

A faster exact method for solving the robust multi-mode resource-constrained project scheduling problem

Matthew Bold^{*1} and Marc Goerigk²

¹STOR-i Centre for Doctoral Training, Lancaster University, Lancaster, UK

²Network and Data Science Management, University of Siegen, Siegen, Germany

Abstract

Keywords: project scheduling; optimisation under uncertainty; robust optimisation; budgeted uncertainty

1 Introduction

The multi-mode resource-constrained project scheduling problem (MRCPSP) is a generalisation of the widely-studied resource-constrained project scheduling problem (RCPSP) to include multiple processing modes for each activity. The purpose of introducing activity processing modes is to model situations in which there is more than one way of executing project activities, with each option having its own duration and resource requirements. The MRCPSP consists selecting the processing modes and start times for a given set of activities, subject to a set of precedence constraints and limited resource availability, with the objective of minimising the overall project duration, known as the makespan.

The work in the paper extends the compact reformulation for the RCPSP developed by [Bold and Goerigk \(2021\)](#) for application to the MRCPSP. A computational study is presented comparing this approach against the Benders' decomposition approach used in [Balouka and Cohen \(2021\)](#) for solving MRCPSP instances with uncertain activity durations. Note that the method used in [Balouka and Cohen \(2021\)](#) is an extension of the approach used by [Bruni et al. \(2017\)](#) to solve the RCPSP with uncertain activity durations.

* lit review *

*Corresponding author, email: m.bold1@lancaster.ac.uk

2 Problem description

A project consists of a set of non-preemptive activities $V = \{0, 1, \dots, n, n+1\}$, where 0 and $n+1$ denote the dummy source and sink activities respectively. Each activity $i \in V$ has a set of available processing modes given by $M_i = \{1, \dots, |M_i|\}$. We denote a particular choice of mode selections as $\mathbf{m} = (m_1, \dots, m_n)$, and the set of all possible mode selections by $\mathcal{M} \subseteq \mathbb{N}^n$. The nominal duration of activity i when executed in mode $m_i \in M_i$ is given by \bar{d}_{im_i} , whilst its worst-case duration is given by $\bar{d}_{im_i} + \hat{d}_{im_i}$, where \hat{d}_{im_i} is its maximum durational deviation. Each mode for activity i , $m_i \in M_i$, has an associated renewable resource requirement of r_{im_k} for each $k \in K$, where K is the set of renewable resource types involved in the project. Each renewable resource $k \in K$ has an availability of R_k at each time period in the project horizon. As well as resource constraints, the project is subject to a set of strict finish-to-start precedence constraints given by E , where $(i, j) \in E$ enforces that activity i must finish before activity j can begin. A project can be represented on a directed graph $G(V, E)$. Figure 1 shows an example uncertain MRCPSPP instance involving five non-dummy activities and a single renewable resource.

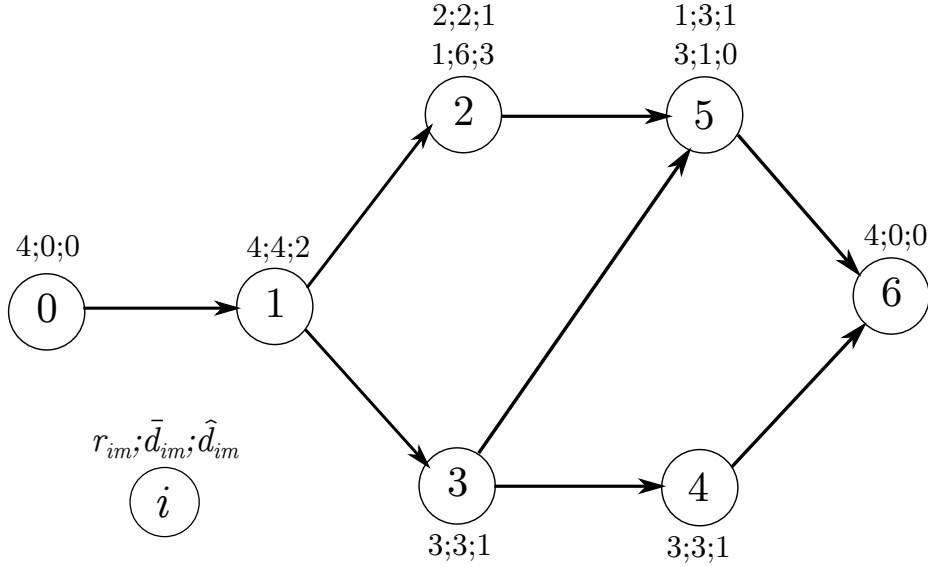


Figure 1: Example of an uncertain MRCPSPP instance involving five non-dummy activities and a single renewable resource with an availability of 4. Activities 2 and 5 each have two available processing modes, whilst the other activities have only a single mode.

