



**HAL**  
open science

## **A branch-and-check approach for a wind turbine maintenance scheduling problem**

Aurélien Froger, Michel Gendreau, Jorge E. Mendoza, Eric Pinson,  
Louis-Martin Rousseau

► **To cite this version:**

Aurélien Froger, Michel Gendreau, Jorge E. Mendoza, Eric Pinson, Louis-Martin Rousseau. A branch-and-check approach for a wind turbine maintenance scheduling problem. *Computers and Operations Research*, 2017, 88, pp.117-136. 10.1016/j.cor.2017.07.001 . hal-01617150

**HAL Id: hal-01617150**

**<https://hal.science/hal-01617150>**

Submitted on 26 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A branch-and-check approach for a wind turbine maintenance scheduling problem

Aurélien Froger<sup>1</sup>    Michel Gendreau<sup>2</sup>    Jorge E. Mendoza<sup>3</sup>    Eric Pinson<sup>1</sup>  
Louis-Martin Rousseau<sup>3</sup>

<sup>1</sup> UCO-IMA, Université Bretagne Loire, LARIS EA7315, Angers, France

<sup>2</sup> Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Montreal, Canada  
Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montreal, Canada

<sup>3</sup> Université François-Rabelais de Tours, CNRS, LI EA 6300, OC ERL CNRS 6305, Tours, France

## Abstract

In this paper we deal with a maintenance scheduling problem arising in the onshore wind power industry. We consider a short-term horizon and a multi-skilled workforce. The goal is to schedule maintenance operations to maximize electricity production while taking into account forecast wind-speed values, multiple task execution modes, and daily restrictions on the routes of the technicians. We first introduce two integer linear programming formulations of the problem. Then, building on one of our models, we propose a branch-and-check (B&C) approach that uses both generic Benders cuts and cuts specially crafted for our problem. We report experiments on a 160-instance testbed. For 80% of the instances, our exact approach finds an optimal solution in a reasonable computational time. The remaining instances reach the three-hour time limit, and our B&C gives solutions with average gaps of 1.7% with respect to the upper bounds. The results suggest that our method significantly outperforms commercial solvers running our integer linear programming models.

This is a post-peer-review, pre-copyedit version of an article published in *Computers & Operations Research*. The final authenticated version is available online at: <https://doi.org/10.1016/j.cor.2017.07.001>

### Citation for published version:

Froger A., Gendreau M., Mendoza J.E., Pinson E., and Rousseau L-M. (2017). A branch-and-check approach to solve a wind turbine maintenance scheduling problem. *Computers & Operations Research*, 88:117-136, doi:10.1016/j.cor.2017.07.001.

©2017. This manuscript version is made available under the CC-BY-NC-ND 4.0 license  
(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

# Introduction

The energy sector is being challenged to produce low-carbon electricity, and so the share of renewables has significantly increased in recent years. Boosted by the need to mitigate climate change and adaptation support (e.g., tax incentives, the Paris climate change agreement) as well as the constantly decreasing cost of turbines, wind energy is currently the world’s fastest-growing source of electricity (63 GW of new capacity in 2015). It accounts for around 3.3% of world electricity production<sup>1</sup>.

Although the availability factor of wind turbines exceeds 95%<sup>2</sup>, their capacity factor<sup>3</sup> is usually around 30–40% as a result of the variability of wind and of design decisions (for a fixed wind speed, the larger the blades the more electricity the turbine can produce). The impact of operational decisions is also non-negligible. As the wind industry grows, the reliability and profitability of wind farms naturally becomes one of the priorities of the sector. In this context, the efficient scheduling of wind turbine maintenance operations is essential to prevent unnecessary downtime and excessive operational costs.

Maintenance planning and scheduling has been widely studied in the electricity industry, particularly for thermal power plants; see Froger et al. (2016) for a comprehensive review. Many of the existing methods are not applicable to the wind power industry, where the impact of shutting down equipment depends on an uncontrollable factor: the weather. Studies of maintenance optimization in the wind energy sector primarily focus on reliability-centered maintenance (RCM). RCM is a methodology incorporating reactive, preventive, and condition-based maintenance decisions. These decisions, essentially taken under financial considerations, are mostly based on monitoring and on the use of failure prediction models. Studies usually focus on a single turbine or a single wind farm and explore why failures happen, what should be done when they occur, and how they can be predicted or prevented. See Ding et al. (2013) for a survey. The results of these studies can help to define the preventive maintenance operations that need to be performed in the short term. However, they may provide only rough estimates of turbine downtime and lost production. Indeed, it is difficult to produce a maintenance plan in which all the maintenance operations are scheduled during periods where the wind speed is below  $3.5 \text{ m.s}^{-1}$ , which is too low to produce electricity. This is because human resources is a major bottleneck.

To the best of our knowledge, only a few studies have addressed turbine maintenance scheduling considering fine-grained resource management. Kovács et al. (2011) scheduled maintenance for onshore wind turbines over a one-day horizon. They aimed to minimize lost production due to maintenance and failures. They solved an integer linear programming (ILP) formulation of the problem with a commercial solver. For offshore wind farms, Irawan et al. (2017) solved a maintenance routing and scheduling problem minimizing labor, travel, and penalty costs. They proposed a method based on Dantzig–Wolfe decomposition in which the feasible routes for each vessel are generated a priori.

We consider the wind turbine maintenance scheduling problem, focusing on onshore wind farms, introduced by Froger et al. (2017). The problem is to provide a maintenance plan over a short-term horizon that maximizes wind electricity production while taking into account fine-grained resource management involving task assignments to a multi-skilled workforce. Froger et al. (2017) introduced several models based on ILP and constraint programming (CP). They found that the computational time grows prohibitively with problem size, and they therefore proposed a CP-based large neighborhood search (CPLNS) approach.

The primary contribution of this paper is an efficient exact approach for the wind turbine maintenance scheduling problem. We decompose the problem into a task scheduling subproblem and a technician-to-task assignment (TTA) subproblem, and we solve it using a branch-and-check (B&C) approach. While solving

---

<sup>1</sup>The Global Wind Energy Council - Global wind report annual market update 2015 - [http://www.gwec.net/wp-content/uploads/vip/GWEC-Global-Wind-2015-Report\\_April-2016\\_22.04.pdf](http://www.gwec.net/wp-content/uploads/vip/GWEC-Global-Wind-2015-Report_April-2016_22.04.pdf), last accessed: 2017-03-22

<sup>2</sup>percentage of time that the wind turbine is available to provide energy to the grid (downtime results from unexpected breakdowns and maintenance)

<sup>3</sup>ratio of the net electricity generated to the energy that could have been generated at continuous full-power operation during the given time period

the task scheduling subproblem, we discard, via cuts throughout the branch-and-bound tree, maintenance plans that cannot be performed by the technicians. In addition to the generic Benders cuts, we introduce problem-specific cuts and demonstrate they are key to speeding up the convergence of the approach.

The remainder of this paper is organized as follows. In Section 1 we describe the problem. In Section 2 we introduce new ILP formulations, and in Section 3 we present a B&C approach as an exact solution method. In Section 4 we discuss experiments with the 160-instance testbed proposed by Froger et al. (2017). Finally, in Section 5, we present our conclusions and discuss future research.

## 1 Problem statement

The goal is to schedule a set  $\mathcal{I}$  of maintenance tasks over a discrete and finite planning horizon  $\mathcal{T}$  while maximizing the revenue from the electricity production of a set  $\mathcal{W}$  of wind turbines. The turbines are spread across a set of locations  $\mathcal{L}$  (consisting of wind farms and possibly home depots). Let  $l_w \in \mathcal{L}$  be the location of turbine  $w \in \mathcal{W}$  and  $l_i$  the location where task  $i \in \mathcal{I}$  is performed.

The time horizon is a totally ordered set partitioned into  $|\mathcal{T}|$  periods of identical length.  $\mathcal{T}$  spans several days from a set  $\mathcal{D}$ . Let  $\mathcal{T}_d$  be the time periods of day  $d \in \mathcal{D}$ . Moreover, since the execution of a task can impact production during non-working hours, we introduce a special time period (a *rest period*) between two consecutive days to represent a night or weekend. The maintenance tasks are non-preemptive, but they are interrupted during the rest periods if they cover different days (e.g., a technician can start a task at the end of one day and complete it at the beginning of the next).

Although we do not include rest periods in  $\mathcal{T}$ , in the objective function we consider the loss of production generated by tasks covering these periods. In more detail, we use two parameters to define the impact of the tasks on the availability of the turbines. Binary parameter  $b_{wi}$  is 1 if and only if task  $i \in \mathcal{I}$  shuts down turbine  $w \in \mathcal{W}$  when technicians are working on the task. Binary parameter  $\tilde{b}_{wi}$  is 1 if and only if task  $i$  shuts down turbine  $w$  during the rest periods it covers. Parameters  $b_{wi}$  and  $\tilde{b}_{wi}$  are 0 if turbine  $w$  is not located where the task  $i$  is performed (i.e., if  $l_i \neq l_w$ ). A maintenance task can shut down more than one turbine, but this is rare in practice.

We have a finite set  $\mathcal{R}$  of maintenance technicians. Each technician has one or more skills from a set  $\mathcal{S}$ , and we introduce a binary vector  $\lambda_r$  over  $\mathcal{S}$  such that  $\lambda_{rs} = 1$  if and only if technician  $r$  has skill  $s \in \mathcal{S}$ . We assume that a technician cannot perform more than one task during a given period. Each task  $i \in \mathcal{I}$  requires technicians with a specific skill  $s_i \in \mathcal{S}$ . For convenience, we define  $\mathcal{R}_i$  to be the set of technicians that can perform task  $i$  (i.e.,  $\mathcal{R}_i = \{r \in \mathcal{R} \mid \lambda_{rs_i} = 1\}$ ).

To avoid expensive travel time, technicians must spend each day on tasks at *compatible locations*. Compatible locations are those that can be reached from each other in travel times that are negligible with respect to the duration of a time period. Let  $t_{max}$  be the maximum travel time between two locations that we can consider to be “negligible.” The top of Figure 1 shows the locations that are compatible with  $l_1$  (i.e.,  $l_2$  and  $l_3$ ). To model these *daily location-based incompatibilities* (DLBIs), we introduce the binary parameter  $\sigma_{ll'}$  taking the value 1 if and only if locations  $l$  and  $l'$  are compatible (naturally  $\sigma_{ll'} = \sigma_{l'l}$ ). The bottom of Figure 1 shows the four sets of compatible locations in our example. During a single day, a technician can execute tasks at  $l_1$  and  $l_2$  or  $l_3$  but not both. The maintenance tasks usually take hours (or days), and therefore the technicians travel between only a few locations during a single day.

Each technician  $r \in \mathcal{R}$  has an individual availability schedule expressed by a binary vector  $\pi_r$ , with  $\pi_r^t = 1$  if and only if  $r$  is available during period  $t \in \mathcal{T}$ . The availability schedule takes into account training days, vacations, and assignments to tasks that are not part of the optimization<sup>4</sup>. When technician  $r$  is not available

---

<sup>4</sup>tasks that have been already started before the beginning of the planning horizon or that are performed along with external companies

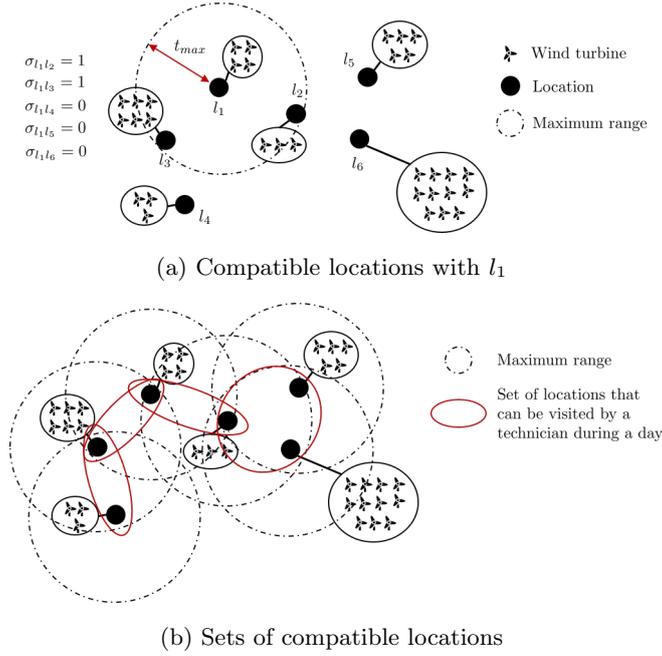


Figure 1: Illustration of daily location-based incompatibilities.

during period  $t$ , his or her location is fixed to  $l_r^t \in \mathcal{L}$ . For vacations and training days, this parameter is set to a dummy location  $l^*$  such that  $\forall l \in \mathcal{L}, \sigma_{l^*l} = 1$ . We assume that all the technicians work the same shift, which is common practice in this industry.

Each task  $i \in \mathcal{I}$  has a set  $\mathcal{M}_i$  of execution modes. For each mode  $m \in \mathcal{M}_i$ , there is an associated task duration  $d_{im}$  and a number  $q_{im}$  of required technicians. It is forbidden to switch modes after starting a task. Moreover, the technician assigned to a task must complete it, even if the task covers one or more rest periods.

The tasks must be executed during specific periods. These take into account the availability of spare parts and the external restrictions imposed by the operator. For safety, no task can be executed when the forecast wind speed is too strong. To model these restrictions, we introduce the binary parameter  $\gamma_i^t$  that is 1 if and only if task  $i \in \mathcal{I}$  can be performed during period  $t \in \mathcal{T}$ . We also impose nonoverlapping constraints on each subset of tasks that belongs to the set  $ov(\mathcal{I})$ .

The objective is to determine a schedule that maximizes the revenue generated by electricity production while satisfying the constraints described above. Let  $g_w^t$  be the revenue generated by turbine  $w$  if it can produce electricity during period  $t \in \mathcal{T}$ . Similarly, let  $\tilde{g}_w^d$  be the revenue generated by turbine  $w$  if it can produce electricity during the rest period following day  $d \in \mathcal{D}$ . These revenues are estimated based on forecast wind speed.

It is possible to postpone some tasks into the next planning horizon. When task  $i$  is postponed, there is penalty of  $o_i \geq 0$ . The value of the penalty includes an estimate of the loss of revenue induced by the postponement and the future planning of the task and may include outsourcing costs (the decision-maker will decide if the task should be outsourced rather than postponed). It also depends on the priority of the task (the higher the priority, the higher the penalty). The priority is based on reliability considerations (the longer a delay, the higher the probability of failure) and contract commitments. If the penalty is sufficiently high, a task will be postponed only if there is no available technician. In short, the objective function corresponds to the difference between the revenue and the penalties. We summarize the notation used in A.

We now analyze the complexity of this wind turbine maintenance scheduling problem (WTMSP). Let us first define the decision problem  $\text{WTMSP}^{dec}$  associated with WTMSP. In this problem, a parameter  $G \in \mathbb{R}$  is given as a lower bound on the value of the objective function.  $\text{WTMSP}^{dec}$  involves determining if there exists

a task schedule such that the objective value is greater than or equal to  $G$ . By polynomially reducing the cumulative scheduling problem<sup>5</sup>, which is known to be NP-complete in the strong sense (Baptiste et al., 1999), to WTMSP<sup>dec</sup>, we easily prove that WTMSP is strongly NP-hard. In some special cases, WTMSP is trivial. For instance, if the penalties are all 0, it is optimal to delay all the tasks. A similar observation is that each task with a penalty of 0 can be set to delayed a priori, without affecting the value of the optimal solution.

## 2 Integer linear programming formulations

In this section, we propose an ILP model that will serve as a baseline for the exact approach. This formulation is broadly inspired by the model introduced by Froger et al. (2017). We then propose an alternative formulation of the problem that provides a better basis for comparison.

### 2.1 Baseline formulation

The baseline formulation relies on the a priori generation for each task of all possible combinations of start period and execution mode. Such a combination is called a *plan*<sup>6</sup>; see Table 4 in C for examples of plans. Since the number of periods is limited and there are only a few execution modes, the total number of plans is reasonable. Let  $\mathcal{P}$  be the set of plans,  $i_p$  the task involved in plan  $p \in \mathcal{P}$ , and  $\mathcal{P}_i$  the set of all plans involving task  $i$  (i.e.,  $\mathcal{P}_i = \{p \in \mathcal{P} | i_p = i\}$ ). For each task  $i$ , we also create a plan  $p_i^0 \in \mathcal{P}_i$  representing the postponement of the task. For a plan  $p$ , the execution periods of  $i_p$  are expressed by a binary vector  $a_p$  over  $\mathcal{T}$  such that  $a_p^t = 1$  if and only if  $i_p$  is executed during period  $t \in \mathcal{T}$ . Let  $S_p$  and  $C_p$  be the start and finish periods of plan  $p$  (i.e.,  $S_p = \min_{t \in \mathcal{T}} a_p^t t$  and  $C_p = \max_{t \in \mathcal{T}} a_p^t t$ ). Similarly, we introduce a binary vector  $\tilde{a}_p$  over  $\mathcal{D}$  such that  $\tilde{a}_p^d = 1$  if and only if  $i_p$  covers the rest period following day  $d \in \mathcal{D}$ . Let  $\mathcal{D}_p$  be the set of days covered by plan  $p$ . For convenience and with a slight abuse of notation, we introduce the parameters  $s_p, l_p, b_{wp}$ , and  $\tilde{b}_{wp}$  equal to  $s_{i_p}, l_{i_p}, b_{wi_p}$ , and  $\tilde{b}_{wi_p}$ . Moreover, we denote by  $\mathcal{R}_p$  the set of technicians that can be assigned to plan  $p$ . Specifically,  $\mathcal{R}_p$  contains the technicians  $r \in \mathcal{R}_{i_p}$  such that for every period  $t$  we have  $\pi_r^t \geq a_p^t$  and for every day  $d \in \mathcal{D}_p$  and every period  $t \in \mathcal{T}_d$ , we have  $\pi_r^t = 1$  or both  $\pi_r^t = 0$  and  $\sigma_{i_p l_r^t} = 1$ . Let  $q_p$  be the number of technicians required for plan  $p$ . Finally, parameter  $o_p$  is the penalty incurred if plan  $p$  is selected (note that  $\forall i \in \mathcal{I}, \forall p \in \mathcal{P}_i \setminus \{p_i^0\}, o_p = 0$  and  $o_{p_i^0} = o_i$ ).

The ILP formulation has several types of decision variables. Binary variable  $x_p$  is 1 if and only if plan  $p \in \mathcal{P}$  is selected. Binary variable  $f_w^t$  is 1 if and only if turbine  $w \in \mathcal{W}$  can produce energy during period  $t \in \mathcal{T}$ , and binary variable  $\tilde{f}_w^d$  is 1 if and only if turbine  $w$  can produce energy during the rest period following day  $d \in \mathcal{D}$ .

