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

Matthew Bold*[1] and Marc Goerigk[2]

[1]STOR-i Centre for Doctoral Training, Lancaster University, Lancaster, UK
[2]Network 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 inclusion of these 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.

In this paper we consider the MRCPSP under uncertain activity durations and model it using a two-stage adjustable robust optimisation framework. In this setting, a first-stage problem is solved to determine activity mode selections and make activity sequencing decisions to resolve resource conflicts. Following this, the actual activity durations are realised and a complete schedule is computed. The aim of the two-stage robust MRCPSP is to find a feasible first-stage solution (i.e. mode selection and sequencing decisions) in order to minimise the realised worse-case makespan, as computed in the second-stage. We refer to this problem as the robust MRCPSP.

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

In recent years this two-stage robust optimisation approach has been applied to the RCPSP. First to use this approach were Artigues et al. (2013), who present an iterative scenario-relaxation algorithm for this problem with the objective of minimising the worst-case absolute regret. Bruni et al. (2017) propose a Benders'-style decomposition approach for solving the robust RCPSP with the objective of minimising the worst-case project makespan. This work was extended in Bruni et al. (2018), which presents a computational study comparing an additional Benders' decomposition approach against a primal decomposition algorithm. Recently, Bold and Goerigk (2021) introduced a compact reformulation of the robust RCPSP and presented results which showed the superiority of that formulation over the iterative methods developed in those papers.

The application of the this two-stage robust optimisation approach for the MRCPSP is a more recently development. To the best of our knowledge, the only existing paper to consider this problem is Balouka and Cohen (2021), in which the Benders' decomposition approach used by Bruni et al. (2017) for the robust RCPSP has been extended for application to the MRCPSP. Analogous to that work, the aim of this paper is to apply the compact reformulation first developed for the RCPSP by Bold and Goerigk (2021) to the MRCPSP. Specifically, we present a computational comparison of our compact reformulation approach and a strengthened version of the Benders' decomposition approach.

2 Problem description

A project consists of a set of non-preemptive activities $V = \{0, 1, \ldots, n, n+1\}$, 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 $M_i = \{1, \ldots, |M_i|\}$. 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_ik} 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 renewable resource requirements, mode m_i of activity i also has a non-renewable resource requirement of r'_{im_ik} for each non-renewable resource type $k \in K'$, with each non-renewable resource having an overall availability of R'_k for the entire project horizon. Additionally, 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. These form a project network that can be represented using a directed graph $G(V, E)$. Figure 1 shows an example instance involving five non-dummy activities and a single renewable resource.

For a given choice of processing modes for each activity $\boldsymbol{m} = (m_1, \ldots, m_n)$, we assume that the activity durations lie somewhere in a budgeted uncertainty set of the form

