{Z}_{\geq 0} && (k\in\mathcal{R}, t\in T) && (1.7) \\
&\qquad x_{it} \in \{0,1\} && (i\in V, t\in W_i) && (1.8)
\end{aligned}$$

Constraints (1.2) state that every activity has to be started exactly once. With $S_i = \sum_{t\in W_i} t\cdot x_{it}$ (1.3) ensure all temporal restrictions between activities are satisfied. Lower bounds on the minimal value for every k and t are assigned to variables z_{kt} by constraints (1.4). The available (rented) resources at time t are given by $\sum_{\tau=0}^{t}(a_{k\tau} - w_{k\tau})$, hence inequalities (1.5) ensure that a feasible renting policy $\varphi_k(S,t) \geq z_{kt}$ for all $k \in \mathcal{R}$ and $t \in T$ is obtained. Also all added resources have to be withdrawn by the end of the project, see (1.6). Alternative constraints to ensure temporal relations, are the disaggregated precedence constraints proposed by Christofides *et al.* (1987). Here we use the formulation,

$$\sum\nolimits_{\tau=t}^{LS_i} x_{i\tau} + \sum\nolimits_{\tau=ES_j}^{\min\{LS_j, t+\delta_{ij}-1\}} x_{j\tau} \leq 1 \qquad (\langle i,j\rangle \in E, t\in T) \quad (1.9),$$

by Rieck *et al.* (2012) to take minimal and maximal time lags into consideration. The second MILP formulation is based on binary on/off variables where $\mu_{it} = 1$ if activity i is in progress at t and otherwise $\mu_{it} = 0$. Here we adapt the model from Artigues (2013) to the RRP/max. By defining $K_{it} = \lfloor t/p_i \rfloor$ for all $i \in V | p_i > 0$ and otherwise $K_{it} = 0$, as the number of potential time windows of length p_i in which an activity could be executed, our second MILP model (RRP-OO) for the RRP/max can be stated as follows.

$$
\begin{aligned}
\text{Min} \quad & \sum\nolimits_{k\in\mathcal{R}} c_k^p \sum\nolimits_{t\in T} a_{kt} + \sum\nolimits_{k\in\mathcal{R}} c_k^r \sum\nolimits_{t\in T} t\cdot(w_{kt}-a_{kt}) && && (2.1)\\
\text{s.t.} \quad & \sum\nolimits_{\lambda=0}^{K_{it}} \mu_{i,t-\lambda\cdot p_i} - \sum\nolimits_{\lambda=0}^{K_{i,t-1}} \mu_{i,t-\lambda\cdot p_j-1} \geq 0 && (i\in V|p_i>0, t\in T\setminus\{0\}) && (2.2)\\
& \sum\nolimits_{\lambda=0}^{K_{i,LC_i-\phi_i}} \mu_{i,LC_i-\phi_i-\lambda\cdot p_i} = 1 && (i\in V) && (2.3)\\
& \sum\nolimits_{\lambda=0}^{K_{i,t-\delta_{ij}}} \mu_{i,t-\lambda\cdot p_i-\delta_{ij}} - \sum\nolimits_{\lambda=0}^{K_{jt}} \mu_{j,t-\lambda\cdot p_j} \geq 0 && (\langle i,j\rangle\in E|\delta_{ij}\geq 0, t\in T) && (2.4)\\
& \sum\nolimits_{\lambda=0}^{K_{ijt}^{\delta}} \mu_{i,\bar{d}-\lambda\cdot p_i-\varDelta_{ijt}} \leq 1-\mu_{jt} && (\langle i,j\rangle\in E|\delta_{ij}<0, t\in T) && (2.5)\\
& \sum\nolimits_{i\in V} r_{ik}\cdot\mu_{it} \leq z_{kt} && (k\in\mathcal{R}, t\in T) && (2.6)\\
& \sum\nolimits_{\tau=0}^{t} (a_{k\tau}-w_{k\tau}) \geq z_{kt} && (k\in\mathcal{R}, t\in T) && (2.7)\\
& \sum\nolimits_{t\in T} a_{kt} - \sum\nolimits_{t\in T} w_{kt} = 0 && (k\in\mathcal{R}) && (2.8)\\
& \mu_{00} = 1 && && (2.9)\\
& \mu_{it} = 0 && (i\in V, t\in T\setminus W_i') && (2.10)\\
& a_{kt}, w_{kt}, z_{kt} \in \mathbb{Z}_{\geq 0} && (k\in\mathcal{R}, t\in T) && (2.11)\\
& \mu_{it} \in \{0,1\} && (i\in V, t\in W_i) && (2.12)
\end{aligned}
$$