The fundamental difference between our model and that of Froger et al. (2017) is that our model does not assign technicians to plans. Clearly, it does not matter which specific technicians we assign to plans with the same start and finish periods, the same location, and the same skill. These parameters define what we call a *pattern*. Let  $\mathcal{H}$  be the set of all patterns. For convenience, for pattern  $h \in \mathcal{H}$ , we introduce  $s_h, l_h$ , and  $\mathcal{R}_h$  that respectively define its required skill, its location, and the set of technicians that can be assigned. Conversely, let  $\mathcal{H}_l = \{h \in \mathcal{H} | l_h = l\}$  be the set of patterns associated with location  $l \in \mathcal{L}$ . The active periods for pattern  $h$  are represented by a binary vector  $a_h$  over  $\mathcal{T}$  such that  $a_h^t = 1$  if and only if  $h$  is active during period  $t \in \mathcal{T}$  ( $S_h$  and  $C_h$  are used to represent the start and finish periods of the pattern). Let  $\mathcal{P}_h$  be the set of plans sharing the same parameters as pattern  $h$ . For each pattern  $h \in \mathcal{H}$  and each technician  $r \in \mathcal{R}_h$ , we introduce the binary variable  $y_{rh}$  that is 1 if and only if technician  $r$  is assigned to pattern  $h$ . Assigning technicians to patterns

<sup>5</sup>A instance of the cumulative scheduling problem consists of a single resource with a given capacity  $C$  and a set  $\mathcal{J}$  of  $n$  jobs where each job  $j \in \mathcal{J}$  has a release date  $r_j$ , a deadline  $d_j$ , a processing time  $p_j$ , and a capacity resource requirement  $a_j$ . The problem is to determine if there exists a schedule of all the jobs satisfying the timing and resource capacity constraints.

<sup>6</sup>A plan associated with task  $i$  defines a feasible schedule for  $i$  by setting an execution mode  $m \in \mathcal{M}_i$  (and, by induction, a duration  $d_{im}$  and a technician requirement  $q_{im}$ ) and a consistent start date  $t$  such that  $i$  can be executed during every period  $t'$  such that  $t \leq t' < t + d_{im}$  (i.e.,  $\gamma_i^{t'} = 1$ ).

reduces the number of assignment variables. Lastly, we define the binary variable  $v_{rl}^t$  that is 1 if and only if technician  $r$  is at location  $l$  during period  $t$ .

The following ILP gives the baseline formulation,  $[P_1]$ :

$$[P_1] \quad \max \sum_{w \in \mathcal{W}} \left( \sum_{t \in \mathcal{T}} g_w^t f_w^t + \sum_{d \in \mathcal{D}} \tilde{g}_w^d \tilde{f}_w^d \right) - \sum_{p \in \mathcal{P}} o_p x_p \quad (1)$$

subject to:

$$\sum_{p \in \mathcal{P}_i} x_p = 1 \quad \forall i \in \mathcal{I}, \quad (2)$$

$$\sum_{i \in \mathcal{B}} \sum_{p \in \mathcal{P}_i} a_p^t x_p \leq 1 \quad \forall \mathcal{B} \in \text{ov}(\mathcal{I}), \forall t \in \mathcal{T}, \quad (3)$$

$$f_w^t + \sum_{p \in \mathcal{P}_i} b_{wp} a_p^t x_p \leq 1 \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \quad (4)$$

$$\tilde{f}_w^d + \sum_{p \in \mathcal{P}_i} \tilde{b}_{wp} \tilde{a}_p^d x_p \leq 1 \quad \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \forall d \in \mathcal{D}, \quad (5)$$

$$\sum_{i \in \mathcal{I}} \sum_{s_i \in \bar{\mathcal{S}}} a_p^t q_p x_p \leq |R_{\bar{\mathcal{S}}}^t| \quad \forall t \in \mathcal{T}, \forall \bar{\mathcal{S}} \subseteq \mathcal{S}, \quad (6)$$

$$\sum_{r \in \mathcal{R}_h} y_{rh} = \sum_{p \in \mathcal{P}_h} q_p x_p \quad \forall h \in \mathcal{H}, \quad (7)$$

$$\sum_{h \in \mathcal{H}_l} a_h^t y_{rh} \leq \pi_r^t v_{rl}^t \quad \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T}, \quad (8)$$

$$\sum_{l \in \mathcal{L}} v_{rl}^t = 1 \quad \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \quad (9)$$

$$v_{rl}^t = 1 \quad \forall r \in \mathcal{R}, \forall t \in \mathcal{T} \text{ s.t. } \pi_r^t = 0, \quad (10)$$

$$v_{rl}^t + \sum_{l' \in \mathcal{L} | \sigma_{ll'} = 0} v_{rl'}^t \leq 1 \quad \forall r \in \mathcal{R}, \forall d \in \mathcal{D}, \forall (t, t') \in \mathcal{T}_d \times \mathcal{T}_d, t \neq t', \forall l \in \mathcal{L}, \quad (11)$$

$$x_p \in \{0, 1\} \quad \forall p \in \mathcal{P}, \quad (12)$$

$$f_w^t \in \{0, 1\} \quad \forall w \in \mathcal{W}, \forall t \in \mathcal{T}, \quad (13)$$

$$\tilde{f}_w^d \in \{0, 1\} \quad \forall w \in \mathcal{W}, \forall d \in \mathcal{D}, \quad (14)$$

$$y_{rh} \in \{0, 1\} \quad \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_h, \quad (15)$$

$$v_{rl}^t \in \{0, 1\} \quad \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T} \quad (16)$$

The objective in (1) is the difference between the revenue generated by the turbines and the postponement penalties. Constraints (2) ensure that at least one plan involving each task is selected (i.e., each task is either executed or postponed). Constraints (3) are the nonoverlapping constraints. Constraints (4) and (5) compute the impact of the tasks on the availability of the turbines during the periods of  $\mathcal{T}$  and the rest periods. Constraints (7) ensure that the technician requirements are fulfilled: the number of technicians assigned to every pattern  $h$  must be equal to the number of technicians required by the selected plans associated with  $h$ . Constraints (8) couple the locations of the technicians to the tasks they perform. Constraints (9) prevent technicians from performing multiple tasks during the same period. When technicians are not available, constraints (10) ensure compliance with their known locations. Constraints (11) define the DLBIs. Finally, constraints (12)–(16) are the binary restrictions.

Sometimes commercial solvers can more easily solve models with redundant constraints (because of their different structure). With this in mind, we include the cumulative scheduling constraints (6) in  $[P_1]$  although they can be deduced from constraints (7) and (8). To build these constraints, we introduce for each period  $t \in \mathcal{T}$  the bipartite graph  $\mathcal{G}^t = ((\mathcal{S}, \mathcal{R}^t), \mathcal{U}^t)$  in which, with a slight abuse of notation, vertices from  $\mathcal{S}$  represent skills, vertices from  $\mathcal{R}^t$  indicate the technicians available during period  $t$  (i.e.,  $\mathcal{R}^t = \{r \in \mathcal{R} | \pi_r^t = 1\}$ ), and

edges from  $\mathcal{U}^t$  are defined as follows:  $\forall s \in \mathcal{S}, \forall r \in \mathcal{R}^t \quad (s, r) \in \mathcal{U}^t$  if and only if  $\lambda_{r,s} = 1$ . Via a generalization of the König–Hall theorem, constraints (6) then correspond to a necessary and sufficient condition for the existence of a maximum cardinality b-matching (MCbM) from  $\mathcal{S}$  to  $\mathcal{R}^t$ . Here the function  $b$  is defined by  $b(s) = \sum_{i \in \mathcal{I} | s_i = s} \sum_{p \in \mathcal{P}_i} a_p^t q_p x_p$  for every vertex  $s$  in  $\mathcal{S}$ , and by  $b(r) = 1$  for every vertex  $r$  in  $\mathcal{R}^t$ . To express these constraints, let  $\mathcal{R}_{\mathcal{S}}^t = \{r \in \mathcal{R} | \exists s \in \mathcal{S}, \lambda_{r,s} = 1 \wedge \pi_r^t = 1\}$  be the set of technicians available during period  $t$  with at least one skill in subset  $\mathcal{S} \subseteq \mathcal{S}$ . The number of these constraints is exponential:  $(2^{|\mathcal{S}|} - 1) \times |\mathcal{T}|$ . In our experiments, however, the number tends to be small; we therefore add them all to our model.

## 2.2 Alternative formulation

A potential improvement to the model concerns the space-time tracking of the technicians. Observing that the number of constraints (11) is usually large, we develop an alternative technician management strategy. This strategy is based on the daily assignment of each technician to a set of compatible locations. To compute these sets, we compute all the maximal cliques (cliques that cannot be enlarged) in a graph where each vertex represents a location, and there is an edge between two vertices if the underlying locations  $l$  and  $l'$  can be visited during the same day by the same technician (i.e.,  $\sigma_{ll'} = 1$ ). Figure 2 illustrates the construction of the graph and highlights the maximal cliques. We use the algorithm of Bron and Kerbosch (1973) to find these cliques.

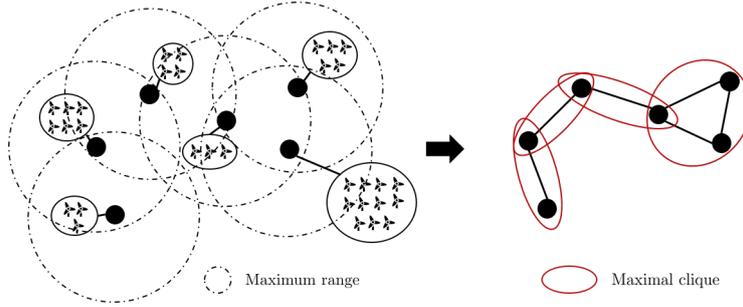


Figure 2: Construction of the graph used to compute the maximal cliques on an example.

Let  $\mathcal{K}$  be the set of all maximal cliques in the graph. We define  $\mathcal{K}_r^d$  to be the set of cliques to which technician  $r$  can be assigned during day  $d$ , i.e.,  $\{k \in \mathcal{K} | \forall t \in \mathcal{T}_d, \pi_r^t = 1 \vee (\pi_r^t = 0 \wedge l_r^t \in k)\}$ . We then introduce binary variable  $u_{rk}^d$  that is 1 if and only if during day  $d \in \mathcal{D}$  technician  $r \in \mathcal{R}$  can perform tasks only at locations included in clique  $k \in \mathcal{K}_r^d$ . Let  $d_t$  be the day to which period  $t$  belongs. We can track the location of the technicians with the following constraints:

$$\sum_{k \in \mathcal{K}_r^d} u_{rk}^d = 1 \quad \forall r \in \mathcal{R}, \forall d \in \mathcal{D}, \quad (17)$$

$$\sum_{h \in \mathcal{H} | r \in \mathcal{R}_h} a_h^t y_{rh} \leq \pi_r^t \quad \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \quad (18)$$

$$\sum_{h \in \mathcal{H}_l} a_h^t y_{rh} \leq \sum_{k \in \mathcal{K}_r^{d_t} | l \in k} u_{rk}^{d_t} \quad \forall r \in \mathcal{R}, \forall l \in \mathcal{L}, \forall t \in \mathcal{T}, \quad (19)$$

$$u_{rk}^d \in \{0, 1\} \quad \forall r \in \mathcal{R}, \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_r^d \quad (20)$$

Constraints (17) state that a technician is assigned to only one clique on each day. This ensures compliance with the DLBIs. Constraints (18) prevent a technician from being assigned to multiple tasks during a given period. Constraints (19) couple the assignment and space-time tracking variables, ensuring that a technician can be assigned to location  $l$  during period  $t$  only if he or she is assigned to a clique containing  $l$  during the day associated with  $t$ .

The model resulting from replacing constraints (8), (9), (10), (11), and (16) by (17), (18), (19), and (20) in the baseline formulation  $[P_1]$  is called  $[P_2]$ .

### 3 Exact solution approach

#### 3.1 Problem decomposition

The exact approach presented in this section takes advantage of the intrinsic decomposition of the problem into a task scheduling subproblem and a TTA subproblem. The scheduling problem selects a plan for each task to maximize the difference between the revenue from the electricity production and the postponement penalties. In this problem, technician considerations are partially eliminated. Given a fixed selection of plans, the TTA subproblem determines if the technician requests can be satisfied while imposing the DLBIs and the individual resource availabilities. We aim to coordinate these two problems. Note that an optimal solution to the scheduling problem leading to a feasible TTA is optimal for the overall problem.

We first introduce the scheduling problem. An initial formulation  $[ShP_1]$  is:

$$[ShP_1] \quad \max \sum_{w \in \mathcal{W}} \left( \sum_{t \in \mathcal{T}} g_w^t f_w^t + \sum_{d \in \mathcal{D}} \tilde{g}_w^d \tilde{f}_w^d \right) - \sum_{p \in \mathcal{P}} o_p x_p \quad (21)$$

subject to:

$$(2), (3), (4), (5), (6), (12), (13), (14)$$

The cumulative scheduling constraints (6) in  $[ShP_1]$  help to speed up the convergence of a coordination procedure between the two problems. Solving the task scheduling problem without any information on the availability of the technicians would result in a selection of plans that would be unlikely to lead to a feasible TTA.

We now assume that we have a solution to  $[ShP_1]$ , which is a fixed selection  $(\bar{x}_p)_{p \in \mathcal{P}}$  of plans (hereafter referred to simply as  $\bar{x}$ ). An ILP formulation  $[SP_2(\bar{x})]$  of the TTA subproblem is:

$$[SP_2(\bar{x})] \quad \min \sum_{h \in \mathcal{H}} \theta_h \quad (22)$$

subject to:

$$\sum_{r \in \mathcal{R}_h} y_{rh} + \theta_h = \sum_{p \in P_h} q_p \bar{x}_p \quad \forall h \in \mathcal{H}, \quad (23)$$

$$\sum_{\substack{h \in H \\ \text{s.t. } r \in \mathcal{R}_h}} y_{rh} \leq 1 \quad \forall r \in \mathcal{R}, \forall H \in \mathcal{C}^{max}(G), \quad (24)$$

$$\theta_h \geq 0 \quad \forall h \in \mathcal{H}, \quad (25)$$

$$y_{rh} \in \{0, 1\} \quad \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_h \quad (26)$$

We introduce slack variables  $(\theta_h)_{h \in \mathcal{H}}$  for the technician requirement constraints (23). The unavailability periods of each technician are respected by the definition of the set  $\mathcal{R}_h$ . The clique constraints (24) ensure that the TTAs comply with the DLBIs. Specifically, set  $\mathcal{C}^{max}(G)$  contains all the maximal cliques in graph  $G$  where each vertex represents a pattern in  $\mathcal{H}$ , and there is an edge between two vertices if the underlying patterns  $h$  and  $h'$  cannot be visited by the same technician. This edge exists if the following clause holds:

$$(S_{h'} \leq C_h \wedge S_h \leq C_{h'}) \vee ((\exists d \in \mathcal{D}, \mathcal{T}_d \cap \{S_h, S_h + 1, \dots, C_h\} \neq \emptyset \wedge \mathcal{T}_d \cap \{S_{h'}, S_{h'} + 1, \dots, C_{h'}\} \neq \emptyset) \wedge \sigma_{l_{h'} l_h} = 0) \quad (27)$$

One can visualize  $G$  as an extended version of an interval graph. To define the TTA subproblem for a given solution  $\bar{x}$  (and therefore to compute all the maximal cliques), we need only to consider the patterns of  $\mathcal{H}(\bar{x}) = \{h \in \mathcal{H} \mid \sum_{p \in P_h} q_p \bar{x}_p > 0\}$ . This significantly reduces the number of variables in  $[SP_2(\bar{x})]$  and the number of clique constraints (24) since we consider a subgraph of  $G$ .

The TTA subproblem is NP-complete; this follows from its equivalence to a L-coloring problem. Nonetheless, under certain assumptions, it is solvable in polynomial time. We omit the details here; see B for a discussion of the complexity of the TTA subproblem.

We observe that  $[SP_2(\bar{x})]$  always has a feasible solution because of the slack variables  $(\theta_h)_{h \in \mathcal{H}}$ . However, the TTA subproblem is feasible only if the value of the optimal solution is zero.

When the value of the optimal solution to  $[SP_2(\bar{x})]$  is strictly positive (i.e., the TTA subproblem is infeasible), one can use a *combinatorial Benders* (CB) cut (Codato and Fischetti, 2006) to discard the current solution  $\bar{x}$  in  $[ShP_1]$ . This cut is similar to those defined for logic-based Benders decomposition (Hooker and Ottosson, 2003) and have some similarities with the integer optimality cuts for the integer L-shaped method (Laporte and Louveaux, 1993). It is also known as a *no-good* cut. Observing that a solution always contains  $|\mathcal{I}|$  nonzero variables  $x_p$  since exactly one plan must be selected per task, and using the binary variables  $(x_p)_{p \in \mathcal{P}}$ , we can define a CB cut as follows (with  $\mathcal{P}(\bar{x}) = \{p \in \mathcal{P} \mid \bar{x}_p = 1\}$ ):

$$\sum_{p \in \mathcal{P}(\bar{x})} x_p \leq |\mathcal{I}| - 1 \quad (28)$$

Clearly, this cover inequality requires at least one of the variables in  $\bar{x}$  to change its value.

Since the feasible region of our problem is bounded, the number of integer points satisfying all the constraints of  $[ShP_1]$  is finite, and thus the same holds for the number of CB cuts. Let  $\bar{\mathcal{F}}$  be the set of all solutions  $\bar{x}$  to  $[ShP_1]$  that lead to an infeasible TTA subproblem. The overall maintenance scheduling problem can therefore be reformulated as the following master problem  $[P]$ :

$$\begin{aligned} [P] \quad & \max \sum_{w \in \mathcal{W}} \left( \sum_{t \in \mathcal{T}} g_w^t f_w^t + \sum_{d \in \mathcal{D}} \tilde{g}_w^d \tilde{f}_w^d \right) - \sum_{p \in \mathcal{P}} o_p x_p \\ & \text{subject to:} \\ & (2), (3), (4), (5), (6), (12), (13), (14) \\ & \sum_{p \in \mathcal{P}(\bar{x})} x_p \leq |\mathcal{I}| - 1 \quad \forall \bar{x} \in \bar{\mathcal{F}} \end{aligned} \quad (28)$$

We denote by  $[RMP]$  a restricted master problem (RMP) of problem  $[P]$  that contains at most a small subset of constraints (28). To efficiently solve the problem while exploiting this Benders-like decomposition, we adopt a B&C approach. The B&C framework was introduced by Thorsteinsson (2001); it is primarily designed for LP and CP hybridization. Cuts are generated on the fly during the solution of the initial RMP. At each integer node of the branch-and-bound tree, the corresponding solution is sent to the TTA subproblem. Benders and/or problem-specific cuts are generated to discard solutions leading to an infeasible TTA. Thorsteinsson (2001) applied B&C to a planning and scheduling problem, while Sadykov (2008) used it for a complex scheduling problem on a single machine. See Beck (2010) for a survey. This approach is also referred to as a *Benders-based branch-and-cut* algorithm (Naoum-Sawaya and Elhedhli, 2010) and as a *branch-and-Benders-cut method* (Gendron et al., 2014). It has been used to solve several types of problems: hub location (De Camargo et al., 2011), production routing under demand uncertainty (Adulyasak et al., 2015), location-design (Gendron et al., 2014), facility location and network design (Naoum-Sawaya and Elhedhli, 2010), and hop-constrained survivable network design (Botton et al., 2013). Botton et al. (2013) reported a significant improvement using this approach instead of the classical implementation of Benders decomposition, while Gendron et al. (2014) outlined the benefits in terms of solution quality, scalability, and robustness.

The efficiency of B&C is primarily based on the constraints we generate to cut off infeasible solutions to  $[RMP]$ . In the following subsection, we describe our cut generation procedure.