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

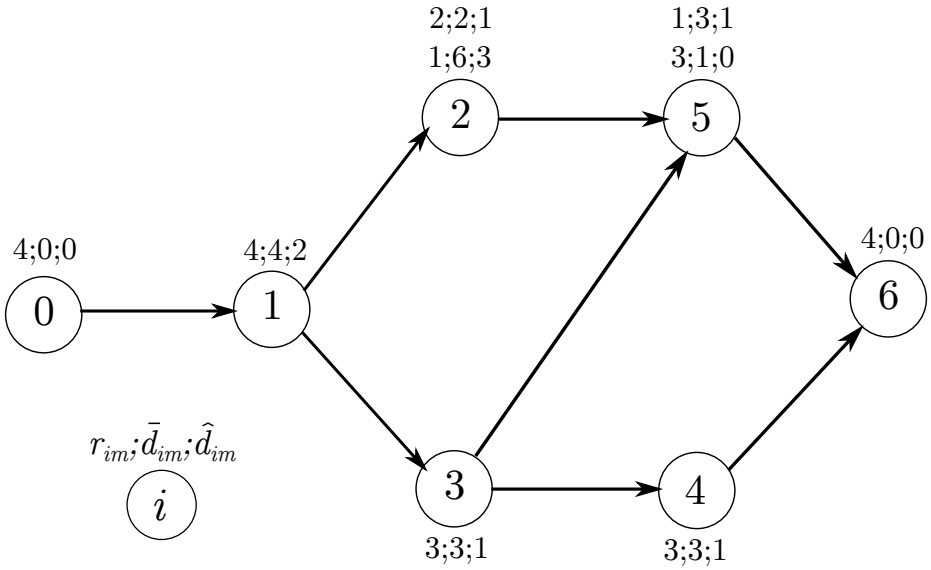


Figure 1: An example instance involving five non-dummy activities and a single renewable resource. Activities 2 and 5 each have two available processing modes, whilst the other activities have only a single mode.

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 set of activity modes $\boldsymbol{m}$, a forbidden set is defined to be any subset of activities $F_{\boldsymbol{m}} \subseteq V$ that are not precedence-related, such that $\sum_{i \in F_{\boldsymbol{m}}} r_{im_i k} > R_k$ 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), for a particular choice of activity modes $\boldsymbol{m}$, a solution to the MRCPSP can be defined by a set of additional precedences $X_{\boldsymbol{m}} \subseteq V^2 \backslash E$ such that the extended precedence network $G(V, E \cup X_{\boldsymbol{m}})$ 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

$$\min_{\boldsymbol{m} \in \mathcal{M},\, X_{\boldsymbol{m}} \in \mathcal{X}_{\boldsymbol{m}}} \max_{\boldsymbol{d} \in \mathcal{U}_{\boldsymbol{m}}(\Gamma)} \min S_{n+1} \tag{1}$$

$$S_0 = 0 \tag{2}$$
$$S_j - S_i \geq d_{im_i} \qquad \forall (i,j) \in E \cup X_{\boldsymbol{m}} \tag{3}$$
$$S_i \geq 0 \qquad \forall i \in V, \tag{4}$$

where $\mathcal{M} \subseteq \mathbb{N}^n$ represents the set of all possible combinations of activity processing mode selections, and $\mathcal{X}_{\boldsymbol{m}}$ is the set of all possible sufficient selections for the choice of processing modes given by $\boldsymbol{m}$. This problem aims to selects activity modes and a corresponding sufficient selection in order to minimise the worst-case project makespan.

3 A compact formulation

We propose solving this problem using an extended version of the compact formulation developed by Bold and Goerigk (2021) for solving the robust RCPSP. This formulation combines the first and second-stage problems into a single compact formulation and was shown to significantly outperform the strongest iterative decompostion-based methods for the robust RCPSP.

This formulation is constructed by recasting the adversarial subproblem of determining the worst-case activity durations for a given set of mode choices and sufficient selection as a longest path problem on an augmented project network. This augmented project network is formed of $\Gamma+1$ connected copies of the original project network, with each level being used to account for a delay to a single activity in the project. Having reformulated the adversarial subproblem as a longest-path problem, it can be dualised an inserted into a formulation for the full problem, to arrive at a complete compact formulation for the first-stage problem. Given that the derivation of this formulation for the MRCPSP is more or less identical to the derivation for the RCPSP, we omit it here and refer the reader to Bold and Goerigk (2021).