For a given choice of processing modes for each activity \mathbf{m} , we assume that the activity

durations lie somewhere in a budgeted uncertainty set of the form

$$\mathcal{U}_{\mathbf{m}}(\Gamma) = \left\{ \mathbf{d} \in \mathbb{R}_+^{|V|} : d_{im_i} = \bar{d}_{im_i} + \xi_i \hat{d}_{im_i}, 0 \leq \xi_i \leq 1, \forall i \in V, \sum_{i \in V} \xi_i \leq \Gamma \right\}.$$

Introduced by [Bertsimas and Sim \(2004\)](#), the motivation of the budgeted uncertainty set is to control the pessimism of the solution by introducing a robustness parameter Γ to limit the number of jobs that can simultaneously achieve their worst-case durations. Observe that when $\Gamma = 0$, each activity takes its nominal duration and the resulting problem is the deterministic MRCPSP. At the other extreme, when $\Gamma = n$, every activity takes its worst-case duration, in which case the uncertainty set becomes equivalent to an interval uncertainty set. In this case the problem can again be solved as a deterministic MRCPSP instance considering only worst-case durations.

For a given selection of activity modes \mathbf{m} , a forbidden set is defined to be any subset of activities $F_{\mathbf{m}} \subseteq V$ that are not precedence-related, such that $\sum_{i \in F_{\mathbf{m}}} r_{im_i k} > R_k$ for at least one resource $k \in K$. That is, a forbidden set is a collection of activities that cannot be executed in parallel only because of resource limitations.

Applying the main representation theorem of [Bartusch et al. \(1988\)](#), given a particular choice of activity modes \mathbf{m} , a solution to the MRCPSP can be defined by a set of additional precedences $X_{\mathbf{m}} \subseteq V^2 \setminus E$ such that the extended precedence network $G(V, E \cup X_{\mathbf{m}})$ is acyclic and contains no forbidden sets. Such an extension to the project network is referred to as a *sufficient selection*.

Hence the two-stage robust MRCPSP under budgeted uncertainty, which we refer to as the robust MRCPSP, can be written as

$$\min_{\mathbf{m} \in \mathcal{M}, X_{\mathbf{m}} \in \mathcal{X}_{\mathbf{m}}} \max_{\mathbf{d} \in \mathcal{U}_{\mathbf{m}}(\Gamma)} \min_{\mathbf{S}} S_{n+1} \quad (1)$$

$$S_0 = 0 \quad (2)$$

$$S_j - S_i \geq d_{im_i} \quad \forall (i, j) \in E \cup X_{\mathbf{m}} \quad (3)$$

$$S_i \geq 0 \quad \forall i \in V. \quad (4)$$

This problem aims to select activity modes and a corresponding sufficient selection in order to minimise the worst-case project makespan.

3 A compact reformulation

We propose solving this problem using an extended version of the compact reformulation from [Bold and Goerigk \(2021\)](#). The development of this formulation for the RCPSP involved a reformulation of the adversarial sub-problem of determining the worst-case distribution of Γ delays among the project activities. The reformulated sub-problem took the form of a longest-path problem through an augmented project network.