## 3.2 Cut generation procedure

For every solution  $\bar{x}$  to  $[RMP]$ , we must check the feasibility of the TTA subproblem. We can solve  $[SP_2(\bar{x})]$  to optimality using a commercial solver, but this approach has two major drawbacks. First, since  $[SP_2(\bar{x})]$  is a pure ILP model, solving it may be too time-consuming. Second, if the optimal value of  $[SP_2(\bar{x})]$  is strictly positive, the resulting CB cut (28) may be too weak because it discards only the current solution. To overcome these drawbacks and to build stronger cuts, we propose three cut generation strategies based on different approximations to the TTA subproblem.

### 3.2.1 Benders feasibility cuts

Let  $[SP_2^{LR}(\bar{x})]$  be the linear relaxation of  $[SP_2(\bar{x})]$ . Constraints (26) of  $[SP_2(\bar{x})]$  are replaced in  $[SP_2^{LR}(\bar{x})]$  by the following constraints:

$$y_{rh} \leq 1 \quad \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_h, \quad (29)$$

$$y_{rh} \geq 0 \quad \forall h \in \mathcal{H}, \forall r \in \mathcal{R}_h \quad (30)$$

We can generate cuts based on the solution to the linear relaxation  $[SP_2^{LR}(\bar{x})]$  of  $[SP_2(\bar{x})]$ . Since a solution  $\bar{x}$  to  $[RMP]$  is feasible for the whole problem only if the optimum of  $[SP_2^{LR}(\bar{x})]$  is zero,  $\bar{x}$  is feasible for the whole problem only if the optimum of the dual  $[DSP_2^{LR}(\bar{x})]$  of  $[SP_2^{LR}(\bar{x})]$  is less than or equal to zero (by the duality theorem). Let us associate the dual variables  $\iota_h$ ,  $\varrho_r^H$ , and  $\varphi_{rh}$  with constraints (23), (24), and (29), respectively. The objective function of  $[DSP_2^{LR}(\bar{x})]$ , denoted  $\Theta_{\bar{x}}(\iota, \varrho, \varphi)$ , is:

$$\Theta_{\bar{x}}(\iota, \varrho, \varphi) = \sum_{h \in \mathcal{H}} \left( \sum_{p \in \mathcal{P}_h} q_p \bar{x}_p \iota_h + \sum_{r \in \mathcal{R}_h} \varphi_{rh} \right) + \sum_{r \in \mathcal{R}} \sum_{\substack{H \in \mathcal{C}(\mathcal{H}) \\ \text{s.t. } r \in \bigcup_{h \in H} \mathcal{R}_h}} \varrho_r^H \quad (31)$$

Let  $D$  be the polyhedron defined by the constraints of the dual problem  $[DSP_2^{LR}(\bar{x})]$ . Since  $[SP_2^{LR}(\bar{x})]$  always has a feasible solution,  $[DSP_2^{LR}(\bar{x})]$  is bounded and achieves its optimum at an extreme point of  $D$ . Let  $\eta^1, \eta^2, \dots, \eta^n$  (with  $\eta^k = (\iota^k, \varrho^k, \varphi^k)$ ) be the finite set of extreme points of  $D$ . By the weak duality theorem, the following inequalities must hold to ensure the existence of a zero-valued solution to  $[SP_2^{LR}(x)]$ :

$$\Theta_x(\iota^k, \varrho^k, \varphi^k) \leq 0 \quad \forall k \in \{1, \dots, n\} \quad (32)$$

Constraints (32) are the classical Benders feasibility (BF) cuts. Because the constraint matrix of  $[SP_2^{LR}(\bar{x})]$  is not totally unimodular, relaxing the integrality constraints (26) may lead to a fractional solution. A cut generation algorithm responsible for identifying violated constraints (32) will therefore not, in general, find a feasible solution to  $[P]$ . Nevertheless, identifying violated BF cuts may help us to generate fewer CB cuts. The advantages are: i) BF cuts can be computed by solving a continuous linear model, and ii) they may discard more solutions than just the current solution to  $[RMP]$ . One drawback of BF cuts is that they are generic and therefore likely to be weak.

### *Problem specific-cuts*

For the efficient coordination of the task scheduling and the TTA subproblem, we must find reduced subsets of plans causing the infeasibility of the latter. In the following subsections, we describe two problem-specific procedures to find these reduced subsets, and we show how we build stronger problem-specific cuts.

First, we introduce the following TTA requirements:

[C1] Each task must be performed by technicians with the appropriate skill.

[C2] A technician cannot perform more than one task during a given period.

[C3] The technician assignments must not violate the DLBIs.

[C4] Each technician has an availability schedule that must be respected.

[C5] The technician assigned to a task must complete it, even if it covers some rest periods.

Obviously, the TTA subproblem is feasible if and only if [C1], [C2], [C3], [C4], and [C5] are satisfied.

Second, to simplify the discussion, we introduce the concept of *jobs*. A job refers to either a pattern or a technician unavailability period. More precisely, a job  $j$  is defined by  $(l_j, S_j, C_j, s_j, \mathcal{R}_j, q_j)$ , where  $l_j$  is the location where  $j$  is executed,  $S_j$  its start period,  $C_j$  its finish period,  $\mathcal{S}_j$  a set of skills such that the technician performing  $j$  must have at least one of these skills,  $\mathcal{R}_j$  the set of technicians who can perform  $j$ , and  $q_j$  the number of technicians required for job  $j$ . For every unavailability period of technician  $r \in \mathcal{R}$  occurring at period  $t$  ( $t \in \mathcal{T}$  such that  $\pi_r^t = 1$ ), we build an artificial job defined by  $(l_r^t, t, t, \mathcal{S}_r, \{r\}, 1)$  where  $\mathcal{S}_r = \{s \in \mathcal{S} \mid \lambda_{rs} = 1\}$  is the set of skills of technician  $r$ . If a technician is unavailable during contiguous periods and if he or she is assigned each time to the same location (with regard to  $l_r^t$ ), we associate only one job with these unavailability periods<sup>7</sup>. Let  $\mathcal{J}^{\mathcal{R}}$  be the set of jobs associated with the technician unavailability periods. We also associate with each solution  $\bar{x}$  to [RMP] a set  $\mathcal{J}(\bar{x})$  of jobs. We build this set in two steps. First, we add to  $\mathcal{J}(\bar{x})$  all the jobs of  $\mathcal{J}^{\mathcal{R}}$ . Then, for every pattern  $h \in \mathcal{H}$  such that  $q_h(\bar{x}) > 0$  (with  $q_h(\bar{x}) = \sum_{p \in \mathcal{P}_h} q_p \bar{x}_p$ ), we create a job defined by  $(l_h, S_h, C_h, \{s_h\}, \mathcal{R}_h, q_h(\bar{x}))$ . Let  $\mathcal{J}^{\mathcal{H}}(\bar{x})$  be the set of jobs associated with patterns. We also define the parameter  $h_j$  to be the pattern associated with job  $j \in \mathcal{J}^{\mathcal{H}}(\bar{x})$ .

### 3.2.2 Maximum cardinality b-matching cuts

In this section, we extend the idea used to generate constraints (6) by partially taking into account [C3] and [C5] while fully taking into account [C1], [C2], and [C4]. More precisely, when building the potential assignments of technicians to tasks, we jointly consider the unavailability periods of the technicians and the restriction of not switching technicians during the execution of a task. We then show that the TTA subproblem can be approximated by a series of MCbM problems (as many as the length of the time horizon).

First, for a fixed period  $t \in \mathcal{T}$  and a given solution  $\bar{x}$  to [RMP], we introduce an undirected graph  $\check{G}^t(\bar{x})$  composed of:

- A set of vertices  $\check{V}^t$  where  $\check{V}^t = \check{V}_{\mathcal{J}}^t \cup \check{V}_{\mathcal{R}}^t$ :
  - $\check{V}_{\mathcal{J}}^t$ : for each job  $j \in \mathcal{J}(\bar{x})$  such that  $S_j \leq t \leq C_j$ , we add a vertex in  $\check{V}_{\mathcal{J}}^t$ . Parameter  $j_\nu$  denotes the job associated with  $\nu \in \check{V}_{\mathcal{J}}^t$ . Conversely,  $\nu_j$  denotes the vertex associated with job  $j$ .
  - $\check{V}_{\mathcal{R}}^t$ : a vertex of  $\check{V}_{\mathcal{R}}^t$  represents a technician  $r \in \mathcal{R}$  during period  $t$ . Let  $r_\nu$  be the technician associated with  $\nu \in \check{V}_{\mathcal{R}}^t$ .
- A set of edges  $\check{U}^t$  such that  $\forall \nu_1 \in \check{V}_{\mathcal{J}}^t, \forall \nu_2 \in \check{V}_{\mathcal{R}}^t: (\nu_1, \nu_2) \in \check{U}^t \Leftrightarrow r_{\nu_2} \in \mathcal{R}_{j_{\nu_1}}$ .

We formally describe in Proposition 3.1 and Corollary 3.1 the link between the TTA subproblem and a series of MCbM problems defined in the above graphs.

**Proposition 3.1.** *Let  $\bar{x}$  be a solution to [RMP] and assume that [C3] and [C5] are not taken into account. The TTA subproblem for  $\bar{x}$  is equivalent to a series of  $|\mathcal{T}|$  MCbM problems in  $\check{G}^t(\bar{x})$  where for each period  $t \in \mathcal{T}$  function  $b$  is defined by  $b_\nu = q_{j_\nu}$  for  $\nu \in \check{V}_{\mathcal{J}}^t$  and by  $b_\nu = 1$  for  $\nu \in \check{V}_{\mathcal{R}}^t$ .*

<sup>7</sup>If for technician  $r$  there exists  $t_1, t_2 \in \mathcal{T}, t_1 < t_2$  and  $l \in \mathcal{L}$  such that for all  $t \in \mathcal{T}, t_1 \leq t \leq t_2$ , we have  $\pi_r^t = 0$  and  $l_r^t = l$ , then we create a single job  $(l, t_1, t_2, \mathcal{S}_r, \{r\}, 1)$ .

*Proof.* Since [C3] and [C5] are not taken into account, the TTA subproblem can be independently solved for each period of the planning horizon.

Assuming that during each period  $t$  we can find a MCbM in each of the graphs  $G^t(\bar{x})$  with a cardinality equal to  $\sum_{j \in \mathcal{J}(\bar{x})} \mathbb{1}_{[S_j, C_j]}(t) q_j$ , we can immediately build a solution to the TTA subproblem from the selected edges by making the underlying assignments.

Assume now that we have a feasible solution to the TTA subproblem. For each period  $t$ , we can build a b-matching in  $\check{G}^t(\bar{x})$  from the working schedule of each technician during this specific period. If technician  $r \in \mathcal{R}$  is assigned to pattern  $h \in \mathcal{H}$  during period  $t$ , we select the edge  $(\nu_1, \nu_2) \in \mathcal{U}$  where  $j_{\nu_1} = j_h$  ( $j_h$  denoting the job associated with pattern  $h$ ) and  $r_{\nu_2} = r$ . This construction ensures the building of a b-matching. Moreover, since all the requirements are fulfilled, this b-matching has maximum cardinality.  $\square$

**Corollary 3.1.** *If we assume that [C3] and [C5] are not taken into account, the TTA subproblem is feasible for a solution  $\bar{x}$  to [RMP] if and only if the MCbM in  $\check{G}^t(\bar{x})$  for each period  $t \in \mathcal{T}$  contains  $\sum_{j \in \mathcal{J}(\bar{x})} \mathbb{1}_{[S_j, C_j]}(t) q_j$  edges of  $\check{\mathcal{U}}^t$ .*

*Proof.* This is a direct consequence of Proposition 3.1.  $\square$

Given a fixed period  $t$ , finding the MCbM in  $\check{G}^t(\bar{x})$  from  $\check{\mathcal{V}}_{\mathcal{J}}^t$  to  $\check{\mathcal{V}}_{\mathcal{R}}^t$  is equivalent to solving a maximum flow problem in a slightly modified version of this graph. We use this equivalence to derive new cuts. Let  $\hat{G}^t(\bar{x})$  be this new directed graph and  $\hat{\mathcal{V}}^t$  and  $\hat{\mathcal{U}}^t$  the new sets of vertices and arcs. We build  $\hat{G}^t(\bar{x})$  as follows:

1. We define  $\hat{\mathcal{V}}^t = \check{\mathcal{V}}^t \cup \{\bar{\omega}^t, \underline{\omega}^t\}$  where  $\bar{\omega}^t$  and  $\underline{\omega}^t$  represent the source and sink vertices.
2. For every directed arc  $(\nu_1, \nu_2) \in \hat{\mathcal{U}}^t$ , let  $\gamma_{\nu_1 \nu_2}^{max}$  be its maximal capacity and  $f_{\nu_1 \nu_2}$  the flow on the arc. We formally define  $\hat{\mathcal{U}}^t$  as follows:
  - $\forall \nu_1 \in \check{\mathcal{V}}_{\mathcal{J}}^t, \forall \nu_2 \in \check{\mathcal{V}}_{\mathcal{R}}^t: (\nu_1, \nu_2) \in \hat{\mathcal{U}}^t \Leftrightarrow r_{\nu_2} \in \mathcal{R}_{j_{\nu_1}}$  and  $\gamma_{\nu_1 \nu_2}^{max} = +\infty$
  - $\forall \nu \in \check{\mathcal{V}}_{\mathcal{J}}^t: (\bar{\omega}^t, \nu) \in \hat{\mathcal{U}}^t$  and  $\gamma_{\bar{\omega}^t \nu}^{max} = q_{j_\nu}$
  - $\forall \nu \in \check{\mathcal{V}}_{\mathcal{R}}^t: (\nu, \underline{\omega}^t) \in \hat{\mathcal{U}}^t$  and  $\gamma_{\nu \underline{\omega}^t}^{max} = 1$

Let  $f^*(t, \bar{x})$  be the value of the maximum flow in  $\hat{G}^t(\bar{x})$ . If  $f^*(t, \bar{x}) < \sum_{\nu \in \check{\mathcal{V}}_{\mathcal{J}}^t} q_{j_\nu}$  then the jobs of  $\mathcal{J}(\bar{x})$  covering  $t$  cannot be fully scheduled during this period. We therefore have to discard the solution  $\bar{x}$ . We first compute the minimum flow cut in  $\hat{G}^t(\bar{x})$  (see Figure 3). This cut can be described by two disjoint sets  $\hat{\mathcal{V}}^t(+)$  and  $\hat{\mathcal{V}}^t(-)$  where the source  $\bar{\omega}^t$  and sink  $\underline{\omega}^t$  belong to  $\hat{\mathcal{V}}^t(+)$  and  $\hat{\mathcal{V}}^t(-)$ , respectively. For cut modeling purposes, we also introduce the sets  $\check{\mathcal{V}}_{\mathcal{J}}^t(-) = \check{\mathcal{V}}_{\mathcal{J}}^t \cap \hat{\mathcal{V}}^t(-)$ ,  $\check{\mathcal{V}}_{\mathcal{J}}^t(+)$ ,  $\check{\mathcal{V}}_{\mathcal{R}}^t(-) = \check{\mathcal{V}}_{\mathcal{R}}^t \cap \hat{\mathcal{V}}^t(-)$ , and  $\check{\mathcal{V}}_{\mathcal{R}}^t(+)$ .

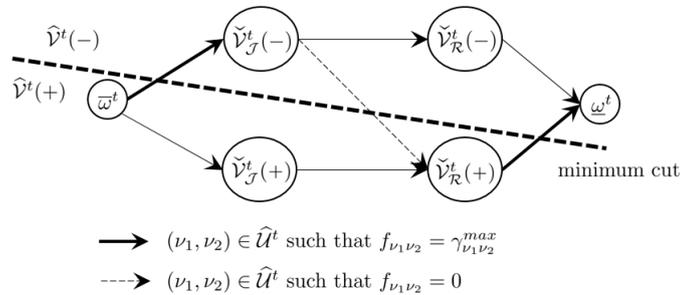


Figure 3: Minimum cut in graph  $\hat{G}^t(\bar{x})$ .

Applying the *max-flow/min-cut* theorem to  $\hat{G}^t(\bar{x})$ , we have

$$f^*(t, \bar{x}) = \sum_{\nu \in \check{\mathcal{V}}_{\mathcal{J}}^t(-)} \gamma_{\bar{\omega}^t \nu}^{max} + \sum_{\nu \in \check{\mathcal{V}}_{\mathcal{R}}^t(+)} \gamma_{\nu \underline{\omega}^t}^{max} \quad (33)$$

If we replace the capacity of each arc by its value, we obtain

$$f^*(t, \bar{x}) = \sum_{\nu \in \check{\mathcal{V}}_{\mathcal{J}}^t(-)} q_{j_\nu} + |\check{\mathcal{V}}_{\mathcal{R}}^t(+)| \quad (34)$$

The valid minimum flow cut (that invalidates  $\bar{x}$ ) is

$$\sum_{\nu \in \check{\mathcal{V}}_{\mathcal{J}}^t(-)} q_{j_\nu} + |\check{\mathcal{V}}_{\mathcal{R}}^t(+)| \geq \sum_{\nu \in \check{\mathcal{V}}_{\mathcal{J}}^t(+)} q_{j_\nu} \quad (35)$$

which we can reformulate as follows:

$$\sum_{\nu \in \check{\mathcal{V}}_{\mathcal{J}}^t(+)} q_{j_\nu} \leq |\check{\mathcal{V}}_{\mathcal{R}}^t(+)| \quad (36)$$

Inequality (36) leads to the following valid constraint (hereafter referred to as a MCbM cut) that eliminates  $\bar{x}$  from the feasible region of  $[RMP]$ :

$$\sum_{\substack{j \in \mathcal{J}^{\mathcal{H}}(\bar{x}) \\ \text{s.t. } \nu_j \in \check{\mathcal{V}}_{\mathcal{J}}^t(+)}} \sum_{p \in \mathcal{P}_{h_j}} q_p x_p \leq |\check{\mathcal{V}}_{\mathcal{R}}^t(+)| - \sum_{\substack{j \in \mathcal{J}^{\mathcal{R}} \\ \text{s.t. } \nu_j \in \check{\mathcal{V}}_{\mathcal{J}}^t(+)}} q_j \quad (37)$$

It is possible to strengthen the MCbM cut by considering its composition. First,  $|\check{\mathcal{V}}_{\mathcal{R}}^t(+)| \neq |\mathcal{R}|$  because at least one of constraints (6) would be unsatisfied otherwise. This also implies that  $|\check{\mathcal{V}}_{\mathcal{R}}^t(-)| \neq 0$ . By definition of the max-flow/min-cut, the technicians associated with  $\check{\mathcal{V}}_{\mathcal{R}}^t(-)$  are either not connected to any other vertices or they are assigned to  $j \in \mathcal{J}(\bar{x})$  such that  $\nu_j \in \check{\mathcal{V}}_{\mathcal{J}}^t(-)$ , but they cannot be assigned to any  $j$  such that  $\nu_j \in \check{\mathcal{V}}_{\mathcal{J}}^t(+)$ . Either these technicians do not have the required skills for those jobs or they have at least one unavailability period that prevents the assignment. We deduce that inequality (36) is also valid for every potential job that covers period  $t$  and cannot be performed by any of the technicians associated with a vertex of  $\check{\mathcal{V}}_{\mathcal{R}}^t(-)$ . The MCbM cut (37) can therefore be rewritten as

$$\sum_{h \in \mathcal{H}} \Psi(h, t, \check{\mathcal{V}}_{\mathcal{R}}^t(-)) \sum_{p \in \mathcal{P}_h} q_p x_p \leq |\check{\mathcal{V}}_{\mathcal{R}}^t(+)| - \sum_{\substack{j \in \mathcal{J}^{\mathcal{R}} \\ \text{s.t. } \nu_j \in \check{\mathcal{V}}_{\mathcal{J}}^t(+)}} q_j \quad (38)$$

where  $\Psi(h, t, \check{\mathcal{V}}_{\mathcal{R}}^t(-))$  is 1 if and only if pattern  $h$  covers period  $t$  and none of the technicians associated with  $\check{\mathcal{V}}_{\mathcal{R}}^t(-)$  can be assigned to  $h$ .

