Journal Pre-proof

A decomposed branch-and-price procedure for integrating demand planning in personnel staffing problems

M. Van Den Eeckhout, M. Vanhoucke, B. Maenhout

PII: S0377-2217(19)30648-4
DOI: <https://doi.org/10.1016/j.ejor.2019.07.069>
Reference: EOR 15970

To appear in: *European Journal of Operational Research*

Received date: 19 May 2018

Accepted date: 31 July 2019

Please cite this article as: M. Van Den Eeckhout, M. Vanhoucke, B. Maenhout, A decomposed branch-and-price procedure for integrating demand planning in personnel staffing problems, *European Journal of Operational Research* (2019), doi: <https://doi.org/10.1016/j.ejor.2019.07.069>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2019 Published by Elsevier B.V.

Highlights

- Project staffing with discrete time/resource trade-offs and calendar constraints
- A cut-branch-and-price procedure is proposed
- Decomposition into subproblems with a specific staffing composition.
- Dedicated cuts include personnel information in the workload pricing problem

JOURNAL PRE-PROOF

A decomposed branch-and-price procedure for integrating demand planning in personnel staffing problems

M. Van Den Eeckhout¹, M. Vanhoucke^{1,2,3}, and B. Maenhout¹

¹Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Gent (Belgium), Mick.vanDenEeckhout@ugent.be, Broos.Maenhout@ugent.be, Mario.Vanhoucke@ugent.be

²Technology and Operations Management Area, Vlerick Business School, Belgium

³UCL School of Management, University College London, UK

Abstract

The personnel staffing problem calculates the required workforce size and is determined by constructing a baseline personnel roster that assigns personnel members to duties in order to cover certain staffing requirements. In this research, we incorporate the planning of the duty demand in the staff scheduling problem in order to lower the staffing costs. More specifically, the demand originates from a project scheduling problem with discrete time/resource trade-offs, which embodies additional flexibility as activities can be executed in different modes. In order to tackle this integrated problem, we propose a decomposed branch-and-price procedure. A tight lower and upper bound are calculated using a problem formulation that models the project scheduling constraints and the time-related resource scheduling constraints implicitly in the decision variables. Based upon these bounds, the strategic problem is decomposed into multiple tactical subproblems with a fixed workforce size and an optimal solution is searched for each subproblem via branch-and-price. Fixing the workforce size in a subproblem facilitates the definition of resource capacity cuts, which limit the set of eligible project schedules, accelerating the column generation procedure in each node. In addition, in order to find the optimal integer solution, we propose a specific search strategy based upon the lower bound and dedicated rules to branch upon the workload generated by a project schedule. The computational results show that applying the proposed search space decomposition and the inclusion of resource capacity cuts lead to a well-performing procedure outperforming different other heuristic and exact methodologies.

Keywords: Project scheduling; Manpower scheduling; Decomposition; Branch-and-price; Discrete time/resource trade-offs

1 Introduction

Personnel staffing problems define the minimum required number of employees and the associated budget satisfying the staffing requirements to perform certain duties. To determine an accurate staffing plan, a baseline personnel schedule is constructed according to which employees are assigned to duties subject to calendar constraints (Ernst et al., 2004). Dropping the assumption that the personnel demand is known a priori and integrating the demand planning in staff scheduling problems creates benefits in both ways (Maenhout and Vanhoucke, 2017). On the one hand, the flexibility resulting from planning the demand can increase the resource utilisation and/or decrease the required workforce size. On the other hand, the flexibility of incorporating the time-related resource scheduling constraints can lead to a more accurate demand planning.

In this research, we study a project staffing problem that minimises the required personnel budget to carry out the activities to realise a project. Hence, the personnel resource demand stems from a project scheduling problem, which is integrated in the personnel staffing problem to calculate the budget accurately. The personnel staffing problem embodies a days-off manpower scheduling problem. This integration results in an increased flexibility when scheduling the activities as the daily staffing requirements resulting from the project schedule can be adapted to decrease the workforce size. Moreover, additional flexibility is incorporated in two ways. In the activity scheduling, discrete time/resource trade-offs are incorporated by including multiple execution modes, which are characterised by a trade-off between the duration and the resource demand of an activity, i.e. the duration can be decreased when assigning a larger number of employees. In addition, temporary workers can be hired on a day-by-day basis, which increases the flexibility and may further decrease the resource budget.

To tackle this problem, we propose a decomposed branch-and-price procedure that decomposes the strategic budgeting problem in multiple tactical subproblems with a fixed workforce size. For each subproblem, a branch-and-price is invoked to determine the optimal project and personnel staffing plan. The problem is formulated based on the Dantzig-Wolfe decomposition using personnel column variables, which represent feasible individual lines-of-work for employees, and workload column variables, which denote the daily staffing requirements derived from feasible project schedules. Hence, the feasibility towards the time-related scheduling constraints and project scheduling constraints are considered implicitly in the model. For each node, a column generation procedure is applied that iterates between the restricted master problem and two pricing problems, i.e.

the personnel pricing problem and the workload pricing problem, to identify the most promising column variables. The fixing of the workforce size in the subproblem definition allows the development of resource capacity cuts, added to the workload pricing problem, which limit the number of eligible project schedules and accelerate the procedure. These cuts avoid the pitfall of disregarding personnel information while generating project schedules and lead to schedules of higher quality. A three-step branching rule is proposed to branch on fractional workload columns. Extensive computational experiments are conducted to compose a well-performing algorithm. In these experiments, the focus lies on the validation of the added value of the search space decomposition, the resource capacity cuts and the branching rules. Furthermore, the benefits of the proposed problem formulation and the search and bounding strategy are shown. Last, benchmarking is performed with dedicated heuristic and exact procedures proposed in the literature solving an integrated project and personnel staffing problem.

The remainder of this paper is structured as follows. Section 2 discusses the related staffing and project scheduling problems studied in the literature. Section 3 elaborates on the problem definition and provides a mathematical problem formulation. The decomposed branch-and-price procedure is detailed in Section 4. Section 5 discusses the computational experiments to validate the different components in the proposed procedure. Conclusions and directions for future research are presented in Section 6.

2 Literature review

The literature concerning staff scheduling problems with a known and fixed demand is discussed in Section 2.1. The related project scheduling problems are discussed in Section 2.2. Section 2.3 reviews the literature that incorporates demand planning in staff scheduling problems.

2.1 Staff scheduling

Staff scheduling problems can be classified into shift scheduling, days-off scheduling and tour scheduling (Baker, 1976; Ernst et al., 2004). Shift scheduling arises when the staffing requirements vary between different demand periods of a particular day and determines the set of shifts to be worked together with the assignment of the number of employees to each shift, in order to meet demand. Days-off scheduling problems encompass the

assignment of days on and days off to employees spanning the rostering horizon taking different time-related constraints into account. Tour scheduling is the combination of both previous scheduling problems. Each of these problems is thoroughly investigated and an overview of the personnel scheduling literature is given in Alfares (2004), Ernst et al. (2004) and Van den Bergh et al. (2013). Alfares (2004) focuses on the different solution procedures for the tour scheduling problem. Ernst et al. (2004) describe the different variants related to the personnel rostering problem, whereafter the application areas are discussed and the solution methods are categorized. Van den Bergh et al. (2013) classify the personnel scheduling literature based upon the problem characteristics related to the personnel resources, the different types of decisions, the constraints and the different objectives (financial objectives, job satisfaction, etc.). Different solution methods are developed for solving days-off scheduling problems. Some authors focus on cyclic personnel scheduling problems with specific time-related constraints and determine the workforce size in an analytical manner (Alfares, 1998; Baker and Magazine, 1977; Burns and Carter, 1985; Burns and Koop, 1987). Other authors propose mathematical formulations for acyclic personnel scheduling and use mixed integer programming as a solution procedure (Beaumont, 1997; Billionnet, 1999; Caprara et al., 2003). Different formulations are proposed in the literature using decision variables assigning the employees either to individual days on and days off or to individual lines-of-work, which embody a sequence of days on and days off over the problem horizon (Bechtold et al., 1991; Eitzen et al., 2004). The latter decision variables are widely accepted in the personnel scheduling literature and the associated formulations are known to tighten the linear programming relaxation (Barnhart et al., 1998). However, when the number of individual lines-of-work becomes too large, these techniques become inefficient and, as a result, different authors have proposed a branch-and-price methodology (e.g. Brunner et al. (2013); Dohn and Mason (2013); Maenhout and Vanhoucke (2016)).

2.2 Project scheduling

Scheduling project activities under resource constraints is known as the resource constrained project scheduling problem (RCPSPP). The RCPSPP has different variants and extensions, which are summarised in Hartmann and Kolisch (2000) and Brucker et al. (1999). Hartmann and Kolisch (2000) discuss the different objectives in project scheduling and indicate that makespan minimisation is the most common objective. Brucker et al. (1999) review the solution procedures for different project planning problems with

resource constraints.

Weglarz et al. (2011) provide an extensive review of project scheduling problems with multiple alternative execution modes. The discrete time/resource trade-off problem (DTRTP) considers only a single renewable resource, for which personnel resources are exemplary, and stipulates additional flexibility in the planning of the activities via the definition of multiple alternative execution modes for each activity (Ranjbar et al., 2009). Both exact (Demeulemeester et al., 2000) and heuristic (Ranjbar and Kianfar, 2007; Ranjbar et al., 2009) solution procedures are developed for the DTRTP. When the resource is non-renewable, the problem is known as the discrete time/cost trade-off problem (DTCTP) and the activity durations can be decreased at a crashing cost (De et al., 1995). The multi-mode resource-constrained project scheduling problem (MRCPSP) incorporates multiple renewable and non-renewable resources and stipulates time/resource, time/cost and resource/resource trade-offs between the execution modes of an activity (Weglarz et al., 2011). The objective of these problems minimise typically the makespan. There are, however, different other project scheduling problems that better represent the capacity planning of personnel. The Resource Availability Cost Problem (RACP) minimises the resource availability levels, constant over the planning horizon, for all resources (Möhring, 1984). The Time-Constrained Project Scheduling Problem (TCPSP) differs from the RACP since a fixed resource availability is defined and only a decision is made on hiring extra capacity for a certain resource type, which is budgeted as overtime and temporary resources, in order to complete the project within a specific deadline (Verbeeck et al., 2017).