Constraints (2.2) state that each activity has to be performed during p_i consecutive periods. For all potential time windows between the project start and the latest completion LC_i of activity i, an activity can only be in progress during one of them (2.3). Here $\phi_i = 1$ if $p_i \geq 1$ else $\phi_i = 0$. Constraints (2.4), are modifications of the disaggregated precedence constraints used by Artigues (2013) and model all time lags where $\langle i,j\rangle \in E|\delta_{ij} \geq 0$, by ensuring an activity j can only be processed if activity i has been in progress at least δ_{ij} periods before. To model time lags, where $\delta_{ij} < 0$, we define $K_{ijt}^{\delta} = \lfloor(\bar{d} - t + p_i + \delta_{ij})/p_i\rfloor$ for all $i \in V|p_i > 0$ and $K_{ijt}^{\delta} = 0$ otherwise. Moreover $\varDelta_{ijt} = mod([\bar{d} - t + \delta_{ij}]/p_i)$ for all $i \in V|p_i > 0$ and $\varDelta_{ijt} = 0$ otherwise, is defined. Inequalities (2.5) ensure that for every time lag $\langle i,j\rangle \in E|\delta_{ij} < 0$, if activity i is executed during the time interval $[t + p_i + |\delta_{ij}|, ..., \bar{d}]$ activity j can not be in progress at t, since this would result in $S_j - S_i < \delta_{ij}$ and, therefore, in an infeasible schedule. The calculation of the resource demand is straightforward for this formulation, see (2.6). Constraints (2.7) and (2.8) ensure a feasible renting policy is used. (2.9) state that the project starts at $t = 0$ and (2.10) set all infeasible μ_{it} to 0 where $W_i' = \{ES_i, ES_{i+1}..., LC_i - 1\}$.

4 Lower Bounds for variables

To reduce the number of potential solutions and, therefore, limit the search space, when solving the RRP/max, additional restrictions are devised. We establish a lower bound for the minimal necessary number of resources to procure for the whole project, by defining $\sum_{t\in T} a_{kt} \geq LB_k^a$ for all $k \in \mathcal{R}$ (3.1) where $LB_k^a = \max(LB_{1,k}^a, LB_{2,k}^a, LB_{3,k}^a)$. $LB_{1,k}^a = \max_{i\in V}(r_{ik})$ gives the maximum resource demand of k for any activity i. For $LB_{2,k}^a = \lceil\sum_{i\in V} r_{ik}\cdot p_i/\bar{d}\rceil$ the total workload for every resource k is distributed equally over the time horizon of the project. The third lower bound is given by $LB_{3,k}^a = \max_{t\in T}(\sum_{i\in V_t^{nc}} r_{ik})$. Here we use the resource demand of near-critical activities $V_t^{nc} = \{i \in V|LS_i \leq t < ES_i + p_i\}$ to determine the minimal number of resources needed for the project at a given point of time. For a second type of restriction, the near-critical resource demand is used to define a bound $z_{kt} \geq \sum_{i\in V_t^{nc}} r_{ik}$ for all $(t \in T)$ (3.2), where the minimal number of units needed of resource k for every t is determined. By substituting constraints (1.3) with (1.9) and adding the restrictions (3.1) and (3.2), models (RRP-SPC-LB) and (RRP-OO-LB) are obtained.