We must solve the MCbM problem only for every period  $t$  where at least one technician cannot be assigned to a job because of an unavailability period occurring at a period other than  $t$ . Otherwise, constraints (6) are necessary and sufficient conditions for the existence of b-matchings with the desired cardinality.

### 3.2.3 Maximum-weight clique cuts

Another strategy to discard a solution  $\bar{x}$  to  $[RMP]$  that leads to an infeasible TTA relies on showing that it is impossible to assign the technicians to the tasks without violating the DLBIs. Since the DLBIs are defined by day, this search decomposes into  $|\mathcal{D}|$  independent searches in which for each  $d \in \mathcal{D}$  we consider only the jobs of  $\mathcal{J}(\bar{x})$  that cover  $d$ . During the search, it is also necessary to take into account the skills required to perform the different jobs. For a fixed subset  $\bar{\mathcal{S}} \subseteq \mathcal{S}$ , we consider only the jobs  $j \in \mathcal{J}(\bar{x})$  such that  $\mathcal{S}_j \cap \bar{\mathcal{S}} \neq \emptyset$  and the technicians with at least one skill in this subset. This procedure increases the likelihood of finding violated DLBIs if the current solution to the RMP does not lead to a feasible TTA subproblem. This is particularly true when the ratio between the requirements and the number of available technicians varies widely across skills.

To look for violated DLBIs, we solve for each day  $d$  and each subset  $\bar{\mathcal{S}} \subseteq \mathcal{S}$  a maximum-weight clique (MWC) problem in an undirected graph  $\tilde{G}_{\bar{\mathcal{S}}}^d(\bar{x})$ . The graph  $\tilde{G}_{\bar{\mathcal{S}}}^d(\bar{x})$  is composed of:

- A set of vertices  $\tilde{\mathcal{V}}_{\bar{\mathcal{S}}}^d$  such that each vertex maps a job  $j$  of set  $\mathcal{J}(\bar{x})$  that i) covers day  $d$  (i.e.,  $\mathcal{T}_d \cap \{S_j, \dots, C_j\} \neq \emptyset$ ) and ii) requires at least one of the skills in  $\bar{\mathcal{S}}$  (i.e.,  $S_j \cap \bar{\mathcal{S}} \neq \emptyset$ ). Let  $j_\nu$  be the job associated with  $\nu \in \tilde{\mathcal{V}}_{\bar{\mathcal{S}}}^d$ . We associate with every vertex  $\nu$  a weight equal to the number of technicians  $q_{j_\nu}$  required to perform job  $j_\nu$ .
- A set of edges  $\tilde{\mathcal{U}}_{\bar{\mathcal{S}}}^d$  where for  $\nu_1, \nu_2 \in \tilde{\mathcal{V}}_{\bar{\mathcal{S}}}^d$ :  
 $(\nu_1, \nu_2) \in \tilde{\mathcal{U}}_{\bar{\mathcal{S}}}^d \Leftrightarrow \nu_1 \neq \nu_2 \wedge \left( (S_{j_{\nu_2}} \leq C_{j_{\nu_1}} \wedge S_{j_{\nu_1}} \leq C_{j_{\nu_2}}) \vee \sigma_{l_{j_{\nu_1}} l_{j_{\nu_2}}} = 0 \right)$   
There is an edge between  $\nu_1$  and  $\nu_2$  in  $\tilde{G}_{\bar{\mathcal{S}}}^d(\bar{x})$  if and only if a technician cannot be assigned to both jobs  $j_{\nu_1}$  and  $j_{\nu_2}$  with regard to [C2] and [C3].

Proposition 3.2 formally describes the link between the solution of the TTA subproblem and the solution of MWC problems.

**Proposition 3.2.** *The TTA subproblem is feasible for a solution  $\bar{x}$  to [RMP] if for each subset  $\bar{\mathcal{S}} \subseteq \mathcal{S}$  of skills and each day  $d \in \mathcal{D}$  the maximum weight of a clique in  $\tilde{G}_{\bar{\mathcal{S}}}^d(\bar{x})$  is less than or equal to  $|\mathcal{R}_{\bar{\mathcal{S}}}|$ .*

*Proof.* For a fixed day  $d$  and a fixed subset of skills  $\bar{\mathcal{S}}$ , suppose for the sake of contradiction that the maximum weight of a clique in  $\tilde{G}_{\bar{\mathcal{S}}}^d(\bar{x})$  is strictly greater than  $|\mathcal{R}_{\bar{\mathcal{S}}}|$ . By construction of  $\tilde{G}_{\bar{\mathcal{S}}}^d(\bar{x})$ , a technician cannot perform more than one job among the jobs whose vertices belong to that clique. Since we need more technicians than are available, the TTA subproblem is infeasible.  $\square$

Let us now assume a fixed day  $d \in \mathcal{D}$  and a fixed subset of skills  $\bar{\mathcal{S}}$ . Let  $\mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})$  be the set of vertices that belong to the MWC of  $\tilde{G}_{\bar{\mathcal{S}}}^d(\bar{x})$ . If the total weight of the vertices in  $\tilde{G}_{\bar{\mathcal{S}}}^d(\bar{x})$  is strictly greater than  $|\mathcal{R}_{\bar{\mathcal{S}}}|$ , the constraint that eliminates  $\bar{x}$  is:

$$\sum_{\nu \in \mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})} q_{j_\nu} \leq |\mathcal{R}_{\bar{\mathcal{S}}}| \quad (39)$$

The valid constraint (hereafter referred to as a MWC cut) that discards a solution  $\bar{x}$  to [RMP] is therefore

$$\sum_{\substack{j \in \mathcal{J}^{\mathcal{H}}(\bar{x}) \\ \text{s.t. } \nu_j \in \mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})}} \sum_{p \in \mathcal{P}_{h_j}} q_p x_p \leq |\mathcal{R}_{\bar{\mathcal{S}}}| - \sum_{\substack{j \in \mathcal{J}^{\mathcal{R}} \\ \text{s.t. } \nu_j \in \mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})}} q_j \quad (40)$$

This cut simply states that the number of technicians required by the jobs associated with the vertices of the clique must be less than the number of technicians with at least one skill in  $\bar{\mathcal{S}}$ . We have derived this cut not only for the MWC but also for all the cliques that have a weight greater than  $|\mathcal{R}_{\bar{\mathcal{S}}}|$ .

It is possible to tighten the MWC cut (40) by adding additional patterns on its left side. We proceed as follows. First, we consider the subgraph  $G^{sub}(\bar{x})$  of  $G$  ( $G$  is used to derive the clique inequalities in  $[SP_2(\bar{x})]$ ) that includes a vertex for pattern  $h \in \mathcal{H}$  if i)  $h \in \mathcal{H}(\bar{x})$  (i.e.,  $q_h(\bar{x}) = 0$ ), ii)  $h$  covers day  $d$ , iii)  $s_h \in \bar{\mathcal{S}}$ , and iv) if a technician is assigned to  $h$ , he or she cannot be assigned to any job involved in  $\mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})$  with regard to [C2] and [C3]. We then solve a maximum clique problem in  $G^{sub}(\bar{x})$ . Let  $\mathcal{H}[\mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})]$  be the set of patterns associated with the vertices that are part of the maximum clique of  $G^{sub}(\bar{x})$ . Since a technician cannot be assigned to more than one of the jobs in  $\mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})$  or one of the patterns of  $\mathcal{H}[\mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})]$ , we can rewrite the MWC cut (40) as follows:

$$\sum_{\substack{j \in \mathcal{J}^{\mathcal{H}}(\bar{x}) \\ \text{s.t. } \nu_j \in \mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})}} \sum_{p \in \mathcal{P}_{h_j}} q_p x_p + \sum_{h \in \mathcal{H}[\mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})]} \sum_{p \in \mathcal{P}_h} q_p x_p \leq |\mathcal{R}_{\bar{\mathcal{S}}}| - \sum_{\substack{j \in \mathcal{J}^{\mathcal{R}} \\ \text{s.t. } \nu_j \in \mathcal{C}_{\bar{\mathcal{S}}}^d(\bar{x})}} q_j \quad (41)$$

For efficiency, we reduce the size of  $G^{sub}(\bar{x})$  since it suffices to consider one vertex for all the patterns with the same location and coverage of the same portion of the day. This remark also applies to  $\tilde{G}_{\bar{\mathcal{S}}}^d(\bar{x})$ .

Last but not least, to avoid overloading our algorithm, we solve the MWC problem only if i) the sum of the weights of the vertices is greater than the number of available technicians, and ii) during a particular day there

are at least two jobs that do not overlap and are executed at incompatible locations. Otherwise, constraints (6) ensure for each day and each subset of skills the nonexistence of a clique with a weight strictly greater than the number of available technicians. We use the algorithms introduced by Östergård (2001) and Östergård (2002) to solve the maximum clique and MWC problems.

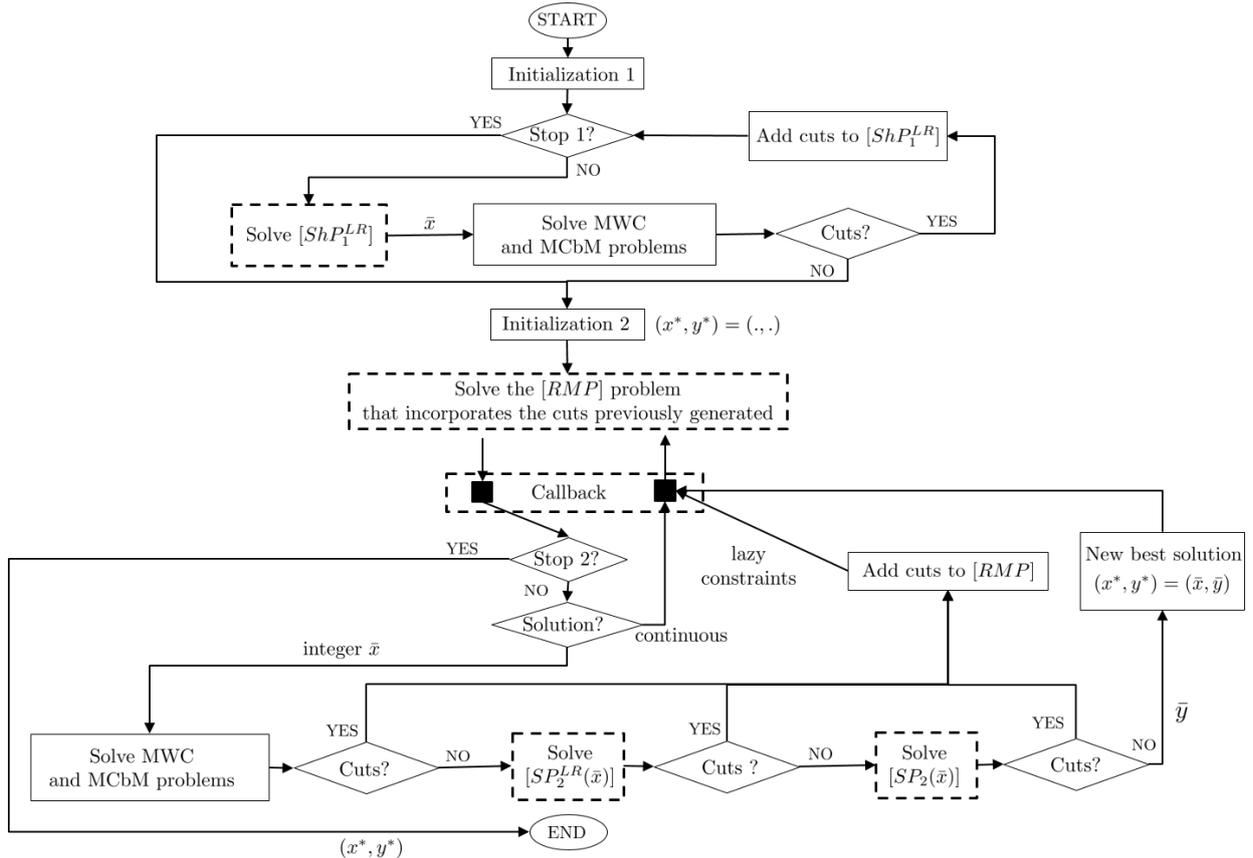
Since the approximations to the TTA subproblem described in Sections 3.2.2 and 3.2.3 can be decomposed into a series of small problems, we can potentially identify multiple subsets of plans that cause the infeasibility of the TTA subproblem. This usually leads to the generation of multiple cuts, which is known to significantly improve the efficiency of a cut generation process. We can also consider solving the small problems in parallel.

For clarity, we provide three examples in C to illustrate how we build up the different cuts. For the approximations described in Sections 3.2.2 and 3.2.3, the first two examples show that neither the MCbM nor MWC cuts strictly dominate the other. The third example illustrates a case where the two approximations do not find any cuts although the TTA subproblem is infeasible.

### 3.3 General structure of algorithm

We now provide a full description of our two-stage B&C approach. Figure 4 outlines the general structure. The two stages are as follows:

Figure 4: Flow chart of our B&C approach.



N.B.: The steps with dashed outlines are performed by an ILP solver.

- Stage 1 (solve a linear relaxation of  $[P]$ ):

In the first stage we generate potentially useful MCbM and MWC cuts while working with a much easier problem. We consider  $[ShP_1^{LR}]$ , the linear relaxation of  $[ShP_1]$  (“Initialization 1”). We solve the problem to optimality with an LP solver (“Solve  $[ShP_1^{LR}]$ ”). We then solve the approximations to the TTA subproblem

described in Section 3.2.2 (the MCbM becomes a fractional MCbM) and in Section 3.2.3 (at this stage, we derive a cut only for the MWC) for the continuous solution  $\bar{x}$  of  $[ShP_1^{LR}]$  (“Solve MWC and MCbM problems”). If we generate cuts, we add them to  $[ShP_1^{LR}]$  (“Add cuts to  $[ShP_1^{LR}]$ ”) and reoptimize the resulting problem. Otherwise, we terminate this first stage (“Cuts? No”).

To avoid wasting time generating cuts that may not be useful, we perform at most 100 iterations (“Stop 1”). Note that this stage could have been performed at the root-node of the search tree defined for the second stage. However, our aim is to take advantage of the preprocessing techniques embedded in commercial ILP solvers.

- Stage 2 (Solve the master problem  $[P]$ ):

In the second stage, we solve the problem by a branch-and-check method implemented on top of a commercial solver. We initialize  $[RMP]$  with all the cuts that have been generated during the first stage (“Initialization 2”) and we solve it by a branch-and-bound algorithm. For every integer solution  $\bar{x}$  to  $[RMP]$  obtained in the branch-and-bound tree (“Solution? integer”), we determine if  $\bar{x}$  is feasible for the TTA subproblem. We start by solving the MCbM and MWC problems (“Solve MWC and MCbM problems”). If this produces at least one cut, we discard the current solution  $\bar{x}$  by adding the generated cut(s) to  $[RMP]$  (“Add cuts to  $[RMP]$ ”). Otherwise, we solve the LP formulation  $[SP_2^{LR}(\bar{x})]$  (“Solve  $[SP_2^{LR}(\bar{x})]$ ”). If we identify a violated BF cut of type (32), we add it to  $[RMP]$  (“Add cuts to  $[RMP]$ ”). Otherwise, we cannot determine the feasibility of the TTA subproblem before solving the ILP formulation  $[SP_2(\bar{x})]$  (“Solve  $[SP_2(\bar{x})]$ ”). If its optimal value is strictly positive, we generate a CB cut of type (28) and add it to  $[RMP]$  (“Add cuts to  $[RMP]$ ”). Otherwise, we conclude that  $(\bar{x}, \bar{y})$  (with  $\bar{y}$  the solution of  $[SP_2(\bar{x})]$ ) is a new feasible solution (“New best solution”). Note that the branch-and-bound scheme ensures that  $(\bar{x}, \bar{y})$  is strictly better than the best previous solution. We can therefore update the best solution so far,  $(x^*, y^*)$ , to  $(\bar{x}, \bar{y})$ . We terminate this phase (“Stop 2”) with the optimal solution to  $[P]$  or a feasible solution if the time limit has been reached (if no solution is found within the time limit, one can always consider the feasible solution in which all the tasks are postponed).

From an implementation point of view, we start this stage by simply forwarding an ILP formulation of the RMP to the solver. The cut generation procedure is implemented inside a callback routine that is invoked by the solver at every node of the branch-and-bound tree. The cuts computed during this stage are provided to the solver as *lazy constraints*. After adding them at a particular node, the solver checks the feasibility of the integer solutions in terms of these constraints. If the candidate solution at an integer node is not feasible, the solver discards it and adds the violated lazy constraints to the active nodes of the branch-and-bound tree.

## 4 Computational experiments

### 4.1 Instances

We report results for the 160-instance testbed proposed by Froger et al. (2017). These instances were randomly generated based on insight into wind prediction and maintenance operations obtained from industrial partners. Froger et al. (2017) considered time horizons of different lengths (5 and 10 days with 2 or 4 periods per day), different numbers of tasks (20, 40, 80), and different numbers of skills (1 or 3). For each combination of parameters, they generated two categories of instances: five “type A” instances with a tight technician-to-work ratio (i.e., the technicians can perform most but not necessarily all of the tasks during the planning horizon), and five “type B” instances with a regular technician-to-work ratio (i.e., the technicians can perform all the tasks). The resulting 32 families of instances are as representative as possible of the situations that may occur. The cost of postponement is set in such a way that postponing a task is nonprofitable. We refer to each

instance family via “a\_b.c.d.e” where a, b, c, d, and e refer to the number of periods in the planning horizon, the number of periods within a day, the number of skills, the number of tasks, and the technician-to-work ratio, respectively. For a discussion of the instance generation process see Froger et al. (2017).

## 4.2 Results

We implemented our algorithms using *Java 8 (JVM 1.8.0.25)*. We use *Gurobi 6.5.1* to solve the LP and ILP models. We ran our experiments on a Linux 64 bit-machine, with an Intel(R) Xeon(R) X5675 (3.07 Ghz) and 12 GB of RAM. We set a 3-hour time limit (the CPU times are reported in seconds and rounded to the nearest integer). We implemented the cut generation procedure of our B&C approach using the callback provided by Gurobi<sup>8</sup>. To assess our results, we compute the gap with respect to the optimal solution when it is known, or to the best upper bound found by the solver over all the tests reported, i.e.,  $gap = (z^{UB} - z)/|z|$ , where  $z$  is the objective of the computed solution and  $z^{UB}$  is the objective of the optimal solution or the best upper bound found by any of our methods.