2.3 The Integration of demand planning in staff scheduling problems

In recent literature, the integration of demand planning in staff scheduling problems has been studied in different application areas and different methodologies have been proposed to solve these problems efficiently. Beliën and Demeulemeester (2008) integrate the surgery and nurse scheduling problem. In their research, surgery scheduling represents the demand planning problem and nurse scheduling represents the staff scheduling problem. The objective is to minimise the number of nurses and the problem is solved by a branch-and-price algorithm. Each master surgery schedule is represented by a column variable, which denotes the daily required number of nurses resulting from the master surgery schedule. Beliën et al. (2013) investigate the integration of scheduling aircraft maintenance and the related staffing problem. To avoid non-linearity in the proposed

mathematical model, the problem is decomposed into different subproblems with fixed team sizes. These subproblems are solved to optimality in an enumerative framework via mixed integer linear programming. Mercier et al. (2005) present an integrated aircraft routing and crew scheduling problem to minimise the routing costs and propose a solution procedure based on Benders' decomposition to handle the linking constraints between both problems. Mesquita and Paias (2008) discuss different set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem, for which the objective is to minimise the total trip and duty costs. The integration of crew scheduling and vehicle routing has been studied in different application domains such as home care (e.g. Rasmussen et al. (2012)), technician routing (e.g. Zamorano and Stolletz (2017)) and public transport (e.g. Mesquita et al. (2015)). Artigues et al. (2009) integrate employee timetabling with a job-shop scheduling problem considering the objective of makespan minimisation and propose different constraint programming formulations and linear programming relaxations. The proposed integrated problem formulations focus on scheduling the workforce rather than determining the optimal workforce size.

Several authors incorporate the scheduling of personnel resources in project planning. Alfares and Bailey (1997) consider a cyclic days-off scheduling problem for which the objective is a trade-off between personnel costs and project duration. They apply a heuristic dynamic programming procedure as a solution procedure. Alfares et al. (1999) extends the problem formulation to a heterogeneous workforce and the scheduling of multi-skill employees. Maenhout and Vanhoucke (2016) introduce additional personnel scheduling flexibility by including overtime, temporary personnel and acyclic days-off scheduling and minimise the personnel budget and the activity start time costs. This integrated project staffing problem is solved with a dedicated branch-and-price procedure. Their problem formulation is based on the Dantzig-Wolfe decomposition defining a master problem that considers explicitly the activity start time variables to schedule the project activities and personnel pattern variables, which assign the regular workers to individual days-off schedules. The feasibility of these days-off schedules is defined according to a personnel scheduling subproblem formulation. Since not all personnel pattern variables are considered in the restricted master problem, the personnel scheduling subproblem gives rise to a personnel pricing problem to identify the most promising individual personnel days-off schedules. They present different branching rules and upper bounds to find the optimal solution. Their algorithm is compared with a heuristic approach sequentially solving the project scheduling problem and the personnel staffing problem and other exact methodologies thriving on different MIP formulations. Using

the exact algorithm of Maenhout and Vanhoucke (2016), the same authors explored for the same problem the impact of the flexibility to schedule personnel resources on the staffing budget (Maenhout and Vanhoucke, 2017). They vary the time-related constraints imposed on the days-off schedules of the regular workers and consider different scenarios where only certain resource types (regular, temporary, overtime) are included. In addition, they discuss the trade-off between the different objective function components. Van Den Eeckhout et al. (2019) study the integrated project scheduling and personnel staffing problem with discrete time/resource trade-offs. In order to lower the staffing budget, project scheduling flexibility is considered on the activity level by introducing multiple alternative execution modes for the activities. The problem formulation is similar to the formulation presented in Maenhout and Vanhoucke (2016), explicitly modelling the original project scheduling decision variables, i.e. activity start time variables and activity mode selection variables, in the master problem. The objective is to minimise the staffing budget for regular and temporary workers. A matheuristic algorithm thriving on an iterated local search (ILS) was developed for which the local search is based on different activity- and personnel-based decomposition strategies solving different smaller subproblems in an iterative manner.

3 Problem definition and formulation

3.1 Problem definition

The problem under study comprises a strategic personnel staffing problem that determines the minimum required personnel budget to carry out a single project. In order to improve the staffing cost, we integrate the project scheduling problem in the staffing problem, which provides flexibility related to the timing of the activities and the associated workload. In this way, the project scheduling problem and the personnel staffing problem are optimised simultaneously. The project scheduling problem determines the required number of personnel resources per time unit to carry out the project activities. These activities are scheduled by assigning the activity start times and selecting the execution modes out of several alternative operating modes, each with their own duration and resource requirement. Based on the demand for staff generated by the project scheduling problem, the personnel staffing problem gives insight in the required personnel size and mix of regular and temporary resources. Since calendar or time-related constraints are imposed on the schedules of individual workers, the staffing budget is com-

posed more accurately by solving a days-off manpower scheduling problem to construct a baseline personnel roster taking these personnel scheduling constraints into account. The project deadline defines the planning horizon of the personnel baseline roster. The characteristics of both scheduling problems are discussed in detail below.

The project and activity characteristics

A project plan is constructed based upon an activity-on-the-node network $G = (N, A)$, where activities are represented by a set of nodes N and the finish-start precedence relations with time lag 0 between these activities are included in set A . This implies that an activity can only start when all its predecessors are finished. The set N contains $n+2$ activities, numbered from 0 to $n+1$ with activity 0 and activity $n+1$ representing the dummy start and dummy end activity, which have zero duration and zero resource usage. Each activity i can be executed in one of multiple operation modes ($m \in M_i$). Each mode m of an activity i is characterised by a duration d_{im} and a resource requirement r_{im} . All activities need to be scheduled without pre-emption. A project deadline δ_{n+1} is imposed on the completion of the dummy end activity and defines the planning period T . The scheduling of the activities is further guided by the resource availability stemming from the personnel roster.

The personnel staffing characteristics

The personnel budget is defined by the required number of regular and temporary personnel resources. In order to determine this budget, a baseline personnel roster is constructed able to cover the workload stemming from the project schedule on each day of the planning horizon. The *regular workers* are assigned to individual personnel schedules. These schedules embody a sequence of days on and days off. Time-related constraints impose restrictions on the individual personnel schedules as follows. The planning horizon $T = \{1, 2, \dots, |T|\}$ (index t) is divided into different unit time periods $l \in L$ with length T_l . Counter constraint are defined that restrict the minimum (maximum) number of days on $w^{\min}$ ($w^{\max}$) in each such unit time period. The sequence constraints impose a minimum $n^{\min}$ ($f^{\min}$) and maximum $n^{\max}$ ($f^{\max}$) consecutive number of days on (days off). These constraints restrict the availability of regular workers, which are hired for the entire planning period at a daily cost of c^{RG} . *Temporary workers* are scheduled on a day-by-day basis. The daily cost to hire one temporary worker a single day equals c^{TP} . For clarity, regular workers are hereafter referred to as employees. The term workforce refers the total of employees and temporary workers, unless otherwise specified. All personnel members are identical and possess all skills to carry out the activities.

3.2 Problem formulation

Van Den Eeckhout et al. (2019) have proposed a formulation for the integrated project and personnel staffing problem under study. This formulation is based on the Dantzig-Wolfe decomposition as the personnel scheduling constraints are implicitly considered in the definition of the personnel decision variables, which assign regular workers to individual lines-of-work or personnel patterns. This type of decomposition has been widely applied in the personnel scheduling literature (e.g. Bechtold et al. (1991); Maenhout and Vanhoucke (2016)) and leads to a master problem formulation that considers a large set of personnel pattern variables in order to construct a baseline personnel roster. The project scheduling decisions are denoted using start time and mode assignment variables for each activity. In the following, we explain the decision variables of the model and provide a mathematical formulation of the problem.

Project scheduling variables

The activity mode assignments are modelled by the variables y_{im} , which are equal to 1 if mode m is selected for activity i and 0 otherwise. The activity start times are determined by the variables v_{it} , which take the value 1 if activity i starts at time t and 0 otherwise. In addition, the variables z_{imt} are introduced to avoid the non-linearity when calculating the daily resource demand (i.e. $\sum_{i \in N} \sum_{m \in M_i} \sum_{t^*=t-d_{im}+1}^t r_{im} y_{im} v_{it^*}$) in the staffing requirement constraints. These variables equal 1 if activity i is scheduled in mode m and starts at time t and 0 otherwise. Note that the entire mathematical formulation can be presented with only z_{imt} variables included. However, previous research of Van Den Eeckhout et al. (2019) learned that a stronger LP relaxation can be obtained using y_{im} , v_{it} and z_{imt} variables together.