The derivation for the equivalent compact reformulation for the MRCPSP is more or less identical to the one presented in that paper for the RCPSP. We therefore omit its derivation here, and instead just present the resulting model.

$$\min S_{n+1,\Gamma} \quad (5)$$

$$\text{s.t. } S_{00} = 0 \quad (6)$$

$$S_{j\gamma} - S_{i\gamma} \geq \bar{d}_{im}x_{im} - N(1 - y_{ij}) \quad \forall (i, j) \in V^2, \forall m \in M_i, \gamma = 0, \dots, \Gamma \quad (7)$$

$$S_{j,\gamma+1} - S_{i\gamma} \geq (\bar{d}_{im} + \hat{d}_{im})x_{im} - N(1 - y_{ij}) \quad \forall (i, j) \in V^2, m \in M_i, \gamma = 0, \dots, \Gamma - 1 \quad (8)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E \cup \{(n+1, n+1)\} \quad (9)$$

$$f_{ijk} \leq P_{ijk}y_{ij} \quad \forall (i, j) \in V^2, i \neq n+1, j \neq 0, \forall k \in K \quad (10)$$

$$\sum_{i \in V \setminus \{n+1\}} f_{ijk} = \sum_{m \in M_j} r_{jmk}x_{jm} \quad \forall j \in V \setminus \{0\}, \forall k \in K \quad (11)$$

$$\sum_{j \in V \setminus \{0\}} f_{ijk} = \sum_{m \in M_i} r_{imk}x_{im} \quad \forall i \in V \setminus \{n+1\}, \forall k \in K \quad (12)$$

$$\sum_{m \in M_i} x_{im} = 1 \quad \forall i \in V \quad (13)$$

$$\sum_{m \in M_i} r_{imk}^\nu x_{im} \leq R_k^\nu \quad \forall i \in V, k \in K^\nu \quad (14)$$

$$S_{i\gamma} \geq 0 \quad \forall i \in V, \gamma \in 0, \dots, \Gamma \quad (15)$$

$$f_{ijk} \geq 0 \quad \forall (i, j) \in V^2, \forall k \in K \quad (16)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in V^2 \quad (17)$$

$$x_{im} \in \{0, 1\} \quad \forall i \in V, m \in M_i, \quad (18)$$

where $N = \sum_{i \in V} \max_{m \in M_i} (\bar{d}_{im} + \hat{d}_{im})$ is an upper bound on the minimum makespan, and $P_{ijk} = \min\{\max_{m \in M_i} r_{imk}, \max_{m \in M_j} r_{jmk}\}$ is the maximum possible flow of resource k from i into j .

This formulation can be solved directly by a standard mathematical optimisation solver such as Gurobi or CPLEX.

4 A Benders' decomposition approach

In this section we outline a Benders' decomposition approach to solving the robust MRCPSPP first presented in [Balouka and Cohen \(2021\)](#). As mentioned previously, this approach is based on the approach used to solve the robust RCPSP in [Bruni et al. \(2017\)](#).

The main idea behind this approach is to decompose the full problem into a two smaller problems: 1. a master problem that determines activity processing modes and resolves resource conflicts, and 2. a subproblem that takes the solution from the master problem and evaluates it by finding the worst-case makespan for the resulting network by solving a longest path problem. The solution to the master problem forms a lower bound to the optimal objective value of the original problem, whilst the solution to the subproblem provides an upper bound. These two problems are solved iteratively, with

optimality cuts being added to the master problem each iteration based on the solution to the subproblem until the lower and upper bounds meet .

4.1 The master problem

The role of the master problem is to determine a choice of activity processing modes \mathbf{m} , as well as a sufficient selection $X_{\mathbf{m}}$ to resolve any resulting resource conflicts. The processing mode for each activity is determined by binary variables x_{im} . The y_{ij} variables define a complete set of precedences between the project activities, either due to the original project precedence constraints, or due to the resolution of resource conflicts, i.e. $\{(i, j), i, j \in V : y_{ij} = 1\} = T(E \cup X_{\mathbf{m}})$, where $T(E \cup X_{\mathbf{m}})$ is the transitive closure of the extended project precedence network. Resource flow variables f_{ijk} track the amount of renewable resource k that is transferred from activity i to activity j upon its completion. Finally, s_i and ϕ_{ij} are additional variables used to define two relaxations of the subproblem to strengthen the master problem formulation. Using these variables, the master problem can be formulated as follows:

$$\min \eta \tag{19}$$

$$y_{ij} = 1 \quad \forall (i, j) \in E \tag{20}$$

$$y_{ij} + y_{ji} \leq 1 \quad \forall i, j \in V, i < j \tag{21}$$

$$y_{ij} = y_{ip} + y_{pj} - 1 \quad \forall i, j, p \in V, i \neq j \neq p \tag{22}$$