The resulting compact formulation for the two-stage adjustable robust MRCPSP can be written as

$$\min \; S_{n+1,\Gamma} \tag{5}$$
$$\text{s.t.} \; S_{00} = 0 \tag{6}$$
$$S_{j\gamma} - S_{i\gamma} \geq \bar{d}_{im} x_{im} - N(1 - y_{ij}) \qquad \forall (i,j) \in V^2, \forall m \in M_i, \gamma = 0, \ldots, \Gamma \tag{7}$$
$$S_{j,\gamma+1} - S_{i\gamma} \geq (\bar{d}_{im} + \hat{d}_{im}) x_{im} - N(1 - y_{ij}) \qquad \forall (i,j) \in V^2, m \in M_i, \gamma = 0, \ldots, \Gamma - 1 \tag{8}$$
$$y_{ij} = 1 \qquad \forall (i,j) \in E \cup \{(n+1, n+1)\} \tag{9}$$
$$y_{ij} + y_{ji} \leq 1 \qquad \forall i,j \in V, i < j \tag{10}$$
$$y_{ij} = y_{ip} + y_{pj} - 1 \qquad \forall i,j,p \in V, i \neq j \neq p \tag{11}$$
$$f_{ijk} \leq P_{ijk} y_{ij} \qquad \forall (i,j) \in V^2, i \neq n+1, j \neq 0, \forall k \in K \tag{12}$$
$$\sum_{i \in V \setminus \{n+1\}} f_{ijk} = \sum_{m \in M_j} r_{jmk} x_{jm} \qquad \forall j \in V \setminus \{0\}, \forall k \in K \tag{13}$$

$$\sum_{j \in V \setminus \{0\}} f_{ijk} = \sum_{m \in M_i} r_{imk} x_{im} \qquad \forall i \in V \setminus \{n+1\}, \forall k \in K \tag{14}$$

$$\sum_{m \in M_i} x_{im} = 1 \qquad \forall i \in V \tag{15}$$

$$\sum_{m \in M_i} r'_{imk} x_{im} \leq R'_k \qquad \forall i \in V, k \in K' \tag{16}$$

$$S_{i\gamma} \geq 0 \qquad \forall i \in V, \gamma \in 0, \ldots, \Gamma \tag{17}$$

$$f_{ijk} \geq 0 \qquad \forall (i,j) \in V^2, \forall k \in K \tag{18}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in V^2 \tag{19}$$

$$x_{im} \in \{0,1\} \qquad \forall i \in V, m \in M_i, \tag{20}$$

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.

Constraints (6)-(8) are the dual constraints corresponding to the longest-path variables in the adversarial subproblem, and serve to ensure that the activity start times respect the project precedences as well as any delays to activity durations. Constraints (9) capture the original project precedence relationships. Although not required for the correctness of the model, constraints (10) and (11) are additional transitivity constraints that have been included because they were shown in Bold and Goerigk (2021) to provide significant improvements to the computational performance of the model. Constraints (12)-(14) are resource flow constraints which ensure that the transfer of resources between activities follows precedence constraints and that resources are conserved as they flow into and out of each activity in the network. Constraints (15) allows only one processing mode to be selected for each activity. Finally, non-renewable resource constraints are enforced by (16).

With polynomially many constraints this formulation can be implemented straightforwardly using standard mathematical optimisation software such as Gurobi or CPLEX.

4 A Benders' decomposition approach

In this section we outline a Benders' decomposition approach to solving the robust MRCPSP 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 $\boldsymbol{m}$, as well as a sufficient selection $X_{\boldsymbol{m}}$ 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. $\{(i,j), i,j \in V : y_{ij} = 1\} = T(E \cup X_{\boldsymbol{m}})$, where $T(E \cup X_{\boldsymbol{m}})$ 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:

$$\min \eta \tag{21}$$

$$\text{s.t. } \eta \geq S_{n+1} \tag{22}$$

$$S_0 = 0 \tag{23}$$

$$S_j - S_i \geq \bar{d}_{im} x_{im} - N(1 - y_{ij}) \qquad \forall (i,j) \in V^2, \forall m \in M_i \tag{24}$$