Personnel staffing variables

The integer number of temporary personnel workers hired for time $t \in T$ is indicated by the decision variable o_t . The regular workers are assigned to a personnel pattern $j \in J$ and the variables x_j indicate the integer number of regular workers assigned to pattern j . The cost of personnel pattern c_j is determined by the planning period T and the daily personnel cost for regular workers c^{RG} , i.e. $c_j = |T| \times c^{RG}$. A personnel pattern j is a duty roster, formed by the parameters a_{jt} , and is comprised of a sequence of days on ($a_{jt} = 1$) and days off ($a_{jt} = 0$). The feasibility of a personnel pattern j is controlled by the following time-related constraints:

$$\sum_{t \in T_l} a_{jt} \geq w^{min} \quad \forall l \in L \quad (1)$$

$$\sum_{t \in T_l} a_{jt} \leq w^{max} \quad \forall l \in L \quad (2)$$

$$\sum_{t^*=t}^{t+n^{min}-1} a_{jt^*} - n^{min} a_{jt} (1 - a_{j,t-1}) \geq 0 \quad \forall t \in \{2, \dots, |T| - (n^{min} - 1)\} \quad (3)$$

$$\sum_{t^*=t}^{t+n^{max}} a_{jt^*} \leq n^{max} \quad \forall t \in \{1, \dots, |T| - n^{max}\} \quad (4)$$

$$\sum_{t^*=t}^{t+f^{min}-1} (1 - a_{jt^*}) - f^{min} a_{j,t-1} (1 - a_{jt}) \geq 0 \quad \forall t \in \{2, \dots, |T| - (f^{min} - 1)\} \quad (5)$$

$$\sum_{t^*=t}^{t+f^{max}} (1 - a_{jt^*}) \leq f^{max} \quad \forall t \in \{1, \dots, |T| - f^{max}\} \quad (6)$$

$$a_{jt} \text{ binary} \quad \forall j \in J, \forall t \in T \quad (7)$$

Constraints (1) and (2) stipulate the minimum and maximum number of working days in a unit time period l . The minimum and maximum number of consecutive days on are restricted by constraints (3) and (4). Constraints (5) and (6) limit the minimum and maximum number of consecutive days off. Constraints (7) impose the binary restrictions on a_{jt} . The set of personnel schedules J consists of all feasible schedules, for which these constraints are satisfied.

Mathematical problem formulation

$$\text{Minimise} \quad Z = \sum_{j \in J} c_j x_j + \sum_{t \in T} c^{TP} o_t \quad (8)$$

$$\text{subject to} \quad \sum_{j \in J} a_{jt} x_j + o_t - \sum_{i \in N} \sum_{m \in M_i} \sum_{t^* = \max(1, t - d_{im} + 1)}^t r_{im} z_{imt^*} \geq 0 \quad \forall t \in T \quad (9)$$

$$\sum_{t \in T} v_{it} = 1 \quad \forall i \in N \quad (10)$$

$$\sum_{m \in M_i} y_{im} = 1 \quad \forall i \in N \quad (11)$$

$$\sum_{t \in T} t v_{jt} - \sum_{t \in T} t v_{it} - \sum_{m \in M_i} d_{im} y_{im} \geq 0 \quad \forall (i, j) \in A \quad (12)$$

$$\sum_{t \in T} t v_{n+1, t} \leq \delta_{n+1} \quad (13)$$

$$z_{imt} \leq y_{im} \quad \forall i \in N, \forall m \in M, \forall t \in T \quad (14)$$

$$z_{imt} \leq v_{it} \quad \forall i \in N, \forall m \in M, \forall t \in T \quad (15)$$

$$z_{imt} \geq v_{it} + y_{im} - 1 \quad \forall i \in N, \forall m \in M, \forall t \in T \quad (16)$$

$$\sum_{m \in M_i} \sum_{t \in T} z_{imt} = 1 \quad \forall i \in N \quad (17)$$

$$x_j \geq 0 \text{ and integer} \quad \forall j \in J$$

$$o_t \geq 0 \text{ and integer} \quad \forall t \in T$$

$$y_{im} \text{ binary} \quad \forall i \in N, \forall m \in M$$

$$v_{it} \text{ binary} \quad \forall i \in N, \forall t \in T$$

$$z_{imt} \text{ binary} \quad \forall i \in N, \forall m \in M, \forall t \in T \quad (18)$$

The objective function (8) minimises the total resource cost. The personnel and project scheduling problem are linked by constraints (9), i.e. the staffing requirements stipulating that each time period a minimum number of regular and temporary workers should be present to carry out the workload stemming from the scheduled activities. Exactly one mode and one start time should be selected for each activity, which is imposed by constraints (10) and (11), respectively. The precedence relations between the activities are stated in constraints (12). Constraint (13) imposes a deadline on the end dummy activity $n + 1$. The constraints (14)-(16) represent the relationship between the y_{im} , v_{it} and z_{imt} variables to linearise the model. Constraints (17) enforce the assignment of exactly one mode and one start time to each activity via the z_{imt} variables. The domain

constraints on the different variables are imposed by constraints (18).

4 Decomposed branch-and-price algorithm

The project staffing problem under study is solved by a decomposed branch-and-price procedure. The different steps of the procedure are displayed in Algorithm 1. The procedure thrives upon the linear programming (LP) relaxation of the formulation of the strategic problem to define smaller problems with a known staffing composition and branching rules to drive a fractional solution to integrality. In order to solve the problem efficiently, we decompose the mathematical problem formulation (8)-(18) using Dantzig-Wolfe decomposition to tighten the linear programming relaxation. The decomposition is related to the project scheduling decisions, i.e. the activity start time and mode assignment decisions, which are in the new formulation modelled by workload pattern variables representing the workload derived from particular project schedules. In this way, the project scheduling variables and the feasibility of the associated project schedules will be considered implicitly in the corresponding master problem. The resulting master problem formulation is characterised by a large number of personnel and workload pattern variables or column variables. The standard solution procedure for obtaining the optimal LP solution for problems with a huge number of column variables is column generation (Barnhart et al., 1998). Using column generation, only a subset of all workload and personnel pattern variables is considered in the so-called restricted master problem, for which the LP solution is obtained. In order to check for the optimality of this solution, a personnel pricing problem and a workload pricing problem are solved to find feasible columns that can enter the basis and reduce the objective function value. The personnel pricing problem embodies an acyclic days-off planning problem to find a feasible personnel line-of-work with the minimum reduced cost. The workload pricing problem is a project scheduling problem with irregular start time costs to construct the project schedule with the lowest reduced cost. Each time a column is found with a negative reduced cost, the column is added to the restricted master problem and the restricted master problem is solved again. When no such column exists, the optimal LP solution has been found, which embodies a lower bound for the optimal integer solution. If this LP optimal solution is integer, the procedure is stopped and the optimal solution is found. Section 4.1 describes the column generation in more detail and provides the mathematical formulation of the (restricted) master problem and the two pricing problems.

When the LP optimal solution is fractional, additional steps are required to drive this fractional solution to integrality. First an initial upper bound solution is constructed using a restricted branch-and-bound procedure, which considers only those workload and personnel pattern variables generated via the column generation procedure in the root node (Section 4.2). Based on this initial upper bound and the optimal LP solution, the search space of the strategic problem under study is decomposed into multiple tactical subproblems by first fixing the number of regular workers and subsequently defining a range of the allowable number of temporary workers, represented by so-called employee and workforce nodes (Section 4.3). For each tactical subproblem, the staffing composition is postulated beforehand and a baseline personnel roster that covers the workload stemming from a project schedule, is pursued. The optimal LP solution and an upper bound are calculated taking the constraints related to the workforce size into account. The different tactical subproblems are visited in the order defined by the search strategy (Section 4.4). For each workforce composition, the optimal integer solution is enumerated via branch-and-price, invoking a column generation step at each node of the branching tree. In order to accelerate the search, resource capacity cuts are added to the workload pricing problem to incorporate personnel staffing information and to obtain higher-quality project schedules and associated workload patterns (Section 4.5). In the branch-and-price framework, branching rules are applied on the personnel and workload pattern variables to reach integer solutions (Section 4.6). In this way, integer solutions are obtained and the incumbent solution can be updated. The procedure terminates when the best known lower bound solution equals the best known upper bound solution, or when the prescribed time limit is reached.

Algorithm 1 Decomposed branch-and-price procedure

```

1: Calculate LP solution based on column generation (Section 4.1)
2: Calculate initial upper bound based on restricted branch-and-bound (Section 4.2)

   Decompose the search space of the strategic problem into tactical subproblems (Section 4.3)
3: Set number of regular workers to zero
4: do
5:   Add constraint to fix the number of regular workers
6:   Solve column generation
7:   if LP solution value < Initial upper bound then
8:     Create new employee node
9:     Calculate minimum and maximum number of temporary workers for employee node
10:    Create workforce nodes: Impose restrictions on the minimum/maximum number of temporary workers
11:    for each workforce node do
12:      Calculate lower and upper bound of workforce node
13:    Increase the number of regular workers by one
14: while Cost regular workers < Initial upper bound

   Determine optimal solution for each workforce node using Branch-and-Price
15: Order workforce nodes according to Workforce node search strategy (Section 4.4)
16: for each workforce node do
17:   Update lower bound
18:   if Lower bound = Upper bound then
19:     Stop algorithm
20:   Formulate resource capacity cuts related to the workforce node in the workload pricing problem (Section
4.5)
21:   do
22:     Column generation
23:     Restricted branch-and-bound (update upper bound)
24:     if Lower bound = Upper bound then
25:       Stop algorithm
26:     if LP solution is fractional then
27:       if Workload pattern variables are fractional then
28:         Branching on workload pattern variables (Section 4.6)
29:       else Personnel pattern variables are fractional
30:         Branching on personnel pattern variables (Section 4.6)
31:       Increase depth of branching tree
32:     else
33:       if LP solution value < upper bound then
34:         Update upper bound
35:       if Lower bound = Upper bound then
36:         Stop algorithm
37:       Backtracking
38:     if Time limit is reached then
39:       Stop algorithm
40:   while Branching tree is not fully explored

```

4.1 Column generation

In this section, we provide insight in the column generation procedure to obtain the LP relaxation of the problem under study and formulate the restricted master problem and the two associated pricing problems to find promising columns. A pattern is added to the restricted master problem if the pattern has a negative reduced cost in the personnel (workload) pricing problem. In that case, the restricted master problem is solved again. If there is no column found with a negative reduced cost, the column generation procedure is terminated. The column generation procedure is presented in Algorithm 2. Since the personnel pricing problem is easier to solve, the workload pricing problem is only solved when no more personnel patterns with a negative reduced cost exist, which is in correspondence with Beliën and Demeulemeester (2008).