$$\begin{aligned} f_{ijk} \leq & \min(r_{im_k}, r_{jm_k}) \cdot y_{ij} + (1 - x_{i, m_i})(\bar{r}_{ijk} - \min(r_{im_k}, r_{jm_k})) \\ & + (1 - x_{j, m_j})(\bar{r}_{ijk} - \min(r_{im_k}, r_{jm_k})) \\ & \forall i \in V \setminus \{n+1\}, j \in V \setminus \{0\}, i \neq j, m_i \in M_i, m_j \in M_j, k \in K \end{aligned} \tag{23}$$

$$\sum_{i \in V \setminus \{j, n+1\}} f_{ijk} = \sum_{m \in M_j} r_{jm_k} x_{jm} \quad \forall j \in V \setminus \{0\}, k \in K \tag{24}$$

$$\sum_{j \in V \setminus \{0, i\}} f_{ijk} = \sum_{m \in M_i} r_{im_k} x_{im} \quad \forall i \in V \setminus \{n+1\}, k \in K \tag{25}$$

$$\sum_{m \in M_i} x_{im} = 1 \quad \forall i \in V \tag{26}$$

$$\sum_{m \in M_i} r_{im_k}^{\nu} x_{im} \leq R_k^{\nu} \quad \forall i \in V, k \in K^{\nu} \tag{27}$$

$$\eta \geq \lambda(x, y, M^{*\ell}) \quad \forall \ell = 0, \dots, t-1 \tag{28}$$

$$s_j - s_i \geq \bar{d}_{im} - N(1 - y_{ij}) - N(1 - x_{im}) \quad \forall i, j \in V, i \neq j, m \in M_i \tag{29}$$

$$\eta \geq s_{n+1} \tag{30}$$

$$\eta \geq \sum_{i \in V} \sum_{j \in V} \left(\min_{m \in M_i} \bar{d}_{im} \right) \phi_{ij} \tag{31}$$

$$\sum_{j \in V} \phi_{ji} = \sum_{j \in V} \phi_{ij} \quad \forall i \in V \setminus \{0, n+1\} \tag{32}$$

$$\sum_{j \in V} \phi_{0j} = 1 \quad (33)$$

$$\sum_{j \in V} \phi_{j,n+1} = 1 \quad (34)$$

$$\phi_{ij} \leq y_{ij} \quad \forall i, j \in V, i \neq j \quad (35)$$

$$y_{ij} \in \{0, 1\} \quad \forall i, j \in V, i \neq j \quad (36)$$

$$x_{im} \in \{0, 1\} \quad \forall i \in V, m \in M_i \quad (37)$$

$$f_{ijk} \geq 0 \quad \forall i \in V \setminus \{n+1\}, j \in V \setminus \{0\}, i \neq j, k \in K \quad (38)$$

$$s_i \geq 0 \quad \forall i \in V \quad (39)$$

$$\phi_{ij} \geq 0 \quad \forall i, j \in V \quad (40)$$

where $\bar{r}_{ijk} = \max(\max_{m \in M_i} r_{imk}, \max_{m \in M_j} r_{jmk})$ is the maximum usage of resource $k \in K$ by either activity i or j , and as before, $N = \sum_{i \in V} \max_{m \in M_i} (\bar{d}_{im} + \hat{d}_{im})$ is an upper bound on the minimum makespan.

The solution to the optimal objective value of the master problem (19)-(40), denoted by η^* , forms a lower bound to the original robust MRCPSP problem. Note that for the first iteration, the solution to the master problem corresponds to an optimal solution to the MRCPSP with nominal activity durations.

The original project precedence relations are captured by constraint (20). Constraints (21) and (22) ensure that the extended network is transitive and acyclic. Constraints (23) prevents the flow of resource from i to j from exceeding either the amount that becomes available from the completion of i , or the amount that is required by j , accounting for their selected modes. Constraints (24) and (25) ensure that each activity receives the amount of each renewable resource that it requires. Constraints (26) allows only one processing mode to be selected for each activity. Non-renewable resource constraints are defined by (27). Constraints (28) define the optimality cuts generated at each of the previous iterations. These cuts are explained in greater detail in Section 4.3.