We tested four exact approaches. We present the results for the direct solution of the two ILP formulations of the problem ( $[P_1]$  and  $[P_2]$ ). To quantify the impact of the problem-specific cuts of Section 3.2, we present the results of the B&C approach presented in Section 3.3 without  $(\overline{B\&C})$  and with  $(B\&C)$  the MCbM and MWC cuts. These approaches differ only in how we discard an infeasible solution  $\bar{x}$  to  $[RMP]$ . In Table 1, we report the average, over all the instances belonging to the same family or sharing a common characteristic, of the gap (Gap), the solution time (Time), and the percentage of tasks scheduled in the best solution (%S). We also report the number of optimal solutions found within the time limit (#Opt). We use bold font to indicate the best approach for the given family. See D for the detailed results for each instance. For a meaningful comparison, the average solution time takes into account only those instances for which an optimal solution is found within the time limit. Similarly, the average gap and the percentage of tasks scheduled (i.e., not postponed) takes into account only the instances that are not optimally solved. Since postponement is heavily penalized, a large gap often indicates a low percentage of scheduled tasks. This allows a better understanding of the results. On average 99% of the tasks are scheduled in the optimal or best-known solutions.

We observe that  $[P_2]$  seems to outperform  $[P_1]$ . However, it is difficult to compare these formulations: the best model in terms of gap and solution time if the problem is not optimally solved can vary within a family, even if  $[P_2]$  usually gives a tighter LP bound (see in Appendix D). We also observe that the average gap is considerable for most of the families (the solver fails to schedule a large proportion of the tasks), and the computational time is significant. This is because the formulations involve many binary variables and constraints. We conclude that directly solving the ILP formulations with a commercial solver is not the best approach.

The performance of the B&C approach is strongly correlated with the cuts generated from the approximations to the TTA subproblem. The problem-specific cuts allow us to find the optimal solution for 63 additional instances. On the remaining instances, they significantly reduce the gap from around 4.0%. This highlights the strength of the problem-specific cuts.

The B&C approach outperforms by far the direct solution of the ILP formulations for every family. It solves to optimality 80% of the instances and reduces the computational time (by around 3 min on average). Moreover, the overall average gap when optimality is not reached is relatively small (1.7%).

Our results suggest that the number of skills does not have a significant impact on the difficulty of the instances (although instances with three skills appear to be easier to solve). This may be because there are fewer symmetries among technicians and a smaller number of feasible configurations for the task scheduling. On the other hand, the number of tasks seems to have an impact when the technician-to-work ratio is tight.

---

<sup>8</sup>A callback is a user function that is called periodically by Gurobi to allow the user to query or modify the state of the optimization (Gurobi Optimizer Reference Manual: <http://www.gurobi.com/documentation/6.5/refman.pdf>)

Table 1: Detailed computational results for the exact approaches.

Family	[P <sub>1</sub> ]				[P <sub>2</sub> ]				$\overline{B\&C}$				B&C			
	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time
10.2.1.20.A	-	-	5/5	598	-	-	5/5	152	-	-	5/5	2,272	-	-	5/5	<b>5</b>
10.2.1.20.B	-	-	5/5	12	-	-	5/5	37	-	-	5/5	2	-	-	5/5	<b>1</b>
10.2.1.40.A	0.01%	100%	2/5	1,813	0.01%	100%	1/5	7	0.70%	100%	2/5	3,379	-	-	<b>5/5</b>	<b>7</b>
10.2.1.40.B	-	-	5/5	205	-	-	5/5	121	-	-	5/5	8	-	-	5/5	<b>1</b>
10.2.3.20.A	-	-	5/5	2,635	-	-	5/5	1,574	1.0%	97%	2/5	2,871	-	-	5/5	<b>162</b>
10.2.3.20.B	-	-	5/5	31	-	-	5/5	18	-	-	5/5	94	-	-	5/5	<b>2</b>
10.2.3.40.A	-	-	5/5	3,220	-	-	5/5	3,996	2.2%	98%	2/5	4,040	-	-	5/5	<b>17</b>
10.2.3.40.B	-	-	5/5	180	-	-	5/5	295	-	-	5/5	18	-	-	5/5	<b>1</b>
20.2.1.40.A	2.4%	98%	2/5	8,074	1.4%	98%	2/5	2,858	6.4%	96%	1/5	7,301	-	-	<b>5/5</b>	230
20.2.1.40.B	0.01%	100%	4/5	232	-	-	5/5	2,078	-	-	5/5	695	-	-	5/5	<b>4</b>
20.2.1.80.A	436%	0%	0/5	-	334%	20%	0/5	-	8.2%	96%	0/5	-	0.02%	100%	<b>4/5</b>	300
20.2.1.80.B	318%	50%	3/5	1,485	229%	49%	3/5	3,823	0.13%	100%	4/5	580	-	-	<b>5/5</b>	<b>5</b>
20.2.3.40.A	1.25%	99%	1/5	322	1.2%	99%	3/5	5,534	4.6%	95%	1/5	1,136	2.1%	98%	<b>4/5</b>	40
20.2.3.40.B	-	-	5/5	376	-	-	5/5	155	-	-	5/5	525	-	-	5/5	<b>3</b>
20.2.3.80.A	257%	20%	0/5	-	156%	39%	0/5	-	3.3%	98%	0/5	-	-	-	<b>5/5</b>	51
20.2.3.80.B	-	-	5/5	3,415	196%	50%	3/5	2,456	0.02%	100%	4/5	163	-	-	<b>5/5</b>	<b>5</b>
20.4.1.20.A	1.8%	96%	0/5	-	1.3%	97%	0/5	-	2.1%	95%	0/5	-	2.2%	95%	<b>3/5</b>	1,715
20.4.1.20.B	-	-	5/5	204	-	-	5/5	405	-	-	5/5	131	-	-	<b>5/5</b>	<b>2</b>
20.4.1.40.A	264%	0%	0/5	-	61%	75%	0/5	-	8.6%	93%	0/5	-	1.2%	98%	<b>2/5</b>	1,586
20.4.1.40.B	174%	49%	1/5	5,208	107%	74%	1/5	1,968	0.3%	100%	4/5	630	-	-	<b>5/5</b>	<b>6</b>
20.4.3.20.A	1.2%	98%	0/5	-	2.4%	95%	4/5	5,005	3.1%	95%	0/5	-	2.0%	95%	<b>4/5</b>	237
20.4.3.20.B	0.01%	100%	4/5	52	-	-	5/5	113	3.1%	95%	4/5	134	-	-	5/5	<b>9</b>
20.4.3.40.A	373%	20%	0/5	-	5.3%	95%	0/5	-	6.7%	94%	0/5	-	0.85%	98%	<b>1/5</b>	8,888
20.4.3.40.B	1.5%	99%	2/5	6,112	0.03%	100%	3/5	2,108	0.29%	100%	1/5	1,608	-	-	<b>5/5</b>	11
40.4.1.40.A	352%	0%	0/5	-	106%	76%	0/5	-	15.7%	89%	0/5	-	<b>2.1%</b>	98%	0/5	-
40.4.1.40.B	1,594%	40%	0/5	-	3.0%	98%	0/5	-	0.7%	100%	0/5	-	-	-	<b>5/5</b>	31
40.4.1.80.A	4,948%	0%	0/5	-	4,948%	0%	0/5	-	16.1%	90%	0/5	-	<b>1.5%</b>	99%	0/5	-
40.4.1.80.B	331%	0%	0/5	-	331%	0%	0/5	-	1.8%	99%	0/5	-	-	-	<b>5/5</b>	89
40.4.3.40.A	1,087%	20%	0/5	-	4.6%	96%	0/5	-	14.6%	90%	0/5	-	<b>1.5%</b>	99%	0/5	-
40.4.3.40.B	477%	20%	0/5	-	0.84%	99%	2/5	2,118	0.3%	100%	0/5	-	-	-	<b>5/5</b>	36
40.4.3.80.A	2,813%	0%	0/5	-	2,727%	18%	0/5	-	14.3%	90%	0/5	-	<b>2.3%</b>	98%	0/5	-
40.4.3.80.B	3,899%	0%	0/5	-	3,899%	0%	0/5	-	0.96%	100%	0/5	-	-	-	<b>5/5</b>	86

Characteristics		[P <sub>1</sub> ]				[P <sub>2</sub> ]				$\overline{B\&C}$				B&C			
		Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time	Gap	%S	#Opt	Time
S	1	854%	35%	32/80	1,108	621%	61%	34/80	1,035	6.7%	95%	36/80	957	1.6%	98%	<b>64/80</b>	179
	3	1,036%	39%	37/80	1,677	982%	61%	43/80	1,896	4.9%	96%	29/80	722	1.7%	98%	<b>64/80</b>	186
T	2	179%	56%	57/80	1,383	144%	60%	57/80	1,461	3.9%	97%	51/80	982	1.0%	99%	<b>78/80</b>	49
D	4	1,197%	30%	12/80	1,555	1,014%	61%	20/80	1,757	6.5%	95%	14/80	380	1.7%	98%	<b>50/80</b>	390
Type	A	878%	38%	20/80	2,618	759%	65%	25/80	2,838	7.8%	94%	13/80	3,106	1.7%	98%	<b>48/80</b>	456
	B	1,059%	35%	49/80	921	802%	53%	52/80	913	0.8%	100%	52/80	288	-	-	<b>80/80</b>	18
All		940%	37%	69/160	1,413	773%	61%	77/160	1,538	5.7%	96%	65/160	852	1.7%	98%	<b>128/160</b>	182

This can be explained by the greater difficulty of finding a maintenance plan when there are more tasks. For the instances with a regular technician-to-work ratio, although the number of plans is larger when there are more tasks, the number of patterns does not grow proportionally, which results in a moderately more complicated TTA subproblem. Moreover, our exact approaches are more appropriate for two periods per day rather than four, because the computational time is lower and the number of optimal solutions is larger. A plausible explanation is that the DLBIs are tighter on instances with more periods per day. More periods provide a wider choice of task start times and therefore more opportunities to move technicians between locations. There is also a larger number of plans and patterns; this may contribute to the level of difficulty. In conclusion, according to our experiments, the difficulty of an instance increases with the number of periods per day and the tightness of the technician-to-work ratio.

Table 2 gives the average number of cuts generated during the execution of the B&C approach for each group of tests. The overall average (#Cuts) is decomposed into CB, BF, MCbM, and MWC cuts. Detailed results for each family of instances are available in D.

On average, 90% of the cuts are problem-specific and 10% are generic Benders cuts. These results indicate that the approximations are not always able to identify the infeasibility of the TTA subproblem. However, when  $|\mathcal{S}| = 1$ , solving the MCbM and MWC problems almost always identifies the infeasibility. We generate more cuts for instances with three skills, four periods per day, and a tight technician-to-work ratio. This is because of the larger number of patterns in the first two cases and because there is less flexibility in the scheduling of the tasks in the third case. We never generate CB cuts. The optimal solution to  $[SP_2^{LR}(\bar{x})]$  is usually integer although the constraint matrix is not totally unimodular<sup>9</sup>. We generate only a few MCbM cuts and many MWC cuts. This is because the RMP has no information about the DLBIs at the beginning of the optimization. It is thus more likely that these constraints are not satisfied by the RMP solutions.

Table 2: Average number of cuts generated in B&C approach.

Characteristic	#Cuts	CB	Other cuts			
			BF	MCbM	MWC	
$\mathcal{S}$	1	102	0	0.3	3	98
	3	224	0	28	10	186
$\mathcal{T}$	2	56	0	9	3	44
	4	270	0	19	10	241
$\mathcal{D}$	A	262	0	24	10	229
	B	63	0	4	4	56
All		163	0	14	7	142

Furthermore, all the components of the B&C approach have a favorable trade-off between efficiency and computational time. Since the relaxation of the problem considered in the first stage contains only continuous variables and at most a few cuts, this stage requires on average just 1% of the time, and the limit on the number of iterations is never reached in our experiments. The results also show that solving the RMP via the B&C approach is the most time-consuming part of the second stage, requiring on average 99% of the time. This compares with the negligible time to solve, given a solution  $\bar{x}$  to  $[RMP]$ ,  $[SP_2^{LR}(\bar{x})]$  and  $[SP_2(\bar{x})]$  with the commercial solver, or the approximations to the TTA subproblem.

## 5 Conclusions and future research

We have proposed a B&C approach for a challenging maintenance scheduling problem arising in the onshore wind industry. This exact method decomposes the problem into a task scheduling problem and a TTA sub-

<sup>9</sup>We found an instance—not part of our testbed—and a solution  $\bar{x}$  of  $[RMP]$  where the optimal value of  $[SP_2^{LR}(\bar{x})]$  is 0 whereas the optimal value of  $[SP_2(\bar{x})]$  is 1.

problem. For each selection of plans, we determine the existence of an assignment of the technicians to the scheduled tasks that adheres to the technician availability and the daily travel limitations. Since the ILP formulation of the TTA subproblem does not have the integrality property, we use CB cuts to cut off infeasible maintenance plans in the RMP, while trying to identify violated classical Benders cuts beforehand. However, the key components of the algorithm are the approximations of the TTA subproblem as a series of MCbM and MWC problems. Our results show that the resulting problem-specific cuts are effective in speeding up the convergence of the B&C approach. The B&C method finds optimal solutions in reasonable computational times and provides high-quality integer solutions when optimality is not reached. It significantly outperforms the direct solution of the ILP models.

In future work, we will explore speeding up the solution of the problem (especially when integer solutions are scarce) by checking the approximations of the TTA subproblem at non-integer nodes. This may produce cuts that eliminate continuous infeasible solutions. However, adding too many such cuts to the RMP may decrease the efficiency of the approach since it is likely to increase the time needed to solve the LP relaxation at each node. In preliminary experiments, we tested two strategies for generating MWC and MCbM cuts at non-integer nodes. First, we considered all the plans  $p \in \mathcal{P}$  such that  $x_p > 0$  and arbitrarily added the cuts where the value of the slack variable was greater than or equal to 0.5. We also stopped the search for a cut at a node as soon as the first cut was found. Second, we considered all the plans  $p \in \mathcal{P}$  such that  $x_p = 1$ . We did not observe a significant improvement with either of these strategies. However, we could investigate more advanced strategies for generating cuts at non-integer nodes. Another possibility to speed up the solution of the problem would be to solve the LP relaxation of the RMP at each node using column generation. It would then be necessary to express the cuts (generated when the TTA subproblem is infeasible) with alternative variables to simplify the solution of the pricing problem. Last but not least, we have considered the deterministic version of the problem, but there is an inherent uncertainty in wind-speed forecasts. Future research will focus on solving the resulting stochastic optimization problem using stochastic programming and/or robust optimization techniques.

## References

- Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867.
- Baptiste, P., Le Pape, C., and Nuijten, W. (1999). Satisfiability tests and time-bound adjustments for cumulative scheduling problems. *Annals of Operations Research*, 92:305–333.
- Beck, J. (2010). Checking-up on branch-and-check. *Principles and Practice of Constraint Programming CP 2010 SE - 10*, 6308:84–98.
- Biro, M., Hujter, M., and Tuza, Z. (1992). Precoloring extension. I. Interval graphs. *Discrete Mathematics*, 100(1):267–279.
- Botton, Q., Fortz, B., Gouveia, L., and Poss, M. (2013). Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal On Computing*, 25(1):13–26.
- Bron, C. and Kerbosch, J. (1973). Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577.
- Codato, G. and Fischetti, M. (2006). Combinatorial Benders’ cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766.

- De Camargo, R., De Miranda, G., and Ferreira, R. (2011). A hybrid outer-approximation/Benders decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters*, 39(5):329–337.
- Ding, F., Tian, Z., and Jin, T. (2013). Maintenance modeling and optimization for wind turbine systems: A review. In *2013 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)*, pages 569–575.
- Froger, A., Gendreau, M., Mendoza, J., Pinson, E., and Rousseau, L.-M. (2016). Maintenance scheduling in the electricity industry: A literature review. *European Journal of Operational Research*, 251(3):695–706.
- Froger, A., Gendreau, M., Mendoza, J., Pinson, E., and Rousseau, L.-M. (2017). Solving a wind turbine maintenance scheduling problem. *Journal of Scheduling*. doi: 10.1007/s10951-017-0513-5.
- Gendron, B., Scutellà, M., Garroppo, R., Nencioni, G., and Tavanti, L. (2014). A branch-and-Benders-cut method for nonlinear power design in green wireless local area networks. Technical Report CIRRELT-2014-42, CIRRELT.
- Hooker, J. N. and Ottosson, G. (2003). Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60.
- Irawan, C., Ouelhadj, D., Jones, D., Stålhane, M., and Sperstad, I. (2017). Optimisation of maintenance routing and scheduling for offshore wind farms. *European Journal of Operational Research*, 256(1):76–89.
- Jensen, T. and Toft, B. (2011). *Graph Coloring Problems*, volume 39. John Wiley & Sons.
- Kovács, A., Erdős, G., Viharos, Z., and Monostori, L. (2011). A system for the detailed scheduling of wind farm maintenance. *CIRP Annals - Manufacturing Technology*, 60(1):497–501.
- Laporte, G. and Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142.
- Naoum-Sawaya, J. and Elhedhli, S. (2010). An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, 210(1):33–55.
- Östergård, P. (2001). A new algorithm for the maximum-weight clique problem. *Nordic Journal of Computing*, 8(4):424–436.
- Östergård, P. (2002). A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1):197–207.
- Sadykov, R. (2008). A branch-and-check algorithm for minimizing the weighted number of late jobs on a single machine with release dates. *European Journal of Operational Research*, 189(3):1284–1304.
- Thorsteinsson, E. (2001). Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In Walsh, T., editor, *Principles and Practice of Constraint Programming CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 16–30. Springer Berlin Heidelberg.