Algorithm 2 Column generation

```

1: do
2:   do
3:     Solve restricted master problem
4:     Solve personnel pricing problem
5:   while Personnel pattern with negative reduced cost exists
6:   Solve workload pricing problem
7: while Workload pattern with negative reduced cost exists
  
```

The restricted master problem

We apply the Dantzig-Wolfe decomposition to the mathematical problem formulation in Section 3.2 such that the project scheduling constraints and the original project scheduling variables (z_{imt}) are taken implicitly into account via the definition of so-called workload pattern variables. A workload pattern variable represents a project schedule and defines the activity start times and the execution mode assignments. Hence, in the master problem each feasible project schedule is represented by a workload pattern $k \in K$. The variable p_k takes the value 1 if project schedule k is selected and 0 otherwise. The master problem integrates the personnel staffing problem and the project scheduling problem by composing the staffing plan with minimum cost and selecting an adequate project schedule. This approach tightens the LP relaxation and reduces the number of equivalent solutions in the problem under study by breaking symmetries, which are in abundance present in the problem under study. On the one hand, the same project plan can be staffed with different personnel compositions. On the other hand, the same personnel composition is able to fulfil different project plans.

The resulting master problem formulation is characterised by a large number of personnel and workload pattern variables or column variables. In a column generation procedure, the restricted master problem is defined, which considers only a subset of those variables by introducing the sets $\bar{J} \subset J$ and $\bar{K} \subset K$ and is formulated as follows

$$\text{Minimise} \quad Z = \sum_{j \in \bar{J}} c_j x_j + \sum_{t \in T} c^{TP} o_t \quad (19)$$

$$\text{subject to} \quad \sum_{j \in \bar{J}} a_{jt} x_j + o_t - \sum_{k \in \bar{K}} s_{kt} p_k \geq 0 \quad \forall t \in T \quad (20)$$

$$\sum_{k \in \bar{K}} p_k = 1 \quad (21)$$

$$\begin{aligned} x_j &\geq 0 \text{ and integer} & \forall j \in \bar{J} \\ o_t &\geq 0 \text{ and integer} & \forall t \in T \\ p_k &\text{ binary} & \forall k \in \bar{K} \end{aligned} \quad (22)$$

The objective of the restricted master problem (19) minimises the total personnel cost to carry out the project and is identical to the objective of the original problem formulation (cf. equation (8)). Constraints (20) embody the staffing requirements for each time unit. The staffing requirements s_{kt} per time unit t associated with the selected project schedule k should be covered by regular and/or temporary personnel members. The parameter s_{kt} represents the amount of man units required at time t of the set of activities that are in progress according to project schedule k . Note that regular workers can only carry out project activities at time t if they are assigned to a working day ($a_{jt} = 1$). Constraint (21) states that only one project schedule can be selected. The non-negativity and integrality conditions on the project and personnel scheduling decision variables are stated via constraints (22).

The personnel pricing problem

The personnel pricing problem is based on Maenhout and Vanhoucke (2016) and embodies an acyclic days-off scheduling problem to find a feasible line-of-work for an individual personnel member with minimum reduced cost. The duty assignments composing such a line-of-work are determined by the decision variables u_t which take the value 1 if day t is a workday, 0 otherwise. The reduced cost (RC) is based on the cost of a personnel pattern (c_j) and the dual variables π_t stemming from the staffing constraints (20), i.e. $RC = c_j - \sum_{t \in T} \pi_t u_t$. In order to ensure the feasibility of the line-of-work, the time-

related personnel scheduling constraints are included in this personnel pricing problem, leading to the following formulation:

$$\text{Minimise} \quad c_j - \sum_{t \in T} \pi_t u_t \quad (23)$$

$$\text{Subject to} \quad \sum_{t \in T_l} u_t \geq w^{min} \quad \forall l \in L \quad (24)$$

$$\sum_{t \in T_l} u_t \leq w^{max} \quad \forall l \in L \quad (25)$$

$$\sum_{t^*=t}^{t+n^{min}-1} u_{t^*} - n^{min} u_t (1 - u_{t-1}) \geq 0 \quad \forall t \in \{2, \dots, |T| - (n^{min} - 1)\} \quad (26)$$

$$\sum_{t^*=t}^{t+n^{max}} u_{t^*} \leq n^{max} \quad \forall t \in \{1, \dots, |T| - n^{max}\} \quad (27)$$

$$\sum_{t^*=t}^{t+f^{min}-1} (1 - u_{t^*}) - f^{min} u_{t-1} (1 - u_t) \geq 0 \quad \forall t \in \{2, \dots, |T| - (f^{min} - 1)\} \quad (28)$$

$$\sum_{t^*=t}^{t+f^{max}} (1 - u_{t^*}) \leq f^{max} \quad \forall t \in \{1, \dots, |T| - f^{max}\} \quad (29)$$

$$u_t \text{ binary} \quad \forall t \in T \quad (30)$$

The objective function (23) minimises the reduced cost associated with the line-of-work of a personnel schedule. Constraints (24)-(30) are equivalent to constraints (1)-(7).

The workload pricing problem

The workload pricing problem is a project scheduling problem with irregular start time costs that minimises the reduced cost associated with the workload pattern, while satisfying the different project scheduling constraints. The variables q_t are introduced to represent the staffing requirements on day t . The reduced cost of a workload pattern is based on the dual variables π_t stemming from staffing constraints (20) and a dual variable γ stemming from constraint (21), leading to the reduced cost $\sum_{t \in T} \pi_t q_t - \gamma$. The mathematical problem formulation of the workload pricing problem is as follows:

$$\text{Minimise} \quad \sum_{t \in T} \pi_t q_t - \gamma \quad (31)$$

$$\text{Subject to} \quad \sum_{m \in M_i} \sum_{t \in T} z_{imt} = 1 \quad \forall i \in N \quad (32)$$

$$\sum_{m \in M_j} \sum_{t \in T} t z_{jmt} - \sum_{m \in M_i} \sum_{t \in T} (t + d_{im}) z_{imt} \geq 0 \quad \forall (i, j) \in A \quad (33)$$

$$\sum_{m \in M_{n+1}} \sum_{t \in T} t z_{(n+1)mt} \leq \delta_{n+1} \quad (34)$$

$$\sum_{i \in N} \sum_{m \in M_i} \sum_{t^* = \max(1, t - d_{im} + 1)}^t r_{im} z_{imt^*} = q_t \quad \forall t \in T \quad (35)$$

$$\begin{aligned} z_{imt} &\text{ binary} & \forall i \in N, \forall m \in M_i, \forall t \in T \\ q_t &\geq 0 \text{ and integer} & \forall t \in T \end{aligned} \quad (36)$$

The objective function (31) aims to find the project schedule with minimum reduced cost. Constraints (32) assign exactly one mode and one start time to each activity. Constraints (33) represent the direct precedence constraints with time lag 0. Constraint (34) stipulates the deadline δ_{n+1} for the dummy end activity $n+1$. Equations (35) calculate the daily staffing requirements q_t by summing the workload of all active activities at time t . Constraints (36) embody the domain restrictions of the variables.

Implementation note 1

Before solving the initial restricted master problem in the column generation algorithm, at least one workload pattern should be included in set $\bar{K}$. This ensures feasibility and allows the computation of dual prices. To that purpose, the (feasible) project schedule with the smallest workload is selected, which is useful in the presented decomposed branch-and-price framework (cf. Section 4.3). Note that due to the incorporation of temporary workers, a feasible solution exists even when no personnel patterns are included (i.e. $\bar{J} = \emptyset$).

4.2 Initial upper bound

An initial upper UB_{init} is obtained by applying a branch-and-bound procedure based on the restricted master problem formulation (19)-(22), which is referred to as a restricted branch-and-bound procedure (RBB). This branch-and-bound considers only the set of workload patterns $\bar{K}$ and the set of personnel patterns $\bar{J}$ generated via the column generation procedure in the root node. The initial lower bound LB_{init} equals the optimal

LP solution **value** retrieved by the column generation algorithm, rounded up to the next integer.

4.3 Search space decomposition into tactical subproblems

The objective function minimises the resource cost to find the best mix of regular and temporary workers to cover the workload pattern stemming from a project schedule. This observation is key to decompose the strategic project staffing problem. As mentioned before, the LP solution obtained by column generation will most likely contain fractional values for the decision variables. Moreover, the total number of regular workers $W_{1,\cdot}$ ($= \sum_{j \in \bar{J}} x_j$) and/or temporary workers $W_{2,(.,\cdot)}$ ($= \sum_{t \in T} o_t$) may be fractional. Since a feasible solution requires these numbers to be integer, decomposing the search space of the problem into smaller subproblems with integer $W_{1,\cdot}$ and $W_{2,(.,\cdot)}$ provides an increase of the lower bound. Since the staffing composition is fixed in such subproblem, the strategic problem is translated into a tactical subproblem. These tactical subproblems are defined by specifying the number of regular and (a range of) temporary workers and are created in two steps.

In the *first step*, the employee nodes $e \in E$ are created by defining a specific number of employees $W_{1,e}$ for each node. The number of employee nodes is bounded by the initial upper bound UB_{init} , meaning we are only interested in the set of employee nodes $E = \{e \in \mathbb{N} : \lceil LB_{(e,\cdot)} \rceil < UB_{init}\}$. $LB_{(e,\cdot)}$ denotes the optimal LP solution **value** derived via column generation incorporating the corresponding restriction on the number of employees, i.e. $\sum_{j \in \bar{J}} x_j = W_{1,e}$. Note that when $W_{1,e}$ is too low, a high number of temporary workers will be required causing the total cost Z to exceed UB_{init} . When $W_{1,e}$ is too large, the cost of regular workers will exceed UB_{init} .