In line with Bruni et al. (2017) and Balouka and Cohen (2021), two relaxations of the subproblem have been included in the master problem in order to strengthen it. The first relaxation is implemented through constraints (29) and (30). These simply enforce the extended precedence relationships using nominal activity durations for each processing mode. The second relaxation is captured by constraints (31)-(35), which use flow variables ϕ_{ij} to form a longest-path problem through the extended project network using minimal activity durations.

4.2 The subproblem

The subproblem at iteration t considers the worst-case makespan for the mode choices and sufficient selection from the master problem at iteration t , denoted by \mathbf{m}^{*t} and $X_{\mathbf{m}^{*t}}$ respectively. This is done by solving an uncertain longest-path problem through the extended project network, where activity durations lie in the budgeted uncertainty

set $\mathcal{U}_{\mathbf{m}^{*t}}(\Gamma)$. This problem can be formulated as

$$V^{*t} = \max \sum_{(i,j) \in T(E \cup X_{\mathbf{m}^{*t}})} \bar{d}_{im_i^{*t}} \alpha_{ij} + \hat{d}_{im_i^{*t}} w_{ij} \quad (41)$$

$$\sum_{(i,n+1) \in T(E \cup X_{\mathbf{m}^{*t}})} \alpha_{i,n+1} = 1 \quad (42)$$

$$\sum_{(0,i) \in T(E \cup X_{\mathbf{m}^{*t}})} \alpha_{0,i} = 1 \quad (43)$$

$$\sum_{(i,j) \in T(E \cup X_{\mathbf{m}^{*t}})} \alpha_{ij} - \sum_{(j,i) \in T(E \cup X_{\mathbf{m}^{*t}})} \alpha_{ji} = 0 \quad \forall i \in V \setminus \{0, n+1\} \quad (44)$$

$$w_{ij} \leq \xi_i \quad \forall (i,j) \in T(E \cup X_{\mathbf{m}^{*t}}) \quad (45)$$

$$w_{ij} \leq \alpha_{ij} \quad \forall (i,j) \in T(E \cup X_{\mathbf{m}^{*t}}) \quad (46)$$

$$\sum_{i \in V} \xi_i \leq \Gamma \quad (47)$$

$$\alpha_{ij} \in \{0, 1\} \quad \forall (i,j) \in T(E \cup X_{\mathbf{m}^{*t}}) \quad (48)$$

$$w_{ij} \geq 0 \quad \forall (i,j) \in T(E \cup X_{\mathbf{m}^{*t}}) \quad (49)$$

$$0 \leq \xi_i \leq 1 \quad \forall i \in V \quad (50)$$

Variables α_{ij} define the longest path through the network. This path, which we denote as $\pi^{*t} = \{(i,j), i,j \in V : \alpha_{ij} = 1\}$ has length V^{*t} . Variables ξ_i are the activity delay variables, whilst w_{ij} are used to linearise the formulation.

4.3 Optimality cuts

Following the solution to the subproblem, a valid cut is generated and added to the master problem for the next iteration. This cut simply forces an alternative solution in the master problem if it is to achieve a better objective value in the next iteration.

Given the current best lower bound for the original problem LB , the mode selection from the master problem \mathbf{m}^{*t} , and the optimal objective value of the subproblem V^{*t} , the cut at iteration t can be written as

$$\eta \geq (V^{*t} - LB) \cdot \sum_{(i,j) \in \pi^{*t}} \left(1/3(y_{ij} + x_{i,m_i^{*t}} + x_{j,m_j^{*t}}) - (3 - y_{ij} - x_{i,m_i^{*t}} - x_{j,m_j^{*t}}) \right) - (V^{*t} - LB) \cdot (|\pi^{*t}| - 1) + LB, \quad (51)$$

Balouka and Cohen (2021) show that this constraint is a valid optimality cut for the problem, and that the number of these cuts that need to be added to the master problem before finding an optimal solution is finite.

Before presenting a small example of the Benders' decomposition approach in the following subsection, we show an overview of the solution method in /ref.

4.4 Example

5 Computational experiments and results

In this section we compare results from solving the compact reformulation with results from solving the Benders’ decomposition approach. A complete set of the raw results used to generate the tables and plots presented here, in addition to the source code used to implement these experiments, can be found at <https://github.com/boldm1/robust-mrcpsp>.