## A Notation

$\mathcal{T}$	time horizon (totally ordered set)
$\mathcal{D}$	set of days
$\mathcal{T}_d$	set of time periods for day $d \in \mathcal{D}$
$\mathcal{L}$	set of locations
$\mathcal{S}$	set of skills
$\mathcal{R}$	set of technicians
$\mathcal{R}_i$	set of technicians that can be assigned to task $i \in \mathcal{I}$
$\mathcal{W}$	set of turbines
$\mathcal{I}$	set of tasks
$\mathcal{P}$	set of plans
$\mathcal{D}_p$	set of days covered by plan $p$
$\mathcal{P}_i$	set of plans involving task $i \in \mathcal{I}$
$\mathcal{H}$	set of patterns
$\mathcal{P}_h$	set of plans associated with pattern $h \in \mathcal{H}$
$\mathcal{J}^{\mathcal{H}}(\bar{x})$	set of jobs associated with patterns involved in selection of plans $\bar{x}$
$\mathcal{J}^{\mathcal{R}}$	set of jobs associated with technician unavailability periods
$\mathcal{J}(\bar{x})$	set of jobs associated with selection of plans $\bar{x}$ ( $\mathcal{J}(\bar{x}) = \mathcal{J}^{\mathcal{H}}(\bar{x}) \cup \mathcal{J}^{\mathcal{R}}$ )
$d_t$	day associated with time period $t \in \mathcal{T}$
$l_w, l_i$	location of turbine $w \in \mathcal{W}$ and location where task $i \in \mathcal{I}$ must be performed
$l_r^t$	location of technician $r \in \mathcal{R}$ when he or she is not available during period $t$
$\sigma_{ll'}$	binary parameter equal to 1 if and only if locations $l$ and $l'$ are compatible
$ov(\mathcal{I})$	family of sets of tasks that cannot overlap
$s_i$	skill required for task $i \in \mathcal{I}$
$o_i$	penalty if task $i \in \mathcal{I}$ is postponed
$b_{wi}$	binary parameter equal to 1 if and only if execution of task $i \in \mathcal{I}$ shuts down turbine $w \in \mathcal{W}$ when technicians are working on $i$
$\tilde{b}_{wi}$	binary parameter equal to 1 if and only if execution of task $i \in \mathcal{I}$ shuts down turbine $w \in \mathcal{W}$ during the rest periods it covers
$g_w^t$	profit if turbine $w \in \mathcal{W}$ can produce electricity during period $t \in \mathcal{T}$
$\tilde{g}_w^d$	profit if turbine $w \in \mathcal{W}$ can produce electricity during rest period following day $d \in \mathcal{D}$
$\lambda_{rs}$	binary parameter equal to 1 if and only if technician $r \in \mathcal{R}$ has skill $s \in \mathcal{S}$
$\pi_r^t$	binary parameter equal to 1 if and only if technician $r \in \mathcal{R}$ is available during period $t \in \mathcal{T}$
$i_p$	task involved in plan $p \in \mathcal{P}$
$a_p^t$	binary parameter equal to 1 if and only if task $i_p$ is executed during period $t \in \mathcal{T}$
$S_p, C_p$	start and finish period of plan $p \in \mathcal{P}$
$q_p$	number of required technicians if plan $p \in \mathcal{P}$ is selected
$h_j$	pattern associated with job $j$
$q_j$	number of technicians required by job $j$
$l_j$	location of job $j$
$S_j, C_j$	start and finish period of job $j$
$\mathcal{S}_j$	set of skills such that a technician requires at least one of these skills for job $j$
$\mathcal{R}_j$	set of technicians that can be assigned to job $j$

$$(s_p, l_p, \mathcal{R}_p, b_{wp}, \tilde{b}_{wp}, o_p) = (s_{i_p}, l_{i_p}, \mathcal{R}_{i_p}, b_{wi_p}, \tilde{b}_{wi_p}, o_{i_p})$$

$$(S_h, C_h, a_h^t, s_h, l_h, \mathcal{H}_h, \mathcal{R}_h) = (S_p, C_p, a_p^t, s_p, l_p, \mathcal{H}_{l_{i_p}}, \mathcal{R}_p) \text{ with } p \in \mathcal{P}_h$$

## B Complement: TTA subproblem and complexity

### B.1 Equivalence to the L-coloring problem

To determine the complexity of the TTA subproblem, we prove its equivalence to the L-coloring problem. First, we associate a color  $color_r$  with every technician  $r \in \mathcal{R}$ . For a given solution  $\bar{x}$  to [RMP], we introduce the undirected graph  $\ddot{G}(\bar{x})$  composed of:

- A set of vertices  $\ddot{\mathcal{V}}$ 
  - For each job  $j \in \mathcal{J}(\bar{x})$ , we add  $q_j$  vertices in  $\ddot{\mathcal{V}}$ . Parameter  $j_\nu$  denotes the job associated with  $\nu \in \ddot{\mathcal{V}}$  and  $\ddot{\mathcal{V}}_j$  the set of vertices associated with job  $j$ . Let  $L_\nu$  be the set of colors associated with vertex  $\nu$ :  $L_\nu = \{color_r\}_{r \in \mathcal{R}_j}$ .
- A set of edges  $\ddot{\mathcal{U}}$  defined such that  $\forall \nu_1, \nu_2 \in \ddot{\mathcal{V}}$ :
 
$$(\nu_1, \nu_2) \in \ddot{\mathcal{U}} \Leftrightarrow \nu_1 \neq \nu_2 \wedge \left( (S_{j_{\nu_2}} \leq C_{j_{\nu_1}} \wedge S_{j_{\nu_1}} \leq C_{j_{\nu_2}}) \vee \sigma_{l_{j_{\nu_1}} l_{j_{\nu_2}}} = 0 \right)$$
 (There is an edge between  $\nu_1$  and  $\nu_2$  in  $\ddot{G}(\bar{x})$  if and only if a technician cannot be assigned to both jobs  $j_{\nu_1}$  and  $j_{\nu_2}$  with regard to [C2] and [C3].)

We prove in Proposition B.1 the equivalence between the TTA subproblem and the L-coloring problem in  $\ddot{G}(\bar{x})$ .

**Proposition B.1.** *Let  $\bar{x}$  be a solution to [RMP]. The TTA subproblem for  $\bar{x}$  is equivalent to the L-coloring problem in  $\ddot{G}(\bar{x})$ .*

*Proof.* Assume that we have a feasible solution to the TTA subproblem. This solution directly yields the list  $\mathcal{R}_j^{ass}$  of technicians assigned to every job  $j \in \mathcal{J}(\bar{x})$  (in that solution). We can then build a solution to the L-coloring problem by iterating through the vertices of  $\ddot{G}(\bar{x})$ . Specifically, for each vertex  $\nu \in \ddot{\mathcal{V}}$ , we pick a technician  $r \in \mathcal{R}_{j_\nu}^{ass}$  (and remove it from this set) and color  $\nu$  with  $color_r$ . By construction of the graph, for a job  $j$  the set  $\mathcal{R}_j^{ass}$  becomes empty only when every vertex of  $\ddot{\mathcal{V}}_j$  is colored. The graph is L-colorable since each  $\nu \in \ddot{\mathcal{V}}$  acquires an admissible color in  $L_\nu$ .

Alternatively, assume that we have a solution  $c$  to the L-coloring problem in  $\ddot{G}(\bar{x})$ . This solution directly yields the list  $\mathcal{R}_j^{ass}$  of technicians assigned to every job  $j \in \mathcal{J}(\bar{x})$ . Specifically, for each vertex  $\nu \in \ddot{\mathcal{V}}$ , we add to  $\mathcal{R}_j^{ass}$  the technician  $r \in \mathcal{R}$  such that  $color_r = c(\nu)$ . We then induce the underlying assignment of the technicians to the plans selected in the solution to the RMP. By construction of  $\ddot{G}(\bar{x})$ , the assignments satisfy the DLBIs and match the technician availability periods. This produces a feasible solution to the TTA subproblem.  $\square$

Since the graph coloring problem is a special case of the L-coloring problem, the strong NP-completeness of the former (Jensen and Toft, 2011) implies the strong NP-completeness of the latter. Therefore, the TTA subproblem is NP-complete in the strong sense. Note that in the case of interval graphs, the L-coloring problem remains NP-complete (Biro et al., 1992) although the graph coloring problem becomes polynomial.

## C Examples

### C.1 Example 1

Our first example illustrates how we build the different cuts. We consider a fixed time horizon with 8 periods of identical length ( $\mathcal{T} = \{1, 2, \dots, 8\}$ ) partitioned into two days:  $\mathcal{T}_1 = \{1, 2, 3, 4\}$  and  $\mathcal{T}_2 = \{5, 6, 7, 8\}$ . We have three locations:  $\mathcal{L} = \{l_1, l_2, l_3\}$ . Locations  $l_2$  and  $l_3$  cannot both be visited on the same day (i.e.,  $\sigma_{l_2 l_3} = 0$ ). We do not define any other DLBIs. There are 4 tasks ( $\mathcal{I} = \{A, B, C, D\}$ ), 3 technicians ( $\mathcal{R} = \{r_1, r_2, r_3\}$ ), and 3 skills ( $\mathcal{S} = \{s_1, s_2, s_3\}$ ). The characteristics of the tasks and technicians are defined in Tables 3a and 3b. For simplicity, we introduce only the parameters that are useful for the illustration of the cut generation process.

Table 4 shows a solution to [RMP] in which no cuts have been added. The table gives the start and finish periods of each task as well as the number of technicians required. We refer to this solution as  $\bar{x}$ . Note that  $\bar{x}$  satisfies constraints (6) of the RMP<sup>10</sup>.

Table 3: Data of Example 1.

$\mathcal{I}$	$l_i$	$s_i$	$\mathcal{R}$	$\{s \in \mathcal{S} \mid \lambda_{rs} = 1\}$	unavailability period
A	$l_1$	$s_1$	$r_1$	$\{s_1\}$	at location $l_3$ during period 8
B	$l_2$	$s_1$	$r_2$	$\{s_1, s_2, s_3\}$	–
C	$l_1$	$s_2$	$r_3$	$\{s_1, s_3\}$	–
D	$l_3$	$s_3$			

(a) Characteristics of tasks.
(b) Characteristics of technicians.

Table 4: Solution to restricted master problem for Example 1.

$\mathcal{I}$	Selected plan $p$	$S_p$	$C_p$	$q_p$	$\mathcal{R}_p$
A	$p_A$	2	5	1	$\{r_1, r_2, r_3\}$
B	$p_B$	4	7	2	$\{r_2, r_3\}$
C	$p_C$	7	8	1	$\{r_2\}$
D	$p_D$	1	3	2	$\{r_2, r_3\}$

Let  $u_{r_1}$  be the job associated with the unavailability period of technician  $r_1$ . Observing that  $u_{r_1}$  and  $p_B$  both cover day 2 and are defined at the incompatible locations  $l_2$  and  $l_3$ , we deduce that technician  $r_1$  cannot be assigned to plan  $p_B$  (although he or she has the required skill). In  $\bar{x}$  the selected plans are associated with different patterns, and therefore we simply introduce one job for each plan. Job  $j_A$  refers to plan  $p_A$ , job  $j_B$  refers to plan  $p_B$ , and so on. First, we consider the generation of MCbM cuts. Figure 5 shows  $\check{G}^t(\bar{x})$  for each period of the planning horizon.

For period  $t = 7$ , the maximum flow in  $\hat{G}^7(\bar{x})$  is 2, and this is the maximum cardinality of a b-matching in  $G^7(\bar{x})$ . Since  $q_{j_B} + q_{j_C} = 3$ ,  $\bar{x}$  is an infeasible solution to the overall problem. We then compute the minimum cut in  $\hat{G}^7(\bar{x})$  (see Figure 6).

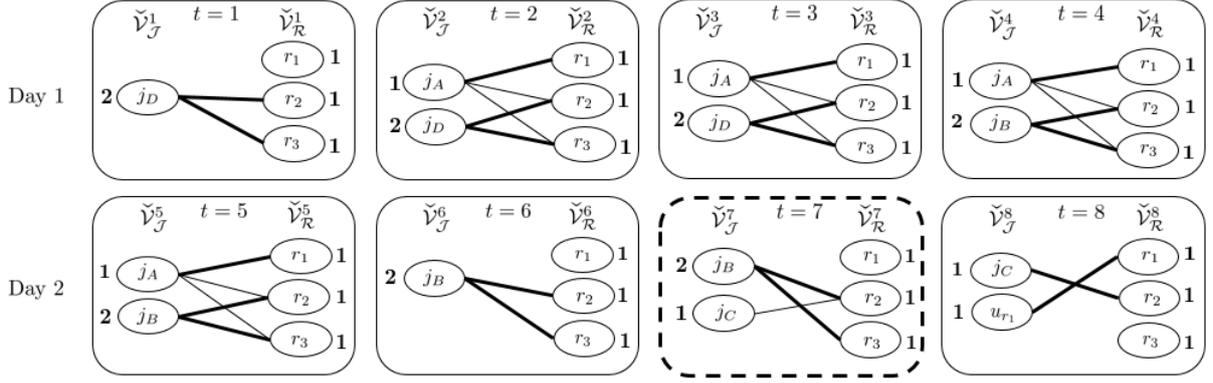
From the general expression (37), we build the MCbM cut (42):

$$2x_{p_B} + 1x_{p_C} \leq 2 \quad (42)$$

Second, we consider the generation of MWC cuts. Figure 7 depicts  $\tilde{G}_{\bar{S}}^d(\bar{x})$  for both days and every subset  $\bar{S} \subseteq \mathcal{S}$  of skills. It shows that the MWC has a weight strictly greater than the number of available technicians in four cases. From the general expression (40), we build the MWC cuts (43) and (44):

$$1x_{p_A} + 2x_{p_B} + 1x_{p_C} \leq 3 \quad (43)$$

<sup>10</sup>For instance, at period 7 we have seven cumulative constraints. Plugging in the values of the variables, we obtain  $2 \leq 3$ ,  $1 \leq 1$ ,  $0 \leq 2$ ,  $3 \leq 3$ ,  $2 \leq 3$ ,  $1 \leq 2$ ,  $3 \leq 3$  when  $\bar{S}$  is respectively equal to  $\{s_1\}, \{s_2\}, \{s_3\}, \{s_1, s_2\}, \{s_1, s_3\}, \{s_2, s_3\}, \{s_1, s_2, s_3\}$ .



N.B.: Bold font indicates a solution to the MCbM problem. A dashed outline indicates that no b-matching with the desired cardinality can be found.

Figure 5: Graphs  $\check{G}^t(\bar{x})$  for Example 1 and different values of  $t \in \mathcal{T}$ .

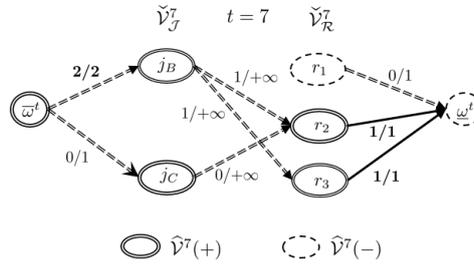
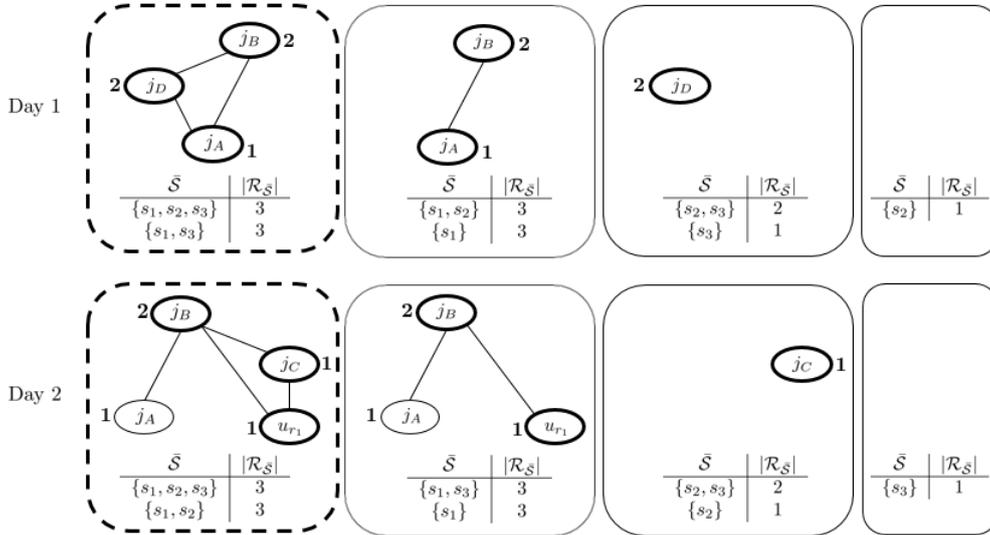


Figure 6: Minimum cut in  $\hat{G}^7(\bar{x})$  for Example 1.



N.B.: Bold font indicates a solution to the MWC problem. A dashed outline indicates that the weight of this clique is strictly greater than the maximum allowed.

Figure 7:  $\tilde{G}_{\mathcal{S}}^d(\bar{x})$  for Example 1.

$$2x_{p_B} + 1x_{p_C} \leq 2 \quad (44)$$

The cut (43) is built from the clique computed in either  $\tilde{G}_{\{s_1, s_2, s_3\}}^1(\bar{x})$  or  $\tilde{G}_{\{s_1, s_3\}}^1(\bar{x})$ , and the cut (44) is built from the clique computed in either  $\tilde{G}_{\{s_1, s_2, s_3\}}^2(\bar{x})$  or  $\tilde{G}_{\{s_1, s_2\}}^2(\bar{x})$ .

Third, we solve  $[SP_2^{LR}(\bar{x})]$  with a commercial solver. Since its optimal value is strictly greater than zero

(equal to 2), we identify the violated BF cut (45) using dual values obtained from the solver:

$$2x_{p_B} + x_{p_C} + 2x_{p_D} \leq 3 \quad (45)$$

Fourth, we directly solve the ILP formulation  $[SP_2(\bar{x})]$ . As for the linear relaxation, the optimal value is 2. We then generate the CB cut (46):

$$x_{p_A} + x_{p_B} + x_{p_C} + x_{p_D} \leq 3 \quad (46)$$

Table 5 gives the infeasible selection of plans discarded by the MCbM, MWC, BF, and CB cuts. We denote a selection of plans using a four dimensional vector where the first, second, third and fourth coordinates refer to the plan selected for task A, task B, task C and task D, respectively. An ellipsis indicates that infeasibility holds for any plan selected at the corresponding coordinate.

Table 5: Infeasible selections of plans discarded

Selections of plans	CB	BF	MCbM	MWC
$(p_A, p_B, p_C, p_D)$	✓	✓	✓	✓
$(p_A, p_B, p_C, \dots)$			✓	✓
$(p_A, p_B, \dots, p_D)$		✓		✓
$(\dots, p_B, p_C, p_D)$		✓	✓	✓
$(\dots, p_B, p_C, \dots)$			✓	✓
$(\dots, p_B, \dots, p_D)$		✓		✓

For the cut (42), we can build a stronger MCbM cut of type (38) as described in Section 3.2.2. We add to the left-hand side of the cut (42) all the patterns covering period 7 to which technician  $r_1$  cannot be assigned (this is the only technician associated with a vertex of set  $\check{V}_{\mathcal{R}}^t(-)$ ).

For the MWC cuts (43) and (44), we can build stronger MWC cuts of type (41) as described in Section 3.2.3. To strengthen MWC cut (43), we consider the subgraph  $G^{sub}(\bar{x})$  of  $G$  that includes the vertices linked to i) patterns at location  $l_1$  covering periods 3 and 4, ii) patterns at location  $l_2$  covering period 4, and iii) patterns at location  $l_3$  covering at least period 2, 3, or 4. A technician cannot be assigned to any of the previous patterns if he or she is assigned to pattern  $p_A$ ,  $p_B$ , or  $p_D$ . We then solve a maximum clique problem in this subgraph, and we add to the left-hand side of the MWC cut (43) all the plans associated with the patterns involved in the maximum clique. We proceed in a similar way for cut (44) by considering the subgraph  $G^{sub}(\bar{x})$  of  $G$  that includes the vertices linked to i) patterns at location  $l_1$  covering periods 7 and 8, ii) patterns at location  $l_2$  covering at least period 7 or 8, and iii) patterns at location  $l_3$  covering at least period 8. A technician cannot be assigned to any of the previous patterns if he or she is assigned to pattern  $p_B$  or  $p_D$  or to the job  $u_{r_1}$ . Again, we solve a maximum clique problem in this subgraph, and we add to the left-hand side of the MWC cut (44) all the plans associated with the patterns involved in the maximum clique. In this example, when building the subgraph, we do not pay attention to the skill associated with the patterns because the MWC cuts (43) and (44) have been computed with  $\bar{S} = \mathcal{S}$ . Otherwise, only the patterns with their skill in  $\bar{S}$  can be added to the left-hand side of the cuts (since the right-hand side of the cuts is based on the total number of technicians with at least one skill of  $\bar{S}$ ).