Furthermore, based on the corresponding LP solution $LB_{(e,\cdot)}$, a minimum and maximum number of temporary workers ($W_{2,(e,\cdot)}^{min}$, resp. $W_{2,(e,\cdot)}^{max}$) can be calculated. A fractional number of temporary workers $W_{2,(e,\cdot)}$ may be found and thus $W_{2,(e,\cdot)}^{min} = \lceil W_{2,(e,\cdot)} \rceil$. $W_{2,(e,\cdot)}^{max}$ equals the maximum number of temporary workers such that UB_{init} is not exceeded and can be calculated based on UB_{init} and the cost of $W_{1,e}$ according to formula $W_{2,(e,\cdot)}^{max} = \lfloor (UB_{init} - W_{1,e} c_j) / c^{TP} \rfloor$.

In the *second step*, the employee nodes are further divided into workforce nodes $(e, f) \in F_e$ by imposing restrictions on the eligible number of temporary workers, i.e. a minimum and maximum number of temporary workers is specified for each workforce node (resp.

$W_{2,(e,f)}^{min}$ and $W_{2,(e,f)}^{max}$). These bounds determine the allowable range of temporary workers and are defined by the parameter S : $W_{2,(e,f)}^{min} = W_{2,(e,\cdot)}^{min} + f \times S$ and $W_{2,(e,f)}^{max} = W_{2,(e,\cdot)}^{min} + (S-1)$. The set F_e is thus defined as follows: $F_e = \{(e, f) : W_{2,(e,\cdot)}^{min} + f \times S \leq W_{2,(e,\cdot)}^{max}, f \in \mathbb{N}\}$ with $|F_e| = \lfloor (W_{2,(e,\cdot)}^{max} - W_{2,(e,\cdot)}^{min})/S \rfloor + 1$. The parameter S impacts the total number of workforce nodes to be investigated. A smaller value for S will lead to tighter bounds and to more tactical subproblems and vice versa. Setting $S = 1$ implies that for each workforce node a scheduling problem with a fixed personnel budget should be solved. Setting $S = M$ (with M a very big number) means the employee nodes are not further decomposed. The set $F = \cup_{e \in E} F_e$ contains all workforce nodes and consequently the total number of workforce nodes $|F| = \sum_{e \in E} |F_e|$.

As a result of this decomposition, constraints (37)-(39) are added to the restricted master problem to define the resource mix of each workforce node. Constraint (37) stipulates the number of regular workers. Constraints (38) and (39) set the minimum and maximum number of temporary workers. These nodes are exhaustive and mutually exclusive, meaning that a certain workforce mix is related to one subproblem only.

$$\sum_{j \in \bar{J}} x_j = W_{1,e} \quad (37)$$

$$\sum_{t \in T} o_t \geq W_{2,(e,f)}^{min} \quad (38)$$

$$\sum_{t \in T} o_t \leq W_{2,(e,f)}^{max} \quad (39)$$

Figure 1 shows an example of the decomposition into employee and workforce nodes for a project with deadline $\delta_{n+1} = 42$. For each node, the corresponding bounds on temporary workers and solution quality are displayed, with $c_j = 2 \times 42 = 84$ and $c^{TP} = 4$. The optimal LP solution value (LB_{init}) and UB_{init} are displayed below the root node of the tree. Six employee nodes are created ($|E| = 6$), together with twenty-one workforce nodes ($S = 5$, $|F| = 21$) from which a subset is displayed. Setting $W_{1,\cdot} = 15$ does not result in an employee node, since the cost of regular workers (i.e. $15 \times 84 = 1260$) exceeds UB_{init} . The same idea holds when $W_{1,\cdot} = 8$, since the high number of temporary workers in the optimal LP solution leads a total cost of 1212 ($= 8 \times 84 + 128 \times 4$), exceeding again UB_{init} .

Figure 1: Example of the search space decomposition into employee and workforce nodes

Lower bound improvement

As a result of the search space decomposition, the initial lower bound LB_{init} can be increased in different ways. This strengthened lower bound is indicated by LB_{deco} . First, decomposing the problem into employee nodes and calculating the different optimal LP solutions may increase the lower bound. Second, for each employee node, the number of temporary workers in the specific LP solution may be rounded up to the next integer. Third, when the workload pattern resulting from the feasible project schedule with the minimum work content cannot be covered by a specific workforce node, that node can be discarded and $LB_{(e,.)}$ increases immediately. As a result, $LB_{(.,.)}$ equals the minimum lower bound of all employee nodes, i.e. $LB_{(.,.)} = \min_{e \in E}(LB_{(e,.)})$. In our example in Figure 1, this means that $LB_{(.,.)}$ is found in the employee node $W_{1,12}$, i.e. $LB_{(.,.)} = LB_{deco} = LB_{(12,.)} = 1068$.

Implementation note 2

Due to the workforce constraints (37)-(39) in the workforce nodes and the workload pattern assignment constraint (21), the restricted master problem can become infeasible. This occurs when, given the set of personnel patterns $\bar{J}$, no staffing plan could be obtained to cover a workload pattern $k \in \bar{K}$. In order to avoid infeasibility, we include an artificial column variable p' in constraint (21), leading to constraint (40). The use of p' is heavily penalised in the objective function¹. When p' is different from 0 in the

¹If $S > 1$, the penalty cost c^{pen} associated with p' should satisfy the following inequality: $c^{pen} > c^{TP}$ since including an extra temporary worker is more favourable than $p' > 0$. When $S = 1$, $c^{pen} = 1$ is

optimal LP solution, the branching node can be pruned.

$$\sum_{k \in \bar{K}} p_k + p' = 1 \quad (40)$$

4.4 Workforce node search strategy

The proposed solution methodology decomposes the problem into different workforce nodes. For each workforce node, a branch-and-price is invoked which focuses on a specific part of the solution space. The order of solving these workforce nodes defines the search throughout the solution space.

The search focuses on increasing the lower bound $LB_{(.,.)}$ and enumerates the workforce nodes in ascending order of their lower bound value $LB_{(e,f)}$. If the selected workforce node is feasible and solved to optimality, the optimal solution of the specific employee node is found since all remaining workforce nodes of the involved employee node have a higher objective function value. The search for the optimal solution of the overall problem continues in the next employee node to be investigated. If no employee node exists with a lower bound $LB_{(e.,.)}$ smaller than the incumbent solution $UB_{(.,.)}$, the overall optimal solution is found. The general idea is thus to increase the lower bound $LB_{(.,.)}$ by gradually increasing the workforce size until a feasible, integer solution is obtained.

In order to improve the incumbent solution and, hence, to decrease the upper bound, a *restricted branch-and-bound* is called after the column generation step in each node of the search process. This column generation step may lead to an infeasible or fractional solution. However, in the latter solution there may be a very good incumbent solution contained in the generated set of personnel patterns $\bar{J}$ and the set of workload patterns $\bar{K}$. In order to try to improve the incumbent, we apply a restricted branch-and-bound (RBB) based on eqs. 19)-(22).

4.5 Resource capacity cuts in the workload pricing problem

The presented cuts are introduced in the workload pricing problem to improve the quality of the generated workload patterns. The different cuts are proposed to take the (maximum) resource capacity of the workforce size postulated by a particular workforce

sufficient since no extra temporary worker can be added.

node into account. In this way, we avoid that the predefined number of regular and temporary workers cannot properly staff the generated workload pattern. These cuts will thus reduce the number of eligible workload patterns and will thus discard a large amount of the infeasible search space. Each proposed type of cut differs in the focus on a predetermined interval in the planning period and/or in the level of detail. More specifically, the cuts specify the minimum number of temporary workers needed to cover the demand by subtracting the resource capacity, provided by the number of regular workers employees $W_{1,e}$ associated with a particular workforce node, from the workload stemming from the scheduling of the activities. Therefore, all proposed cuts include the decision variables o_t in the workload pricing problem, together with constraints (38) and (39). No feasible workload patterns exists if $\sum_{t \in T} o_t$ exceeds $W_{2,(e,f)}^{max}$. The lower bound $W_{2,(e,f)}^{min}$ is incorporated since solutions with a lower number of temporary workers are already visited in earlier workforce nodes. Cuts 1 to 4 compare the workload stemming from the project schedule with the resource capacity of the fixed number of regular workers $W_{1,e}$ in different time intervals. Cut 5 actually schedules the resource capacity over the planning horizon. The different cuts are explained in detail below.