$$(9)\text{-}(16), (18)\text{-}(20) \tag{25}$$

$$S_i \geq 0 \qquad \forall i \in V \tag{26}$$

The solution to he optimal objective value of the master problem (21)-(26), 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 (**??**). Constraints (**??**) and (**??**) ensure that the extended network is transitive and acyclic. Constraints (**??**) 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 (**??**) and (**??**) ensure that each activity receives the amount of each renewable resource that it requires. Constraints (**??**) allows only one processing mode to be selected for each activity. Non-renewable resource constraints are defined by (**??**). Constraints (**??**) 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 (**??**) and (**??**). These simply enforce the extended precedence relationships using nominal activity durations for each processing mode. The second relaxation is captured by constraints (**??**)-(**??**), 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 $\boldsymbol{m}^{*t}$ and $X_{\boldsymbol{m}^{*t}}$ 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 $\mathcal{U}_{\boldsymbol{m}^{*t}}(\Gamma)$. This problem can be formulated as

$$V^{*t} = \max \sum_{(i,j)\in T(E\cup X_{\boldsymbol{m}^{*t}})} \bar{d}_{im_i^{*t}}\alpha_{ij} + \hat{d}_{im_i^{*t}}w_{ij} \tag{27}$$

$$\sum_{(i,n+1)\in T(E\cup X_{\boldsymbol{m}^{*t}})} \alpha_{i,n+1} = 1 \tag{28}$$

$$\sum_{(0,i)\in T(E\cup X_{\boldsymbol{m}^{*t}})} \alpha_{0,i} = 1 \tag{29}$$

$$\sum_{(i,j)\in T(E\cup X_{\boldsymbol{m}^{*t}})} \alpha_{ij} - \sum_{(j,i)\in T(E\cup X_{\boldsymbol{m}^{*t}})} \alpha_{ji} = 0 \qquad \forall i \in V \setminus \{0, n+1\} \tag{30}$$

$$w_{ij} \leq \xi_i \qquad \forall (i,j) \in T(E \cup X_{\boldsymbol{m}^{*t}}) \tag{31}$$

$$w_{ij} \leq \alpha_{ij} \qquad \forall (i,j) \in T(E \cup X_{\boldsymbol{m}^{*t}}) \tag{32}$$

$$\sum_{i\in V} \xi_i \leq \Gamma \tag{33}$$

$$\alpha_{ij} \in \{0,1\} \qquad \forall (i,j) \in T(E \cup X_{\boldsymbol{m}^{*t}}) \tag{34}$$

$$w_{ij} \geq 0 \qquad \forall (i,j) \in T(E \cup X_{\boldsymbol{m}^{*t}}) \tag{35}$$

$$0 \leq \xi_i \leq 1 \qquad \forall i \in V \tag{36}$$

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 $\boldsymbol{m}^{*t}$, and the optimal objective value of the subproblem V^{*t}, the cut at iteration t can be written as

$$\eta \geq (V^{*t} - LB) \cdot \sum_{(i,j)\in\pi^{*t}} \Big(1/3(y_{ij} + x_{i,m_i^{*t}} + x_{j,m_j^{*t}}) - (3 - y_{ij} - x_{i,m_i^{*t}} - x_{j,m_j^{*t}}) \Big) - (V^{*t} - LB) \cdot (|\pi^{*t}| - 1) + LB, \tag{37}$$

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.

Algorithm 1 Benders' decomposition algorithm

Initialise: Set $LB = -\infty$, $UB = +\infty$ and $t = 1$.
while $UB > LB$ **do**
    Solve master problem (21)-(26)
        Get objective value η^{*t}, processing modes $\boldsymbol{m^{*t}}$ and precedences $\boldsymbol{y_{m^{*t}}}$.
        If $\eta^{*t} > LB$, update $LB \leftarrow \eta^{*t}$
    Solve subproblem (27)-(36)
        Get objective value V^{*t} and longest path π^{*t}
        If $V^{*t} < UB$, update $UB \leftarrow V^{*t}$
    Add cut (37) to master problem
    Update $t \leftarrow t + 1$
end while
return UB

t	LB	UB	η^{*t}	V^{*t}	$\boldsymbol{m}^{*t}$	π^{*t}
1	11	16	11	16	1,2,1,1,2	0,1,2,5,6
2	12	15	12	15	1,1,1,1,1	0,1,3,2,4,6
3	12	15	12	15	1,1,1,1,1	0,1,2,3,4,6
4	13	15	13	18	1,2,1,1,1	0,1,2,5,6
5	13	15	13	16	1,1,1,2,1	0,1,3,4,2,5,6
6	15	15	15	-	1,1,1,1,1	-