For Example 1, the approximation of Section 3.2.3 dominates that of Section 3.2.2 (i.e., the MWC cuts are stronger than the MCbM cuts).

## C.2 Example 2

Our second example illustrates a case where the approximation of Section 3.2.2 dominates that of Section 3.2.3 (i.e., the MCbM cuts are stronger than the MWC cuts). We consider a fixed time horizon with 4 periods ( $\mathcal{T} = \{1, 2, 3, 4\}$ ) partitioned into two days:  $\mathcal{T}_1 = \{1, 2\}$  and  $\mathcal{T}_2 = \{3, 4\}$ . We have 1 location ( $\mathcal{L} = \{l\}$ ), 2 tasks

( $\mathcal{I} = \{E, F\}$ ), 1 skill ( $\mathcal{S} = \{s\}$ ), and 2 technicians ( $\mathcal{R} = \{r_4, r_5\}$ ). Technician  $r_2$  is unavailable during periods 1 and 3. The characteristics of the tasks and technicians are summarized in Tables 6a and 6b. Let  $u_{r_5}$  be the jobs associated with the unavailability periods of technician  $r_5$ .

Table 6: Data of Example 2.

(a) Characteristics of tasks.

$\mathcal{I}$	$l_i$	$s_i$
E	$l$	$s$
F	$l$	$s$

(b) Characteristics of technicians.

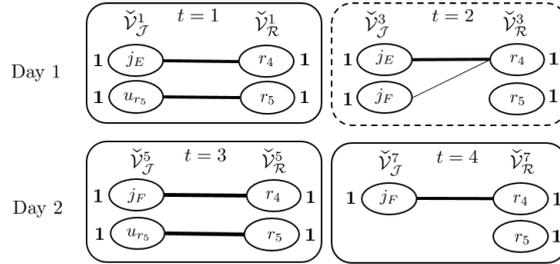
$\mathcal{R}$	$\{s \in \mathcal{S} \mid \lambda_{r,s} = 1\}$	unavailability periods
$r_4$	$\{s\}$	–
$r_5$	$\{s\}$	at location $l$ during periods 1 and 3

Table 7 shows a solution to [RMP] in which no cuts have been added. The table gives the start and finish periods of each task as well as the number of technicians required. Note that  $\bar{x}$  satisfies constraints (6) of the RMP.

Table 7: Solution to restricted master problem for Example 2.

$\mathcal{I}$	Selected plan $p$	$S_p$	$C_p$	$q_p$	$\mathcal{R}_p$
E	$p_E$	1	2	1	$\{r_4\}$
F	$p_F$	2	4	1	$\{r_4\}$

First, we consider the generation of MCbM cuts. Figure 8 shows  $\check{G}^t(\bar{x})$  for each period of the horizon.



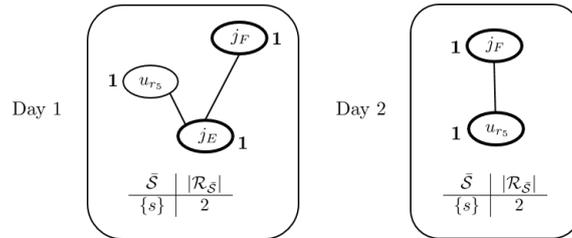
N.B.: Bold font indicates a solution to the MCbM problem. A dashed outline indicates that no b-matching with the desired cardinality can be found.

Figure 8: Graphs  $\check{G}^t(\bar{x})$  for Example 2 and different values of  $t \in \mathcal{T}$ .

Since technician  $r_2$  is unavailable for task B, the maximum cardinality of a b-matching at period 2 is 1, whereas the tasks scheduled during this period require two technicians. The following MCbM cut is produced:

$$x_{p_E} + x_{p_F} \leq 1 \quad (47)$$

Second, we consider the generation of MWC cuts. Figure 9 depicts  $\check{G}_S^d(\bar{x})$  for both days.



N.B.: Bold font indicates a solution to the MWC problem.

Figure 9:  $\check{G}_S^d(\bar{x})$  for Example 2.

Solving the MWC problems does not produce a MWC cut. Therefore, for Example 2, the approximation of Section 3.2.2 dominates that of Section 3.2.3.

### C.3 Example 3

In our third example, the problem-specific approximations do not find a cut although the TTA subproblem is infeasible. We consider a fixed time horizon with 8 periods ( $\mathcal{T} = \{1, \dots, 8\}$ ) partitioned into two days:  $\mathcal{T}_1 = \{1, 2, 3, 4\}$  and  $\mathcal{T}_2 = \{5, 6, 7, 8\}$ . We have 1 location ( $\mathcal{L} = \{l\}$ ), 4 tasks ( $\mathcal{I} = \{G, H, I, J\}$ ), 3 skills ( $\mathcal{S} = \{s_6, s_7, s_8\}$ ), and 2 technicians ( $\mathcal{R} = \{r_6, r_7\}$ ). The characteristics of the tasks and technicians are summarized in Tables 8a and 8b.

Table 8: Data of Example 3.

(a) Characteristics of tasks.

$\mathcal{I}$	$l_i$	$s_i$
G	$l$	$s_6$
H	$l$	$s_6$
I	$l$	$s_7$
J	$l$	$s_8$

(b) Characteristics of technicians.

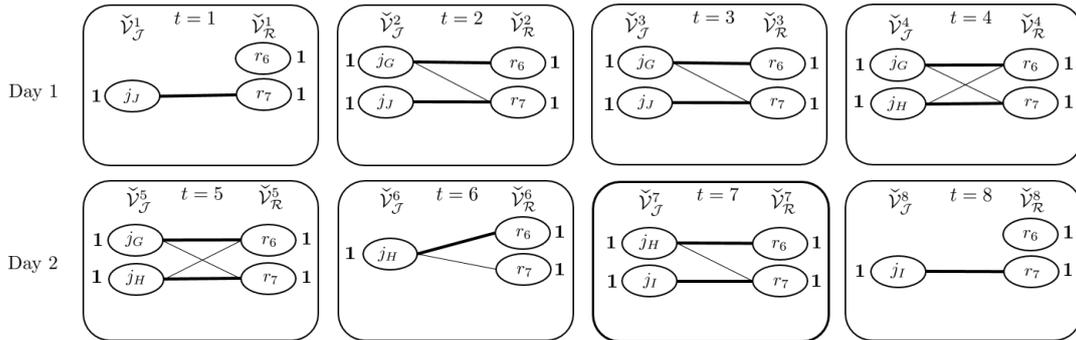
$\mathcal{R}$	$\{s \in \mathcal{S} \mid \lambda_{rs} = 1\}$	unavailability periods
$r_6$	$\{s_6\}$	–
$r_7$	$\{s_6, s_7, s_8\}$	–

Table 9 shows a solution [RMP] in which no cuts have been added. The table gives the start and finish periods of each task as well as the number of technicians required. Again, it is easy to verify that  $\bar{x}$  satisfies constraints (6) of the RMP.

Table 9: Solution to restricted master problem for Example 3.

$\mathcal{I}$	Selected plan $p$	$S_p$	$C_p$	$q_p$	$\mathcal{R}_p$
G	$p_G$	2	5	1	$\{r_6, r_7\}$
H	$p_H$	4	7	1	$\{r_6, r_7\}$
I	$p_I$	7	8	1	$\{r_7\}$
J	$p_J$	1	3	1	$\{r_7\}$

First, we consider the generation of MCbM cuts. Figure 10 shows  $\check{G}^t(\bar{x})$  for each period of the horizon. We



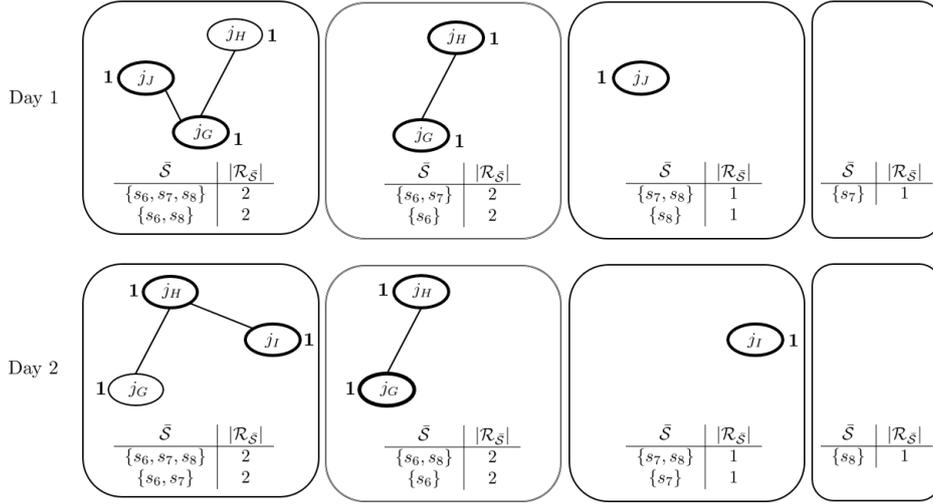
N.B.: Bold font indicates a solution to the MCbM problem.

Figure 10: Graphs  $\check{G}^t(\bar{x})$  for Example 3 and different values of  $t \in \mathcal{T}$ .

observe that solving the MCbM problems does not produce a MCbM cut.

Second, we consider the generation of MWC cuts. Figure 11 depicts  $\check{G}_S^d(\bar{x})$  for both days. Similarly, no MWC cuts are produced from this approximation.

However, it is easy to see that the TTA subproblem is infeasible. First, technician  $r_7$  must perform tasks I and J. Second, tasks G and H cannot be performed by the same technician since they overlap. The same holds for tasks G and J and for tasks H and I. The TTA subproblem is therefore infeasible since technician  $r_7$  cannot perform either tasks G, I, and J, or tasks H, I and J.



N.B.: Bold font indicates a solution to the MWC problem.

Figure 11:  $\tilde{G}_S^d(\bar{x})$  for Example 3.

The solution of  $[SP_2^{LR}(\bar{x})]$  gives an optimal value strictly greater than zero (equal to 1). We then identify the violated BF cut (48) (which has here the same expression as a CB cut):

$$x_{p_G} + x_{p_H} + x_{p_I} + x_{p_J} \leq 3 \quad (48)$$

In this example no MCbM or MWC cuts are identified although the TTA subproblem is infeasible.

## D Detailed experimental results

Table 10: Average number of cuts generated in B&C approach.

Family	All	CB	Other cuts		
			BF	MCbM	MWC
10.2.1_20_A	21	0	0	2	19
10.2.1_20_B	10	0	0	1	9
10.2.1_40_A	36	0	0	0.8	35
10.2.1_40_B	12	0	0	0.6	12
10.2.3_20_A	153	0	94	9	50
10.2.3_20_B	28	0	7	6	15
10.2.3_40_A	66	0	16	9	41
10.2.3_40_B	17	0	0.2	2	15
20.2.1_40_A	67	0	0	1	66
20.2.1_40_B	28	0	0	1	27
20.2.1_80_A	105	0	0	3	101
20.2.1_80_B	18	0	0	1	17
20.2.3_40_A	144	0	17	7	120
20.2.3_40_B	40	0	5	2	32
20.2.3_80_A	118	0	6	5	107
20.2.3_80_B	31	0	0.6	0.8	30
20.4.1_20_A	94	0	0	5	90
20.4.1_20_B	26	0	0	3	23
20.4.1_40_A	202	0	5	12	185
20.4.1_40_B	57	0	0	3	54
20.4.3_20_A	314	0	34	16	263
20.4.3_20_B	84	0	21	13	50
20.4.3_40_A	392	0	38	26	329
20.4.3_40_B	96	0	20	10	66
40.4.1_40_A	229	0	0	3	226
40.4.1_40_B	91	0	0	2	88
40.4.1_80_A	480	0	0	5	476
40.4.1_80_B	151	0	0	4	146
40.4.3_40_A	672	0	65	19	588
40.4.3_40_B	144	0	9	6	129
40.4.3_80_A	1,106	0	102	33	971
40.4.3_80_B	184	0	5	5	174

We write “a\_b\_c\_d\_e\_f” where “a\_b\_c\_d\_e” indicates the family and “f” the instance number. The column “Obj” gives the objective of the best solution computed by any of our methods; “Time” gives the computational time (in seconds); “Relax” gives the value of the linear relaxations of  $[P_1]$  and  $[P_2]$ ; “UB” gives the upper bound used to compute the gap. For reference, the column “CPLNS” reports the best solution found by Froger et al. (2017) with a CPLNS approach.

Table 11: Detailed results for each instance

Instance	$[P_1]$			$[P_2]$			B&C		CPLNS	UB
	Time	Obj	Relax	Time	Obj	Relax	Time	Obj		
10_2.1.20_A.1	86	<b>47923.7</b>	48132.64	110	<b>47923.7</b>	48144.54	1	<b>47923.7</b>	<b>47923.7</b>	47923.7
10_2.1.20_A.2	2475	<b>43231.5</b>	44326.76	430	<b>43231.5</b>	44343.24	13	<b>43231.5</b>	42474.1	43231.5
10_2.1.20_A.3	67	<b>58597.6</b>	62152.91	92	<b>58597.6</b>	62183.79	1	<b>58597.6</b>	57673.5	58597.6
10_2.1.20_A.4	8	<b>63014.9</b>	63179.87	7	<b>63014.9</b>	63167.33	1	<b>63014.9</b>	62952.9	63014.9
10_2.1.20_A.5	354	<b>42646.6</b>	43157.29	120	<b>42646.6</b>	43142.64	10	<b>42646.6</b>	<b>42646.6</b>	42646.6
10_2.1.20_B.1	1	<b>50153.2</b>	50239.27	1	<b>50153.2</b>	50245.33	1	<b>50153.2</b>	<b>50153.2</b>	50153.2
10_2.1.20_B.2	34	<b>66808.4</b>	69890.75	177	<b>66808.4</b>	69890.75	1	<b>66808.4</b>	<b>66808.4</b>	66808.4
10_2.1.20_B.3	10	<b>39222.6</b>	39501.82	2	<b>39222.6</b>	39499.54	1	<b>39222.6</b>	<b>39222.6</b>	39222.6
10_2.1.20_B.4	14	<b>66168.2</b>	66630.57	5	<b>66168.2</b>	66554.25	2	<b>66168.2</b>	<b>66168.2</b>	66168.2
10_2.1.20_B.5	1	<b>96760.9</b>	96819.73	1	<b>96760.9</b>	96878.27	0	<b>96760.9</b>	<b>96760.9</b>	96760.9
10_2.1.40_A.1	24	<b>120976.0</b>	122507.30	7	<b>120976.0</b>	122507.30	2	<b>120976.0</b>	120900.9	120976.0
10_2.1.40_A.2	3602	<b>140280.1</b>	140724.40	10800	140280.1	140767.07	4	<b>140280.1</b>	140265.2	140280.1
10_2.1.40_A.3	10800	129716.8	130029.78	10800	129701.7	130043.32	15	<b>129716.8</b>	129671.4	129716.8
10_2.1.40_A.4	10800	117036.9	117461.17	10800	117040.9	117517.68	7	<b>117058.6</b>	116991.3	117058.6
10_2.1.40_A.5	10800	119352.6	120463.36	10800	119352.6	120463.36	4	<b>119352.6</b>	118932.1	119352.6
10_2.1.40_B.1	662	<b>123614.6</b>	123877.36	225	<b>123614.6</b>	123860.80	2	<b>123614.6</b>	<b>123614.6</b>	123614.6
10_2.1.40_B.2	149	<b>145580.0</b>	145976.50	11	<b>145580.0</b>	146025.68	2	<b>145580.0</b>	<b>145580.0</b>	145580.0
10_2.1.40_B.3	142	<b>116211.1</b>	116682.73	300	<b>116211.1</b>	116682.73	1	<b>116211.1</b>	<b>116211.1</b>	116211.1
10_2.1.40_B.4	64	<b>133112.9</b>	133606.82	3	<b>133112.9</b>	133624.16	1	<b>133112.9</b>	133092.9	133112.9
10_2.1.40_B.5	9	<b>132110.9</b>	132339.03	66	<b>132110.9</b>	132339.03	1	<b>132110.9</b>	132106.9	132110.9
10_2.3.20_A.1	9228	<b>55731.1</b>	59032.02	6668	<b>55731.1</b>	59057.12	696	<b>55731.1</b>	55077.7	55731.1
10_2.3.20_A.2	766	<b>43172.7</b>	44543.94	594	<b>43172.7</b>	44639.25	24	<b>43172.7</b>	43043.7	43172.7
10_2.3.20_A.3	817	<b>34143.2</b>	34901.19	348	<b>34143.2</b>	34939.55	13	<b>34143.2</b>	34035.5	34143.2
10_2.3.20_A.4	2358	<b>82928.1</b>	83613.29	256	<b>82928.1</b>	83591.71	77	<b>82928.1</b>	82919.3	82928.1
10_2.3.20_A.5	4	<b>72423.5</b>	73148.79	3	<b>72423.5</b>	73139.52	2	<b>72423.5</b>	<b>72423.5</b>	72423.5
10_2.3.20_B.1	16	<b>76852.7</b>	77041.72	11	<b>76852.7</b>	77101.17	2	<b>76852.7</b>	<b>76852.7</b>	76852.7
10_2.3.20_B.2	3	<b>66410.9</b>	66861.85	3	<b>66410.9</b>	66863.66	1	<b>66410.9</b>	<b>66410.9</b>	66410.9
10_2.3.20_B.3	14	<b>51749.9</b>	52105.77	17	<b>51749.9</b>	52105.77	2	<b>51749.9</b>	<b>51749.9</b>	51749.9
10_2.3.20_B.4	1	<b>37576.4</b>	37703.78	1	<b>37576.4</b>	37703.78	0	<b>37576.4</b>	<b>37576.4</b>	37576.4
10_2.3.20_B.5	123	<b>69330.7</b>	69648.50	56	<b>69330.7</b>	69648.50	5	<b>69330.7</b>	69306.3	69330.7
10_2.3.40_A.1	2152	<b>91031.9</b>	91460.86	10094	<b>91031.9</b>	91491.49	3	<b>91031.9</b>	90948.0	91031.9
10_2.3.40_A.2	5752	<b>76895.1</b>	77478.31	3064	<b>76895.1</b>	77538.06	10	<b>76895.1</b>	76485.6	76895.1
10_2.3.40_A.3	6370	<b>107914.6</b>	108588.55	3383	<b>107914.6</b>	108592.47	47	<b>107914.6</b>	<b>107914.6</b>	107914.6
10_2.3.40_A.4	268	<b>95022.1</b>	97279.62	217	<b>95022.1</b>	97279.62	5	<b>95022.1</b>	94948.4	95022.1
10_2.3.40_A.5	1560	<b>118961.0</b>	120371.75	3225	<b>118961.0</b>	120371.75	21	<b>118961.0</b>	118506.5	118961.0
10_2.3.40_B.1	18	<b>79260.6</b>	80307.68	53	<b>79260.6</b>	80307.68	1	<b>79260.6</b>	<b>79260.6</b>	79260.6
10_2.3.40_B.2	118	<b>104039.9</b>	104593.14	48	<b>104039.9</b>	104593.14	1	<b>104039.9</b>	<b>104039.9</b>	104039.9
10_2.3.40_B.3	17	<b>96649.2</b>	97001.79	26	<b>96649.2</b>	97001.99	2	<b>96649.2</b>	96617.2	96649.2
10_2.3.40_B.4	263	<b>129859.7</b>	130081.73	69	<b>129859.7</b>	130120.73	1	<b>129859.7</b>	129851.7	129859.7
10_2.3.40_B.5	484	<b>127246.9</b>	127679.46	1279	<b>127246.9</b>	127703.46	1	<b>127246.9</b>	<b>127246.9</b>	127246.9
20_2.1.40_A.1	8357	<b>326654.6</b>	327143.68	1927	<b>326654.6</b>	327178.62	18	<b>326654.6</b>	326357.8	326654.6
20_2.1.40_A.2	10800	282515.6	298001.39	10800	290902.7	298070.51	120	<b>296421.1</b>	294855.6	296421.1
20_2.1.40_A.3	10800	438999.1	450484.82	10800	439346.8	450537.67	899	<b>449480.1</b>	448793.2	449480.1
20_2.1.40_A.4	10800	308698.8	309571.25	10800	308717.2	309559.45	105	<b>308726.8</b>	308249.2	308726.8
20_2.1.40_A.5	7792	<b>201315.3</b>	202357.52	3789	<b>201315.3</b>	202384.96	10	<b>201315.3</b>	201217.3	201315.3
20_2.1.40_B.1	429	<b>236502.3</b>	237806.16	343	<b>236502.3</b>	237813.78	4	<b>236502.3</b>	236478.2	236502.3
20_2.1.40_B.2	47	<b>309385.9</b>	311197.70	25	<b>309385.9</b>	311225.39	2	<b>309385.9</b>	309384.5	309385.9
20_2.1.40_B.3	418	<b>348011.8</b>	348296.62	419	<b>348011.8</b>	348296.74	2	<b>348011.8</b>	347965.7	348011.8
20_2.1.40_B.4	32	<b>317274.1</b>	317570.09	22	<b>317274.1</b>	317614.07	3	<b>317274.1</b>	317223.4	317274.1
20_2.1.40_B.5	10800	283812.3	284821.72	9580	<b>283812.3</b>	284874.76	9	<b>283812.3</b>	283686.9	283812.3
20_2.1.80_A.1	10800	-120636.7	765303.50	10800	-120636.7	765335.54	294	<b>764294.5</b>	762965.0	764294.5
20_2.1.80_A.2	10800	-510542.4	526812.51	10800	-510542.4	526907.73	648	<b>524862.7</b>	521644.7	524862.7
20_2.1.80_A.3	10800	-139492.8	618782.86	10800	-139492.8	618818.27	65	<b>617111.2</b>	614963.3	617111.2
20_2.1.80_A.4	10800	-206213.4	851621.30	10800	838356.9	851665.51	195	<b>851001.1</b>	846333.1	851001.1
20_2.1.80_A.5	10800	-642506.8	564660.32	10800	-642506.8	564810.52	10800	562342.3	532015.6	562452.8
20_2.1.80_B.1	10800	-143552.3	769706.77	4124	<b>769511.4</b>	769727.29	6	<b>769511.4</b>	769316.7	769511.4
20_2.1.80_B.2	1182	<b>706851.6</b>	707266.37	5599	<b>706851.6</b>	707308.20	6	<b>706851.6</b>	706455.9	706851.6
20_2.1.80_B.3	3052	<b>688801.4</b>	689291.04	10800	-193415.9	689368.98	5	<b>688801.4</b>	688509.3	688801.4
20_2.1.80_B.4	220	<b>699114.8</b>	699272.01	1745	<b>699114.8</b>	699305.94	3	<b>699114.8</b>	699007.4	699114.8
20_2.1.80_B.5	10800	690793.0	692218.43	10800	673941.0	692205.58	4	<b>690799.9</b>	690626.2	690799.9
20_2.3.40_A.1	10800	282056.9	296812.75	10800	288845.4	296990.16	10800	289644.5	289408.5	295616.2
20_2.3.40_A.2	322	<b>282694.9</b>	283227.65	2133	<b>282694.9</b>	283241.57	3	<b>282694.9</b>	282629.9	282694.9
20_2.3.40_A.3	10800	347369.9	348060.75	9455	<b>347369.9</b>	348067.21	71	<b>347369.9</b>	347168.9	347369.9
20_2.3.40_A.4	10800	284474.9	285630.36	5012	<b>284963.0</b>	285658.56	68	<b>284963.0</b>	284626.5	284963.0