- *Cut 1* takes the entire planning period into account and enforces that the total generated workload of a pattern ($\sum_{t \in T} q_t$) is smaller than or equal to the maximum resource capacity provided by the regular workers (i.e. $|L| \times w^{max} \times W_{1,e}$) and temporary workers ($\sum_{t \in T} o_t$).

$$\sum_{t \in T} o_t \geq \sum_{t \in T} q_t - |L| w^{max} W_{1,e} \quad (\text{Cut 1})$$

- *Cut 2* imposes that the workload of the generated workload pattern in a unit time period T_l ($\sum_{t \in T_l} q_t$) should be smaller or equal to the total maximum resource capacity provided by the regular workers ($w^{max} \times W_{1,e}$) and temporary workers ($\sum_{t \in T_l} o_t$) in that unit time period. This cut is similar to the previous one since only the maximum number of assignments and no other time-related constraints are taken into account. Note that Cut 2 is a stronger cut compared to Cut 1, since idle resources in one unit time period cannot cover the workload in another unit time period.

$$\sum_{t \in T_l} o_t \geq \sum_{t \in T_l} q_t - w^{max} W_{1,e} \quad \forall l \in L \quad (\text{Cut 2})$$

- *Cut 3* considers a single day in the planning horizon and postulates that the workload

of the workload pattern on each single day should be smaller or equal to the total maximum resource capacity of the regular and temporary workers on that day. This cut does not consider any time-related constraints but ensures that an excess daily resource demand cannot be covered by idle resource capacity on another day like in Cut 1 and Cut 2.

$$o_t \geq q_t - W_{1,e} \quad \forall t \in T \quad (\text{Cut 3})$$

- *Cut 4* defines a rolling horizon with length equal to the maximum number of consecutive workdays (n^{max}) and the minimum number of consecutive days off (f^{min}). This cut imposes that the workload in any period with length $(n^{max} + f^{min})$ should be smaller or equal to the maximum resource capacity provided by the regular workers ($n^{max} W_{1,e}$) and temporary workers ($\sum_{t^*=t}^{t+n^{max}+f^{min}-1} o_{t^*}$). In contrast to Cut 1 and Cut 2, Cut 4 considers time-related sequence constraints, i.e. the maximum number of consecutive workdays and the minimum number of consecutive days off.

$$\sum_{t^*=t}^{t+n^{max}+f^{min}-1} o_{t^*} \geq \sum_{t^*=t}^{t+n^{max}+f^{min}-1} q_{t^*} - n^{max} W_{1,e} \quad \forall t \in \{1, \dots, |T| - (n^{max} + f^{min} - 1)\} \quad (\text{Cut 4})$$

Cuts 1 to 4 consider only partially the time-related constraints imposed on an individual personnel schedule, which may lead to an overestimation of the resource capacity. In order to devise a workload pattern that better incorporates the impact of the time-related constraints, the idea of *Cut 5* is to construct an actual baseline personnel roster when solving the workload pricing problem. For reasons of computational efficiency, we decompose the entire planning horizon and create an actual personnel baseline roster for each unit time period l by including the decision variables x'_{jl} , which indicate the number of regular personnel members assigned to personnel pattern $j \in J_l$ in each unit time period l . Hence, Cut 5 considers the multiple unit time periods in the planning horizon in an independent manner, i.e. without any time-related sequence or counter constraints imposed between the assignments of different unit time periods. As a result, the problem can be reduced to the assignment of regular workers to lines-of-work per unit time period. Given the limited length of a unit time period $|T_l|$, all feasible personnel patterns $j \in J_l$ can be enumerated considering all relevant time-related constraints within this time period. Since this set of patterns is reasonably small, they can be

included in the workload pricing problem for finding a workload pattern with a better quality given the provided resource capacity. Constraints (Cut 5a) indicate that the generated daily workload (q_t) for day t should be adequately covered by the scheduled regular workers ($\sum_{j \in J_l} a_{jt} x'_{jl}$) and the number of temporary workers (o_t). Constraints (Cut 5b) ensure that the sum of the x'_{jl} variables is equal to the regular workforce size $W_{(1,e)}$ for each unit time period. Since no time-related constraints are considered between the different unit time periods, the required number of temporary workers is still a lower bound.

$$o_t \geq q_t - \sum_{j \in J_l} a_{jt} x'_{jl} \quad \forall t \in T \quad (\text{Cut 5a})$$

$$\sum_{j \in J_l} x'_{jl} = W_{1,e} \quad \forall l \in L \quad (\text{Cut 5b})$$

Two-phased approach to solve the workload pricing problem

Adding the resource capacity cuts to the workload pricing problem limits the set of eligible patterns and may lead to a smaller number of branching steps on the column variables p_k . However, the downside of these cuts is the increased complexity of the pricing problem. For that purpose, we adopt a two-phase procedure to solve the workload pricing problem and add or remove certain cuts in phase II whenever the time limit for solving the pricing problem in phase I is reached. The implementation related to the type of cuts explored in each phase is determined via computational experiments and is discussed in Section 5.2.2.

4.6 Branching rules

In the optimal LP solution of any node, two types of variables may be fractional and branching is required to find the optimal integer solution. First, branching on the workload pattern variables p_k is performed. Second, when all p_k variables are integer, branching on the personnel scheduling variables x_j is carried out.

When *branching on the workload pattern variables* p_k , a constraint branching scheme is applied based upon the (aggregated) workload (Ryan and Foster, 1981). The workload patterns $k \in \bar{K}$ that do not satisfy all the branching constraints (imposed on the underlying s_{kt} parameters) are removed from the set $\bar{K}$. Branching constraints are added to the workload pricing problem such that only eligible workload patterns that satisfy the

imposed restrictions on the workload, are generated and added to $\bar{K}$. Besides branching on the workload of individual time periods (e.g. Beliën and Demeulemeester (2008)), branching can be conducted on the aggregated workload resulting in a three-step branching rule with the following associated branching constraints, i.e.

- **Step 1** (WL_T): We first verify if all the fractional patterns have the same total workload. If this is not the case, the fractional pattern with the lowest workload k' is selected, resulting in branching constraints (41).

$$\sum_{t \in T} q_t \leq \sum_{t \in T} s_{k't} \quad \text{and} \quad \sum_{t \in T} q_t \geq \sum_{t \in T} s_{k't} + 1 \quad (41)$$

- **Step 2** (WL_l): When all the fractional patterns have the same total workload, the workload in each unit time period ($\sum_{t \in T_{l'}} s_{kt}$) is calculated for each workload pattern with a fractional value. The period l' with the largest workload range, calculated as the difference between the maximum and minimum of these workload totals, is selected. Branching is performed on the workload of the fractional pattern k' with the minimum workload for this unit time period l' , resulting in constraints (42). If the largest workload range is equal to zero for any unit time period, this branching rule is not applied.

$$\sum_{t \in T_{l'}} q_t \leq \sum_{t \in T_{l'}} s_{k't} \quad \text{and} \quad \sum_{t \in T_{l'}} q_t \geq \sum_{t \in T_{l'}} s_{k't} + 1 \quad (42)$$

- **Step 3** (WL_t): When the workload in each unit time period is the same for all fractional patterns, branching on the workload of a particular day is performed via including the constraints (43) and (44). In this branching strategy, the two workload patterns k' and k'' with the largest fractional values, are selected. Branching is performed on the workload of the first period t for which the workload between those two patterns is different.

$$q_t \leq s_{k't} \quad \text{and} \quad q_t \geq s_{k't} + 1 \quad \text{If } s_{k't} < s_{k''t} \quad (43)$$

$$q_t \leq s_{k''t} \quad \text{and} \quad q_t \geq s_{k''t} + 1 \quad \text{If } s_{k't} > s_{k''t} \quad (44)$$

Note that these branching rules are applied in a hierarchical order with a decreasing length of the time period over which the workload is calculated. The presence of the third branching step is required to ensure a complete branching scheme. The first two

branching steps create extra branching levels, but may lead to faster backtracking.

For *branching on the personnel assignments* x_j , we applied the branching rule developed by Maenhout and Vanhoucke (2016), which is based on research of Vanderbeck and Wolsey (1996). This branching rule branches on the columns characterised by a subset of days-on assignments for which a fractional value is identified. Note that the daily number of temporary workers o_t will be integer if all x_j variables are integer.

5 Computational experiments

This section provides computational insights into the proposed algorithm. The test design is discussed in Section 5.1. Section 5.2 investigates the impact of the search space decomposition, the cuts introduced in the workload pricing problem and the branching rules. Section 5.3 indicates the performance on a larger dataset and we show the lower and upper bound improvements found by the proposed methodology for different problem sizes. Section 5.4 benchmarks the presented procedure with different dedicated exact and heuristic solution procedures for the integrated personnel and project scheduling problem. The algorithm was coded in C++ with the use of Gurobi 7.0. All tests were performed on an Intel Core E5-2660v3 processor with 4GB RAM and 2.6Ghz.

5.1 Test design

We selected the instances with 10, 20 and 30 activities from the multi-mode Project Scheduling Problem Library (PSPLIB) (Kolisch and Sprecher, 1997). From these instances, only the *network topology* is preserved, since only one renewable resource is considered representing man units and most of the modes become inefficient. We applied a random *mode generation* to generate three efficient modes for each activity, with a duration and demand uniformly distributed between 1 and 10. Based on the characteristics of the first created mode, generating two other efficient modes may not be possible, leading to an average of 2.91 modes per activity. This number of modes is further influenced by the imposed *deadline*, since a shorter deadline leads to more infeasible modes. We applied the deadline formula of Vanhoucke and Debels (2007) and set the deadline at the middle of the longest path (LP) and the shortest path (SP), i.e. $\delta_{n+1} = SP + 0.5 \times (LP - SP)$.

The parameters to determine the staffing plan are based on the literature (Van den Bergh et al., 2013; Maenhout and Vanhoucke, 2017). The *objective function weights* reflect that hiring temporary personnel members for a single day ($c^{TP} = 4$) is twice as expensive as hiring regular workers ($c^{RG} = 2$). The parameter values of the *time-related constraints* are set at common values encountered in real-life and literature and are parametrised as follows: $w^{min} = 5$, $w^{max} = 5$, $n^{min} = 2$, $n^{max} = 6$, $f^{min} = 1$ and $f^{max} = 2$.

Table 1 provides insights in the characteristics of the test dataset, which consists of three sets with a different number of activities $|N|$. We display the number of instances, the average deadline and the average number of regular and temporary workers of the best known lower bound solutions. To test the impact of the algorithmic parameters (Section 5.2), a subset of 30 instances from $|N| = 20$ is selected for computational reasons. These instances are selected based on systematic sampling to incorporate different network topologies (Vanhoucke et al., 2008). Computational tests on the entire dataset are presented in Section 5.3.