Table 1: Iterations of the Benders' decomposition approach to solve example instance in Figure 1.

4.4 Example

Here we solve the example shown in Figure 1 with $R_1 = 4$ and $\Gamma = 2$ to illustrate the implementation of the Benders' decomposition approach which is summarised in Algorithm 1. This problem is solved in 6 iterations, and the solution information from the master and subproblem at each of these iterations is shown in Table 1. Note that in the final iteration, the algorithm can terminate after solving the master problem and updating the UB since $LB = UB$, i.e. there is no need to solve the subproblem in the final iteration.

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.

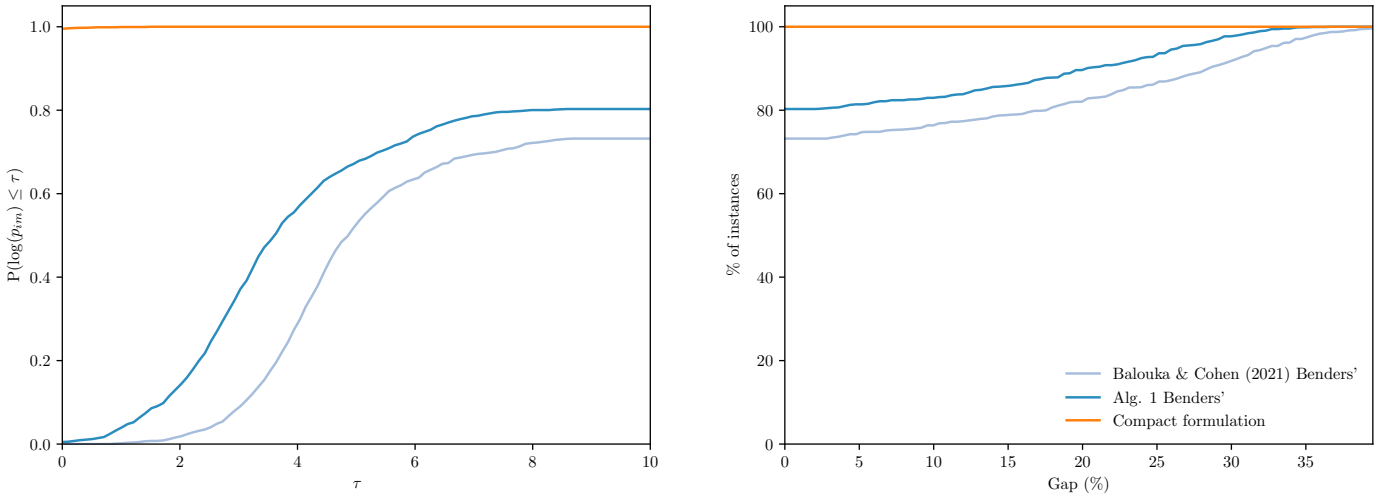
The instances used in this computational study have been created using the deterministic $j10$, and $j20$ MRCPSP instances from the PSPLIB (Kolisch and Sprecher (1997), https://www.om-db.wi.tum.de/psplib/). The $j10$ set contains a total of 536 instances each involving 10 activities, and the $j20$ set 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 2 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. 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. time*) are also reported. The average of these values across the different values for Γ are also provided for each instance set.