The instances used in this computational study have been created using the deterministic $j10$, and $j20$ MRCPSP instances from the PSPLIB (Kolisch and Sprecher (1997), <https://www.om-db.wi.tum.de/psplib/>). The $j10$ set contains a total of 536 instances each involving 10 activities, and the $j20$ set contains a total of 554 instances each involving 20 activities. We introduce uncertainty into these deterministic instances by setting the maximum deviation of the duration for each activity to be $\hat{d}_{im} = \lfloor 0.7 \times \bar{d}_{im} \rfloor$ for each mode $m \in M_i$. We then solve each instance for a range of robustness levels Γ . In particular we solve the $j10$ instances for $\Gamma \in \{0, 3, 5, 7\}$ and the $j20$ instances for $\Gamma \in \{0, 5, 10, 15\}$.

Both the compact reformulation and Benders’ decomposition approach were implemented in Python 3.9.2 and solved using Gurobi 9.0.1 running on a single core of a 2.30 GHz Intel Xeon CPU. A time limit of 2 hours per instance was imposed on each of the solution methods.

Table 1 reports the percentage of instances solved to optimality ($\%sol.$), the average optimality gap (in percent) over instances for which a feasible solution was found (gap), and the average solution time ($sol. \text{ time}$) across the two instance sets and different choices of Γ , for both the Benders’ decomposition approach and the compact reformulation. Note that if no optimal solution was found within the time limit by one of the solution methods, the solution time was recorded as the time limit value of 7200 seconds. Additionally, for the Benders’ approach, the average number of iterations ($it.$) and the average time per iteration ($it. \text{ time}$) are also reported. The average of these values across the different values for Γ are also provided for each instance set.

6 Conclusions

Acknowledgements

The authors are grateful for the support of the EPSRC-funded (EP/L015692/1) STOR-i Centre for Doctoral Training.

References

Balouka, N. and Cohen, I. (2021). A robust optimization approach for the multi-mode resource-constrained project scheduling problem. *European Journal of Operational*

	Γ	Benders'					Compact reformulation		
		%sol.	gap	its.	it. time	sol. time	%sol.	gap	sol. time
$j10$	0	99.6	0.00	1	73.6	100.4	100.00	0.00	1.8
	3	75.0	5.23	48	64.1	2065.6	100.00	0.00	5.0
	5	72.8	6.53	53	67.9	2234.5	100.00	0.00	4.6
	7	71.8	6.75	54	68.1	2278.4	100.00	0.00	6.5
		79.8	4.63	39.4	68.4	1669.7	100.00	0.00	4.47
$j20$	0	80.8	0.00	1	123.5	1484.7	89.4	1.92	941.9
	5	39.6	10.91	173	124.8	5181.7	88.4	1.91	1061.3
	10	36.8	12.31	178	126.6	5406.2	87.7	2.63	1138.4
	15	36.8	12.48	177	135.2	5415.8	86.0	2.96	1279.8
		48.5	8.93	132.5	127.5	4372.1	87.9	2.36	1105.4

Table 1: Comparison of the Benders' decomposition approach and the compact reformulation across the $j10$ and $j20$ instance sets and the different values of Γ .

Research, 291(2):457–470.

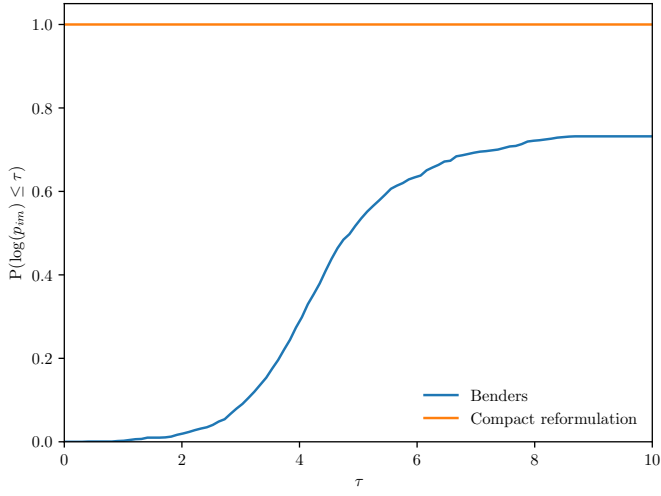
Bartusch, M., Möhring, R. H., and Radermacher, F. J. (1988). Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, 16(1):199–240.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.