20.2.3.40.A.5	10800	241699.2	242685.02	10800	241593.6	242695.90	20	<b>241699.2</b>	241698.8	241699.2
20.2.3.40.B.1	34	<b>270164.6</b>	270660.46	68	<b>270164.6</b>	270660.46	3	<b>270164.6</b>	270112.9	270164.6
20.2.3.40.B.2	1458	<b>389120.4</b>	389942.48	631	<b>389120.4</b>	389896.19	6	<b>389120.4</b>	389104.4	389120.4
20.2.3.40.B.3	377	<b>288338.5</b>	288574.50	51	<b>288338.5</b>	288572.71	2	<b>288338.5</b>	288298.0	288338.5
20.2.3.40.B.4	10	<b>260249.9</b>	260770.92	22	<b>260249.9</b>	260773.65	1	<b>260249.9</b>	<b>260249.9</b>	260249.9
20.2.3.40.B.5	3	<b>294339.9</b>	295485.41	4	<b>294339.9</b>	295493.55	2	<b>294339.9</b>	<b>294339.9</b>	294339.9
20.2.3.80.A.1	10800	-245333.7	670377.98	10800	-245333.7	670420.35	27	<b>668400.8</b>	666513.1	668400.8
20.2.3.80.A.2	10800	-646354.9	474659.12	10800	-646354.9	474669.09	45	<b>472555.6</b>	470761.3	472555.6
20.2.3.80.A.3	10800	-143037.0	582204.75	10800	564090.6	582236.38	31	<b>581701.0</b>	580640.9	581701.0
20.2.3.80.A.4	10800	-491302.5	644291.99	10800	-491302.5	644453.48	142	<b>641505.5</b>	639633.2	641505.5
20.2.3.80.A.5	10800	495464.5	506178.65	10800	495886.9	506204.91	11	<b>504938.6</b>	504337.1	504938.6
20.2.3.80.B.1	453	<b>725872.5</b>	725933.70	357	<b>725872.5</b>	725929.20	5	<b>725872.5</b>	725786.2	725872.5
20.2.3.80.B.2	4591	<b>845843.3</b>	846079.28	10800	845805.2	846123.70	5	<b>845843.3</b>	845613.7	845843.3
20.2.3.80.B.3	2512	<b>767537.5</b>	767750.22	2869	<b>767537.5</b>	767767.20	4	<b>767537.5</b>	767445.5	767537.5
20.2.3.80.B.4	9388	<b>521195.4</b>	523010.25	10800	-178275.7	523041.27	7	<b>521195.4</b>	520344.1	521195.4
20.2.3.80.B.5	132	<b>726925.1</b>	727061.95	4143	<b>726925.1</b>	727061.95	4	<b>726925.1</b>	726653.1	726925.1
20.4.1.20.A.1	10800	47987.0	52907.09	10800	48656.4	52960.85	10800	49461.9	49481.2	49513.9
20.4.1.20.A.2	10800	48781.9	51762.74	10800	48781.6	51990.00	10800	48781.9	48869.4	49568.5
20.4.1.20.A.3	10800	64055.4	66096.07	10800	64055.4	66030.63	1422	<b>64055.4</b>	63981.3	64055.4
20.4.1.20.A.4	10800	42344.5	45992.13	10800	42862.1	46035.08	2108	<b>42862.1</b>	42662.5	42862.1
20.4.1.20.A.5	10800	59136.2	61492.15	10800	58946.7	61632.24	1616	<b>59246.5</b>	<b>59246.5</b>	59246.5
20.4.1.20.B.1	460	<b>44561.0</b>	44905.19	287	<b>44561.0</b>	44905.19	2	<b>44561.0</b>	<b>44561.0</b>	44561.0
20.4.1.20.B.2	103	<b>43918.7</b>	44191.75	167	<b>43918.7</b>	44203.55	2	<b>43918.7</b>	43905.3	43918.7
20.4.1.20.B.3	260	<b>51457.1</b>	51899.49	1451	<b>51457.1</b>	51921.17	2	<b>51457.1</b>	<b>51457.1</b>	51457.1
20.4.1.20.B.4	70	<b>53580.9</b>	54000.58	112	<b>53580.9</b>	54000.58	1	<b>53580.9</b>	<b>53580.9</b>	53580.9
20.4.1.20.B.5	126	<b>53333.8</b>	53748.23	7	<b>53333.8</b>	53768.19	1	<b>53333.8</b>	<b>53333.8</b>	53333.8
20.4.1.40.A.1	10800	-109313.9	123878.41	10800	112850.5	124449.38	10800	118901.9	114794.0	121162.4
20.4.1.40.A.2	10800	-49260.5	85652.91	10800	-49260.5	85903.46	211	<b>81619.7</b>	77998.1	81619.7
20.4.1.40.A.3	10800	-100733.0	135769.89	10800	126266.4	135959.91	10800	131453.8	127332.7	133246.7
20.4.1.40.A.4	10800	-38989.6	118859.30	10800	101566.7	119052.40	2961	<b>118136.0</b>	113381.5	118136.0
20.4.1.40.A.5	10800	-76192.3	87310.73	10800	76808.3	87628.28	10800	83700.1	81028.2	84085.2
20.4.1.40.B.1	10800	118527.9	120195.16	10800	119715.0	120216.86	5	<b>119759.5</b>	119715.0	119759.5
20.4.1.40.B.2	10800	-64217.5	107685.17	10800	106829.8	107691.47	5	<b>106892.4</b>	106847.9	106892.4
20.4.1.40.B.3	10800	-27751.0	90510.51	10800	-27751.0	90537.91	5	<b>90030.4</b>	90005.8	90030.4
20.4.1.40.B.4	5208	<b>110129.3</b>	110272.20	1968	<b>110129.3</b>	110272.20	3	<b>110129.3</b>	110073.1	110129.3
20.4.1.40.B.5	10800	108772.9	112878.84	10800	107242.8	112898.95	14	<b>112119.0</b>	111797.8	112119.0
20.4.3.20.A.1	10800	44955.7	46845.73	6114	<b>45634.2</b>	46921.33	578	<b>45634.2</b>	44689.3	45634.2
20.4.3.20.A.2	10800	70061.6	70829.57	8884	<b>70068.8</b>	70809.22	46	<b>70068.8</b>	70063.3	70068.8
20.4.3.20.A.3	10800	37640.0	39612.86	3596	<b>37731.3</b>	39684.74	286	<b>37731.3</b>	<b>37731.3</b>	37731.3
20.4.3.20.A.4	10800	42641.5	46857.96	10800	43347.5	46876.91	10800	43495.1	43503.0	44405.9
20.4.3.20.A.5	10800	41443.8	42943.34	1428	<b>41443.8</b>	43006.56	39	<b>41443.8</b>	41053.8	41443.8
20.4.3.20.B.1	59	<b>54871.0</b>	55668.25	15	<b>54871.0</b>	55656.46	2	<b>54871.0</b>	54836.0	54871.0
20.4.3.20.B.2	10800	37999.8	39592.83	445	<b>37999.8</b>	39508.00	37	<b>37999.8</b>	37963.5	37999.8
20.4.3.20.B.3	11	<b>64591.5</b>	65116.48	7	<b>64591.5</b>	65132.93	1	<b>64591.5</b>	<b>64591.5</b>	64591.5
20.4.3.20.B.4	5	<b>58536.6</b>	59308.20	5	<b>58536.6</b>	59308.20	1	<b>58536.6</b>	<b>58536.6</b>	58536.6
20.4.3.20.B.5	135	<b>57558.6</b>	57825.86	92	<b>57558.6</b>	57820.35	4	<b>57558.6</b>	<b>57558.6</b>	57558.6
20.4.3.40.A.1	10800	-26612.6	106530.91	10800	102003.7	106577.25	10800	105890.8	102926.1	106024.7
20.4.3.40.A.2	10800	-10730.2	78648.48	10800	75425.5	78853.28	10800	77307.4	75767.5	78065.6
20.4.3.40.A.3	10800	-59316.7	118198.21	10800	110838.5	118238.76	10800	114914.3	111723.3	116578.8
20.4.3.40.A.4	10800	105180.7	107367.81	10800	105361.8	107454.78	10800	105674.4	105311.4	106650.4
20.4.3.40.A.5	10800	-83677.5	120197.35	10800	104767.0	120281.66	8888	<b>118083.7</b>	115428.8	118083.7
20.4.3.40.B.1	6654	<b>81090.2</b>	83105.04	2280	<b>81090.2</b>	83109.28	5	<b>81090.2</b>	80980.2	81090.2
20.4.3.40.B.2	10800	118287.3	118803.83	10800	118482.7	118812.21	37	<b>118492.3</b>	118279.6	118492.3
20.4.3.40.B.3	10800	91969.5	96630.34	10800	95878.3	96656.68	5	<b>95923.1</b>	95755.5	95923.1
20.4.3.40.B.4	5570	<b>91978.7</b>	92386.40	1451	<b>91978.7</b>	92354.62	3	<b>91978.7</b>	91836.3	91978.7
20.4.3.40.B.5	10800	133877.9	134318.43	2592	<b>133891.9</b>	134334.63	6	<b>133891.9</b>	133674.0	133891.9
40.4.1.40.A.1	10800	-223958.2	293282.94	10800	267024.8	293338.01	10800	291212.1	280389.9	292004.5
40.4.1.40.A.2	10800	-69237.6	274973.56	10800	-69237.6	275164.82	10800	266785.7	258794.0	273346.0
40.4.1.40.A.3	10800	109697.2	312651.05	10800	307513.4	313164.38	10800	307971.9	304556.6	311015.0
40.4.1.40.A.4	10800	-61805.7	317869.63	10800	307987.9	317971.74	10800	308696.3	300912.1	316997.4
40.4.1.40.A.5	10800	-219076.0	308390.29	10800	254487.8	308446.37	10800	294894.4	295229.3	307423.2
40.4.1.40.B.1	10800	4384.1	279164.07	10800	272357.6	279167.90	35	<b>278713.3</b>	278656.5	278713.3
40.4.1.40.B.2	10800	279973.2	281936.97	10800	280562.8	281971.63	51	<b>280591.4</b>	280151.4	280591.4
40.4.1.40.B.3	10800	-131151.4	248231.58	10800	228693.1	248268.37	20	<b>246304.2</b>	246187.1	246304.2
40.4.1.40.B.4	10800	283062.2	292438.80	10800	291760.5	292599.69	11	<b>292013.3</b>	291903.9	292013.3
40.4.1.40.B.5	10800	16422.8	251464.68	10800	238763.1	251496.11	36	<b>250168.4</b>	249950.5	250168.4
40.4.1.80.A.1	10800	-2183.9	474617.19	10800	-2183.9	474764.49	10800	466741.3	460047.6	472247.3
40.4.1.80.A.2	10800	-547760.9	461601.00	10800	-547760.9	461786.81	10800	447970.1	434460.7	459664.8
40.4.1.80.A.3	10800	-140332.6	532034.11	10800	-140332.6	532294.37	10800	521855.5	513605.5	529537.8

40_4.1.80_A.4	10800	-26524.3	492236.90	10800	-26524.3	492348.37	10800	484061.0	471742.9	489296.1
40_4.1.80_A.5	10800	-188927.8	585715.26	10800	-188927.8	585823.96	10800	575893.2	557891.9	583819.6
40_4.1.80_B.1	10800	-345453.3	583672.22	10800	-345453.3	583778.56	48	<b>582793.9</b>	581935.8	582793.9
40_4.1.80_B.2	10800	-546013.4	576065.39	10800	-546013.4	576139.58	84	<b>575497.3</b>	574056.3	575497.3
40_4.1.80_B.3	10800	-275997.1	484446.76	10800	-275997.1	484506.84	95	<b>483414.6</b>	483186.7	483414.6
40_4.1.80_B.4	10800	-122460.7	473218.48	10800	-122460.7	473236.56	124	<b>470190.2</b>	469552.6	470190.2
40_4.1.80_B.5	10800	-151361.3	486516.39	10800	-151361.3	486617.21	94	<b>483976.7</b>	482951.6	483976.7
40_4.3.40_A.1	10800	-64605.8	310319.63	10800	291347.8	310569.51	10800	309526.8	301722.8	309611.4
40_4.3.40_A.2	10800	-33791.5	245342.77	10800	231878.4	245385.31	10800	238762.7	238643.5	244515.6
40_4.3.40_A.3	10800	-79668.2	226478.59	10800	210942.5	226498.79	10800	218268.9	223248.8	224635.4
40_4.3.40_A.4	10800	292882.9	306581.70	10800	305292.1	306655.94	10800	305446.3	304228.5	305647.5
40_4.3.40_A.5	10800	-7542.9	269197.24	10800	255191.8	269318.91	10800	261872.6	261210.8	267390.8
40_4.3.40_B.1	10800	43552.6	246844.73	10800	246127.9	246881.72	94	<b>246324.8</b>	246207.0	246324.8
40_4.3.40_B.2	10800	-34033.5	339296.32	10800	331008.1	339341.88	23	<b>338883.1</b>	338688.3	338883.1
40_4.3.40_B.3	10800	264600.3	266023.65	2940	<b>264757.6</b>	266027.49	24	<b>264757.6</b>	264620.8	264757.6
40_4.3.40_B.4	10800	-102894.4	309896.00	1297	<b>309635.3</b>	309937.39	12	<b>309635.3</b>	309527.6	309635.3
40_4.3.40_B.5	10800	67650.3	354114.17	10800	353456.3	354201.24	27	<b>353594.3</b>	353360.8	353594.3
40_4.3.80_A.1	10800	5373.3	673267.68	10800	5373.3	673286.83	10800	671568.0	656176.8	671822.6
40_4.3.80_A.2	10800	92998.3	501865.45	10800	453900.7	502000.59	10800	489859.7	484781.1	499156.2
40_4.3.80_A.3	10800	86907.5	640771.78	10800	86907.5	641024.83	10800	618669.5	620214.6	639179.3
40_4.3.80_A.4	10800	-554530.3	548796.60	10800	-554530.3	549279.37	10800	518208.5	492096.2	544052.3
40_4.3.80_A.5	10800	-275107.2	801164.54	10800	-275107.2	801352.10	10800	787909.8	758956.1	799603.4
40_4.3.80_B.1	10800	-282931.6	577234.98	10800	-282931.6	577252.25	76	<b>576215.4</b>	575695.0	576215.4
40_4.3.80_B.2	10800	-94904.2	521086.76	10800	-94904.2	521135.73	77	<b>519412.8</b>	518592.2	519412.8
40_4.3.80_B.3	10800	-93231.0	442772.02	10800	-93231.0	442779.38	155	<b>440853.6</b>	440313.5	440853.6
40_4.3.80_B.4	10800	3281.7	558883.69	10800	3281.7	558955.03	42	<b>558032.0</b>	557478.7	558032.0
40_4.3.80_B.5	10800	51078.3	595237.42	10800	51078.3	595284.73	78	<b>594651.8</b>	593991.5	594651.8

Note: Bold font indicates optimal solutions (i.e., the gap is below a threshold fixed to  $10^{-6}$ ).