5 Performance analysis

To compare the performance of the devised formulations a computational study was conducted. The four models were implemented in GAMS v.25.1 and solved, by using *IBM CPLEX* v.12.8.0, on a computer with an Intel Core i7-7700 with 4.2 GHz and 64 GB RAM under Windows 10. As problem instances we used adaptations of the well-known benchmark test set UBO (Schwindt 1998) where we introduced a project deadline $\bar{d} = \alpha \cdot ES_{n+1}$ with $\alpha = \{1, 1.25, 1.5\}$ and procurement costs $c_k^r = CQ \cdot c_k^p$ with $CQ = \{0.5, 0.25, 0.1\}$. For every combination of parameters α and CQ, 30 instances with $n = \{10, 20\}$ and an order strength of 0.5 were used. For the computational study a run time limit of 600 seconds was employed. The results of our study are given in Tab. 1, where for all four models the average *gap* [%] (relative deviation from the best lower bound), the average solution time [s] and the number of instances solved to optimality (max. 30) are given. Instances with a larger time horizon have more possibilities to schedule activities and therefore are harder to solve. For most parameter combinations the presented additional restrictions lead to smaller gaps and lower solution times for both formulations. Especially the larger instances with $n = 20$ show improvements. Preliminary tests for instances with greater numbers of activities $n = \{50, 100\}$, showed a decreasing performance of the OO-models compared to the SP-models.

In conclusion, when comparing the two models with additional constraints (RRP-SPC-LB) and (RRP-OO-LB) for all instances with $n = 20$, we found the solver is able to obtain better solutions in a shorter amount of time with the OO-formulation. In future research, the two formulations presented in this paper could be compared to other possible MILP formulations for the (RRP/max).

Table 1. Results of the computational study

			RRP-SP			*RRP-SPC-LB*			*RRP-OO*			*RRP-OO-LB*		
n	α	CQ	*gap*	*time*	*#opt*	*gap*	*time*	*#opt*	*gap*	*time*	*#opt*	*gap*	*time*	*#opt*
10	1	0,25	0,0	1,0	30	0,0	1,0	30	0,0	0,4	30	0,0	0,4	30
10	1,25	0,25	0,29	74,0	28	0,19	55,9	29	0,0	12,7	30	0,0	12,9	30
10	1,5	0,25	1,69	138,4	25	1,55	144,5	25	0,22	61,4	29	0,22	59,9	29
20	1	0,25	0,0	25,4	30	0,0	25,0	30	0,0	5,9	30	0,0	6,0	30
20	1,25	0,25	8,82	562,6	3	6,9	527,2	5	3,56	465,0	10	2,81	451,0	12
20	1,5	0,25	17,81	600,0	0	15,14	600,0	0	12,11	592,3	1	11,62	591,9	1

References

Artigues C., 2015, "A note on time-indexed formulations for the resource-constrained project scheduling problem", Technical Report 13206, LAAS, CNRS, Toulouse.

Christofides N., Álvarez-Valdés R., Tamarit J.M., 1987, "Project scheduling with resource constraints: A branch and bound approach", *European Journal of Operational Research*, Vol. 29, pp. 262-273.

Nübel H., 2001, "The resource renting problem subject to temporal constraints", *OR Spectrum*, Vol. 23, pp. 359-381.

Pritsker A., Watters L., Wolfe P., 1969, "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach", *Management Science*, Vol. 16, pp. 93-108.

Rieck J., Zimmermann J., Gather T., 2012, "Mixed-integer linear programming for resource leveling problems", *European Journal of Operational Research*, Vol. 221, pp. 27-37.

Schwindt C., 1998, "Generation of resource-constrained project scheduling problems subject to temporal constraints", *Technical Report WIOR-543*, University of Karlsruhe.