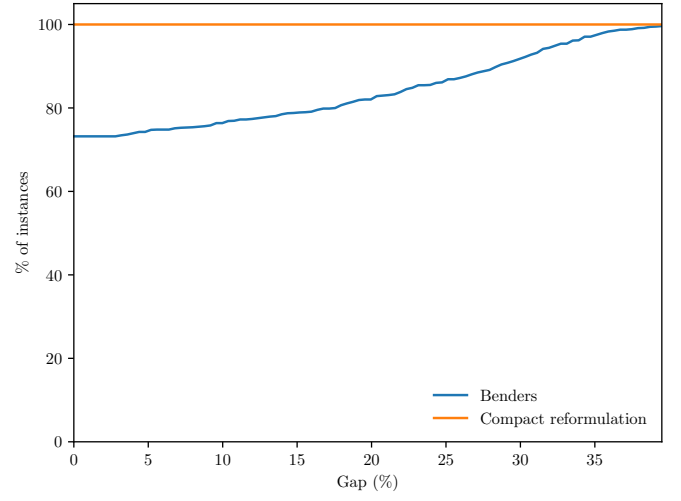
Bold, M. and Goerigk, M. (2021). A compact reformulation of the two-stage robust resource-constrained project scheduling problem. *Computers & Operations Research*, 130:105232.

Bruni, M. E., Pugliese, L. D. P., Beraldi, P., and Guerriero, F. (2017). An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations. *Omega*, 71:66–84.

Kolisch, R. and Sprecher, A. (1997). PSPLIB - A project scheduling problem library: OR Software - ORSEP Operations Research Software Exchange Program. *European Journal of Operational Research*, 96(1):205–216.

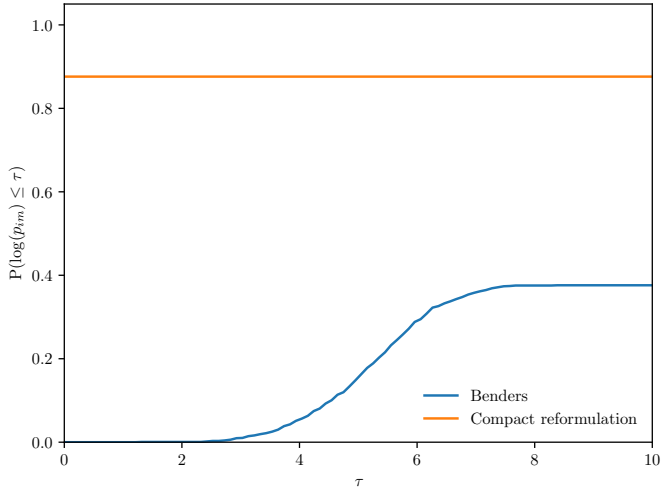


(a) Performance profile of relative solution times.

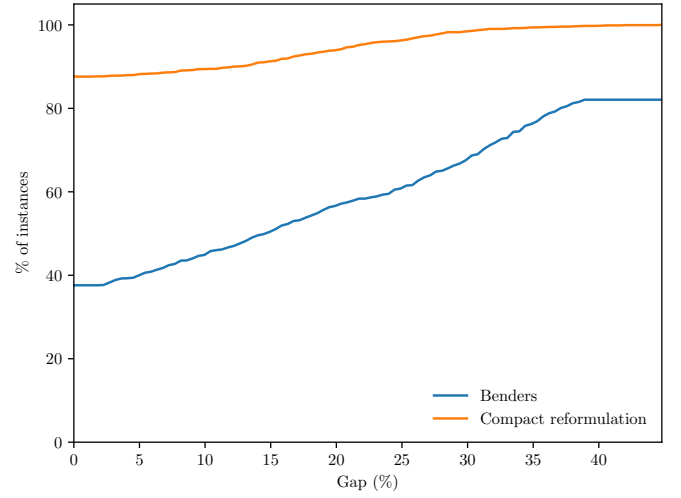


(b) Percentage of instances solved to within given gap of optimality within the two-hour time limit.

Figure 2: Comparison of Benders' approach and compact reformulation over instances in the j10 set.



(a) Performance profile of relative solution times.



(b) Percentage of instances solved to within given gap of optimality within the two-hour time limit.

Figure 3: Comparison of Benders' approach and compact reformulation over instances in the j20 set.