	$ N = 10$	$ N = 20$	$ N = 30$
$\#instances$	536	554	640
δ_{n+1}	25	41	51
W_1	9	11	14
W_2	17	16	12

Table 1: Problem size parameters

5.2 Algorithmic performance

In Section 5.2.1, the impact of the search space decomposition of the overall problem into different employee and workforce nodes is investigated. Section 5.2.2 discusses the impact of including resource capacity cuts in the workload pricing problem. The impact of the branching strategies is analysed in Section 5.2.3. Each analysis is performed in a *ceteris paribus* manner, meaning that the parameters not included in the analysis are set at their best values. The time limit of the procedure is set at 3600 seconds per instance. The time limit for the restricted branch-and-bound is set at 60 seconds. The following symbols are used to describe the computational results averaged over the set of instances:

LB_{fin}	Lower bound retrieved when the stop criterion of the procedure is met
UB_{fin}	Upper bound denoted by the incumbent solution when the stop criterion of the procedure is met
$GAP_{\%}$	Relative optimality gap $(= (UB_{fin} - LB_{fin}) / LB_{fin})$
$Opt_{\%}$	Percentage of instances which are, given the time limit, solved to optimality
CPU	CPU time (in seconds)
$\#nd_{tree}$	Number of nodes in the branching trees
$lvl_{\#}^{wl}$	Number of workload branching levels
$\#pat_{tot}^{wl}$	Number of added workload patterns at the end of the procedure

5.2.1 Decomposition framework

We first analyse the added value of decomposing the search space of the overall problem into smaller subproblems and show the impact of the subproblem definition. To that purpose, we explore the following variants of the algorithm:

- **No DC:** No decomposition is applied, i.e. branching constraints are immediately imposed on the root node without decomposing the problem into employee nodes or workforce nodes.
- **$S = M$:** The root node is further decomposed into employee nodes but not into workforce nodes. Starting from each employee node, branching constraints are imposed.
- **$S \in \{1, 3, 5\}$:** We decompose the root node into employee and workforce nodes. Branching constraints are defined starting from the workforce nodes. We vary the parameter S to analyse the impact of the allowable range of the temporary workers.

Table 2 displays the results and indicates the number of created employee nodes (E) and workforce nodes (F). The number of nodes that are actually visited are indicated between brackets. This table clearly shows the added value of decomposing the problem, since the percentage of optimised instances increases when defining subproblems with a fixed number of regular workers and the allowable range on the number of temporary workers (S) associated with the workforce nodes becomes smaller. When no decomposition is applied, we are not able to find the optimal solution for any instance, leading to an average optimality gap of 19.52%. Creating employee nodes decreases this gap to 9.58%. However, the optimal solution can only be obtained for 3% of the instances. When (more) workforce nodes are created (i.e. S decreases), the percentage of optimised instances increases, reaching 100% when $S = 1$.

	LB_{fin}	UB_{fin}	$Gap\%$	$Opt\%$	CPU	$\#nd_{tree}$	E	F
<i>No DC</i>	863.5	1028.0	19.52	0	3600	12468	- (-)	- (-)
$S = \mathbf{M}$	881.0	959.7	9.58	3	3536	4356	8 (1)	- (-)
$S = \mathbf{5}$	898.1	907.5	1.10	27	2726	778	8 (2)	43 (3)
$S = \mathbf{3}$	898.9	910.1	1.74	47	2245	282	8 (2)	69 (5)
$S = \mathbf{1}$	900.1	900.1	0.00	100	643	45	8 (2)	200 (12)

Table 2: The computational impact of the search space decomposition into subproblems

Tighter resource capacity cuts are defined in the workload pricing problem when the workforce nodes are defined with a smaller allowable range on the number of temporary workers. This leads to a smaller number of nodes in the branching tree ($\#nd_{tree}$), which in turns results in a lower CPU time. On average 8 employee nodes are created and only one or two nodes are visited, indicating the minor importance of UB_{init} when creating these nodes. The total and visited number of workforce nodes increases when S decreases, indicating that infeasibility can be proven faster when the defined range on the number of temporary workers decreases. This explains the significant decrease in the required CPU time in case the workforce nodes are strictly defined ($S = 1$).

5.2.2 Definition of cuts in the workload pricing problem

Table 3 shows different combinations of cuts included in the workload pricing problem. A two-phase solution methodology is proposed to find a suitable workload pattern, where in each phase the workload pricing problem is solved taking a specific set of cuts into account. For the different phases, the numbers between accolades indicate which cuts are included, whereas '-' indicates that the workload pricing problem is not solved via the two-phase method and a workload pattern is obtained via the first phase only. When a two-phased approach is applied and no workload pattern is found in the first phase due to the time limit, less restrictive cuts are included to find a suitable workload pattern in the second phase. The lower index indicates whether we seek for the workload pattern with the lowest reduced cost ('opt') or a feasible workload pattern with a negative reduced cost ('feas'). The upper index shows the particular time limit to solve the corresponding phase. Either the time limit for the first phase is set at 600 seconds based on preliminary experiments or no time limit is imposed, indicated by '-'. For example, setting 4 indicates that the first phase includes cuts 4 and 5 which is solved to optimality with a time limit of 600 seconds, whereas the second phase includes cut 3, which is solved to optimality without a time limit. Further, the table shows different performance measures together

with the time spent for solving the workload pricing problem, which is indicated between brackets in the column 'CPU'.

The first setting (setting 0) in Table 3 displays the results when no cuts are added. Settings 1-3 show different versions that consider only a single phase. Setting 1 (resp. setting 2) shows the impact of solving the workload pricing problem to optimality (resp. feasibility) with (the non-dominated) cuts 4 and 5. Setting 3 allows for the comparison with setting 7, i.e. the best performing two-phase version, as both exploit exactly the same cuts in phase 1. Settings 4-8 display the impact of different settings for the first phase, whereas settings 9-14 show the impact of different settings for the second phase of the workload pricing problem. The impact of dropping cut 5 and 4 are displayed in setting 15 and 16. Note that also other combination of cuts were tested, but only the most interesting ones are shown.

When comparing the different versions, we obtain six insights concerning which cuts should be added to the workload pricing problem. First, and most important, adding the non-dominated cuts to the workload pricing problem (setting 1) increases the percentage of optimal instances from 3% to 87% and decreases the optimality gap from 5.80% to 2.10%. Second, solving the workload pricing problem to feasibility deteriorates the results with an increase of the $Gap\%$ from 2.10% (setting 1) to 4.64% (setting 2). This conclusion holds for all other settings, but for clarity, these results are not displayed. Third, the two-phase solution methodology proved beneficial as can be seen in setting 4-8. Based on these settings, the fourth conclusion is that adding all dominated cuts, comparing settings 4 and 8, leads to solving all instances to optimality while reducing the CPU time (758 to 679 seconds). However, setting 7, which deletes cut 1 and retains the other cuts, further reduces the computational effort (679 to 643 seconds). A similar conclusion can be made when considering only a single phase, i.e. solving $\{2, 3, 4, 5\}_{opt}^-$ (setting 3) leads to a small improvement compared to setting 1 as the $Gap\%$ decreases from 2.10% to 1.69%. Fifth, comparing setting 7 with settings 9-14, we observe that only including cut 3 (setting 7) in the second phase obtains the best results since only this version leads to the optimal solution for all instances. Sixth, the added value of cut 5 (respectively cut 4) is shown in setting 15 (resp. 16), which indicates that removing one of these cuts leads to a higher optimality gap since some instances are not solved to optimality.

	Phase I	Phase II	LB_{fin}	UB_{fin}	$Gap\%$	$Opt\%$	CPU	$lvl_{\#}^{wl}$	$\#pat_{tot}^{wl}$
0.	-	-	884.5	932.3	5.80	3	3487 (2146)	523	7152
1.	$\{4, 5\}_{opt}^{-}$	-	899.4	914.1	2.10	87	960 (952)	5	8
2.	$\{4, 5\}_{feas}^{-}$	-	898.2	933.5	4.64	70	1378 (1373)	5	8
3.	$\{2, 3, 4, 5\}_{opt}^{-}$	-	899.5	910.3	1.69	93	967 (958)	15	18
4.	$\{4, 5\}_{opt}^{600s}$	$\{3\}_{opt}^{-}$	899.9	902.2	0.44	97	758 (749)	5	8
5.	$\{2, 4, 5\}_{opt}^{600s}$	$\{3\}_{opt}^{-}$	899.9	902.2	0.44	97	750 (740)	6	9
6.	$\{3, 4, 5\}_{opt}^{600s}$	$\{3\}_{opt}^{-}$	900.1	900.1	0.00	100	700 (693)	7	10
7.	$\{2, 3, 4, 5\}_{opt}^{600s}$	$\{3\}_{opt}^{-}$	900.1	900.1	0.00	100	643 (634)	14	16
8.	$\{1, 2, 3, 4, 5\}_{opt}^{600s}$	$\{3\}_{opt}^{-}$	900.1	900.1	0.00	100	679 (671)	7	10
9.	$\{2, 3, 4, 5\}_{opt}^{600s}$	$\{2, 3, 4\}_{opt}^{-}$	899.5	912.0	1.89	90	918 (910)	15	17
10.	$\{2, 3, 4, 5\}_{opt}^{600s}$	$\{3, 4\}_{opt}^{-}$	900.1	901.3	0.13	97	737 (728)	15	18
11.	$\{2, 3, 4, 5\}_{opt}^{600s}$	$\{2, 4\}_{opt}^{-}$	899.9	905.9	1.26	93	811 (802)	15	17
12.	$\{2, 3, 4, 5\}_{opt}^{600s}$	$\{2, 3\}_{opt}^{-}$	899.7	910.3	1.67	93	858 (850)	15	17
13.	$\{2, 3, 4, 5\}_{opt}^{600s}$	$\{2\}_{opt}^{-}$	899.9	905.8	1.11	90	827 (818)	14	17
14.	$\{2, 3, 4, 5\}_{opt}^{600s}$	$\{1\}_{opt}^{-}$	899.9	904.9	0.96	97	753 (744)	14	17
15.	$\{2, 3, 4\}_{opt}^{600s}$	$\{3\}_{opt}^{-}$	899.5	908.5	1.04	83	976 (953)	41	48
16.	$\{2, 3, 5\}_{opt}^{600s}$	$\{3\}_{opt}^{-}$	899.3	900.9	0.23	90	839 (806)	77	155

Table 3: The computational impact of including cuts in the workload pricing problem

In review of the computational performance, the results in Table 3 lead to three findings when adding cuts to the workload pricing problem. For that purpose, we compare setting 0, i.e. without cuts, and setting 7, which leads to the best performance. First, solving the workload pricing problem takes almost all of the computation time when cuts are added (62% versus 99% of the time). Second, adding cuts leads to a smaller branching tree since there is a steep decrease in the number of workload branching levels $lvl_{\#}^{wl}$ from 523 to 14. Third, adding cuts leads to less eligible patterns as the total number of added workload patterns ($\#pat_{tot}^{wl}$) drops substantially from 7152 to 16. This clearly shows the need of adding personnel information in the workload pricing problem to construct a small set of highly relevant workload patterns to obtain the optimal solution.