		Benders'					Compact formulation		
	Γ	*%sol.*	*gap*	*its.*	*it. time*	*sol. time*	*%sol.*	*gap*	*sol. time*
$j10$	0	100.0	0.00	1	0.96	1.1	100.0	0.00	1.8
	3	81.7	3.13	67	3.76	1551.4	100.0	0.00	5.0
	5	79.7	4.20	76	3.89	1688.4	100.0	0.00	4.6
	7	79.5	4.52	78	3.98	1701.1	100.0	0.00	6.5
		85.2	2.96	56	3.15	1235.5	100.0	0.00	4.47
$j20$	0	89.7	0.00	1	176.62	901.3	89.4	1.92	941.9
	5	51.3	10.04	228	135.66	4153.6	88.4	1.91	1061.3
	10	50.7	10.81	228	140.37	4230.4	87.7	2.63	1138.4
	15	49.8	11.05	222	148.72	4287.1	86.0	2.96	1279.8
		60.4	7.98	170	150.34	3393.1	87.9	2.36	1105.4

Table 2: Comparison of the Benders' decomposition approach and the compact reformulation across the $j10$ and $j20$ instance sets and the different values of Γ.



(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.

6 Conclusions

Acknowledgements

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

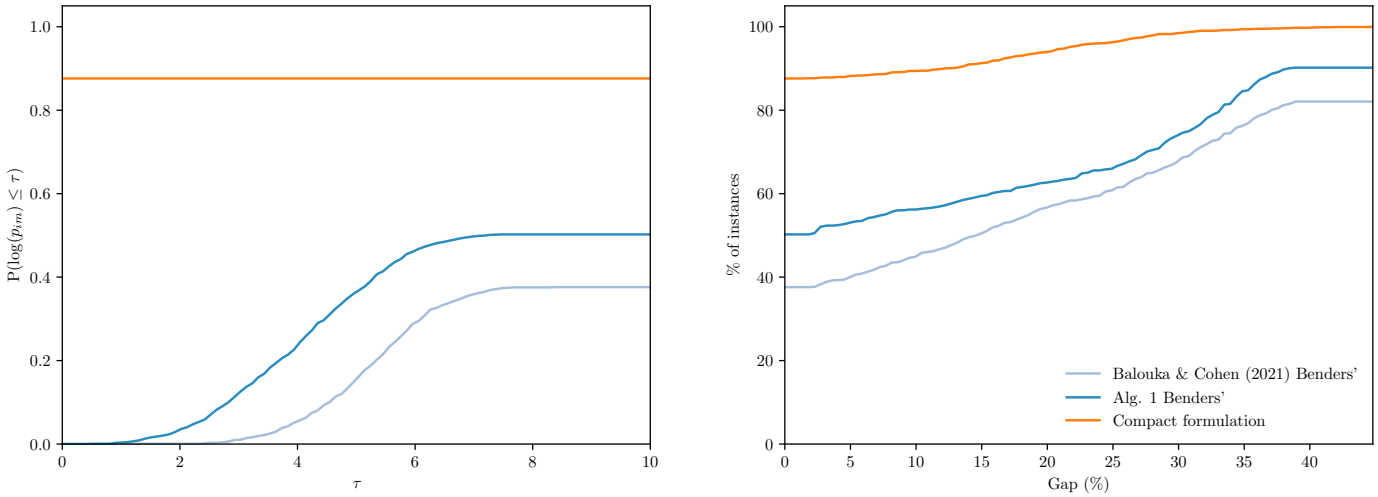
References

Artigues, C., Leus, R., and Talla Nobibon, F. (2013). Robust optimization for resource-constrained project scheduling with uncertain activity durations. *Flexible Services and Manufacturing Journal*, 25(1):175–205.

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



(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.

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.

Bruni, M. E., Pugliese, L. D. P., Beraldi, P., and Guerriero, F. (2018). A computational study of exact approaches for the adjustable robust resource-constrained project scheduling problem. *Computers & Operations Research*, 99:178–190.

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.