5.2.3 Branching rules

We proposed three different branching strategies, i.e. branching on the workload of individual days (WL_t), branching on the total workload during a unit time period (WL_l) or branching on the total workload over the planning horizon (WL_T), and combined them in a three-step branching rule (cf. Section 4.6). The added value of the different steps is shown in Table 4 by analysing the following branching rules:

- $WL_T \rightarrow WL_l \rightarrow WL_t$: The three-step branching rule as presented in Section 4.6.
- $WL_l \rightarrow WL_t$: This strategy incorporates only branching on the workload during a unit time period and individual days.
- $WL_T \rightarrow WL_t$: This strategy incorporates branching on the workload during the entire planning period and individual days.
- WL_t : Single-step branching scheme that branches only on individual days.

All displayed branching rules optimise all the instances and therefore we analyse only the CPU time and the size of the branching tree in Table 4. The average number of branching levels based on each rule is displayed by respectively $lvl_{\#}^T$, $lvl_{\#}^l$ and $lvl_{\#}^t$. The total number of workload branching levels is shown by $lvl_{\#}^{wl}$.

	CPU	$lvl_{\#}^{wl}$	$lvl_{\#}^T$	$lvl_{\#}^l$	$lvl_{\#}^t$
$WL_T \rightarrow WL_l \rightarrow WL_t$	652	13.4	0.2	2.0	11.3
$WL_l \rightarrow WL_t$	706	13.6	0.0	2.2	11.3
$WL_T \rightarrow WL_t$	643	14.0	0.3	0.0	13.8
WL_t	670	14.1	0.0	0.0	14.1

Table 4: Computational performance of different branching rules

When comparing the results in Table 4, the differences between the branching rules are minor. Comparing the rules $WL_T \rightarrow WL_t$ and WL_t , we observe that adding the strategy WL_T decreases the CPU time from 670 to 643 seconds despite the fact that the size of the branching tree remains practically the same. On average 0.3 branching levels based on the strategy WL_T are created, which may cause the lower CPU time. When the strategy WL_l is also included, leading to the three-step branching rule $WL_T \rightarrow WL_l \rightarrow WL_t$, the CPU time increases from 643 to 652 seconds although the size of the branching tree decreases from 14.0 to 13.4 workload levels. This means that branching on the workload of a unit time period ($lvl_{\#}^l = 2.0$) is computationally more difficult, which is also revealed by the performance of the branching rule $WL_l \rightarrow WL_t$.

5.3 Performance on the entire dataset

The purpose of this section is twofold. First, we show the performance of the presented procedure on the entire dataset for different problem sizes. Second, we analyse the lower and upper bound improvements and demonstrate the suitability of the workforce node search strategy (cf. Section 4.4), which visits the subproblems in ascending order of the provided lower bounds calculated in the workforce nodes.

Table 5 presents the bounds retrieved in different stages of the procedure. The table displays different lower bounds, i.e. the optimal LP solution value in the root node (LB_{init}), the strengthened lower bound after the search space decomposition of the strategic problem into multiple tactical subproblems (LB_{deco}) and the final lower bound retrieved when the stop criterion of the procedure is met (LB_{fin}). In addition, we display the upper bound **obtained** by exploiting a branch-and-bound search using the patterns generated in the root node (UB_{init}) and the final upper bound indicated by the incumbent solution when the stop criterion of the procedure is met (UB_{fin}). Each time, the improvement with respect to the previous lower or upper bound is calculated. The results are averaged over all instances of the entire dataset given a particular number of activities.

	LB_{init}	LB_{deco}	LB_{fin}	UB_{init}	UB_{fin}	$GAP\%$	$Opt\%$	CPU
$ N = 10$	475.7	481.9 +1.4 %	507.3 +5.8 %	573.7	507.3 -11.5 %	0.0	100	20
$ N = 20$	944.7	954.9 +1.1 %	979.9 +2.8 %	1141.9	987.5 -13.4 %	0.9	91	758
$ N = 30$	1429.9	1443.8 +1.0 %	1458.8 +1.1 %	1736.6	1487.1 -14.1 %	2.1	76	1730

Table 5: Computational results on the entire dataset

Table 5 reveals that all the instances with 10 activities are solved to optimality, with an average computation time of 20 seconds. 91% of all the instances with $|N| = 20$ are solved to optimality with an average deviation ($Gap\%$) of 0.9% as result. Although the number of activities doubles from 10 to 20, the CPU time increases by a factor 38, i.e. 758 seconds on average. For the instances with 30 activities, the optimal solution is found for 76% of the instances, leading to an average deviation ($Gap\%$) of 2.1%. The CPU time increases to 1730 seconds.

When analysing the lower and upper bound improvements for $|N| = 10, 20$ and 30, we

notice that the best found solution values lie closer to LB_{deco} than to UB_{init} , justifying the proposed search strategy. The search space decomposition of the overall problem into workforce nodes already leads to an increase in the lower bound values of respectively 1.4%, 1.1% and 1.0%. As a result of the conducted brach-and-price search at each workforce node, the lower bound further increases respectively with 5.8%, 2.8% and 1.1%. In contrast, the upper bound solution decreases respectively with 11.5%, 13.4% and 14.1%. Furthermore, we observe that the relative improvements of the different lower bounds decrease when the number of activities $|N|$ increases, whereas the relative improvements of the upper bound increase. This is, however, a consequence of the higher staffing budgets, since the improvements in the lower and upper bound increase in absolute terms when the number of activities $|N|$ increases.

5.4 Benchmarking

In this section, we benchmark the proposed methodology with different other exact and heuristic methodologies that are designed in the literature to solve the integrated personnel and project scheduling problem. First, we compare the quality of the linear programming relaxation and the resulting lower bound based on the mathematical formulation proposed in Section 4.1 and the formulation of Van Den Eeckhout et al. (2019). Second, we benchmark the incumbent solutions obtained using the proposed methodology with an exact branch-and-price methodology thriving on the problem formulation of Section 3.2, the iterated local search of Van Den Eeckhout et al. (2019) and a heuristic restricted branch-and-bound.

5.4.1 The quality of the linear programming relaxation

In order to evaluate the quality of the lower bound resulting from the linear programming relaxation, we compare the performance of the following lower bounds:

- LP_{expl} : the optimal solution value obtained by solving the linear programming relaxation of the formulation (8)-(18), which explicitly considers the original project scheduling variables in the master problem.
- LP_{impl} : the optimal solution value obtained by solving the linear programming relaxation of the formulation (19)-(22), which implicitly considers the project scheduling variables in the master problem via workload pattern variables. Note that LP_{impl} is equal to LB_{impl} since the proposed procedure relies on this problem formulation.

Both these lower bounds are obtained via column generation. Table 6 displays the results for this comparison, i.e. **the values of the lower bound** resulting from the linear programming relaxations and their relative difference ($\Delta = \frac{LP_{impl} - LP_{expl}}{LP_{expl}}$).

	LP_{expl}	LP_{impl}	Δ
$ N = 10$	413.5	475.7	+15.0%
$ N = 20$	870.3	944.7	+8.5%
$ N = 30$	1346.3	1429.9	+6.2%

Table 6: Comparison of different linear programming relaxations

Table 6 clearly shows the higher quality of LP_{impl} , meaning that a stronger lower bound can be obtained by taking the project scheduling constraints implicitly into account. This clearly validates the choice for the proposed formulation (19)-(22). A stronger lower bound improves the insight in the optimality gap and will reduce the number of subproblems created and the size of the branching tree.

5.4.2 Quality of the incumbent solutions

We benchmark the incumbent solutions obtained by the proposed methodology with several other optimisation methodologies:

- BP_{impl} : The exact branch-and-price methodology as proposed in Section 4.
- BP_{expl} : An exact branch-and-price methodology thriving on a problem formulation with the original project scheduling variables explicitly included in the master problem formulation (cf. eqs. (8)-(18)). The master problem considers only a restricted set of personnel patterns and a pricing problem is defined (cf. eqs. (23)-(30)). This procedure applies the branching rules and acceleration mechanisms proposed by Maenhout and Vanhoucke (2016) and includes appropriate mode assignment branching rules.
- RBB_{expl} : A restricted branch-and-bound, which solves the integer problem formulation (8)-(18) taking a restricted set of personnel patterns into account. The set of patterns is determined similarly as done by Maenhout and Vanhoucke (2016) to calculate an upper bound in the root node.
- ILS : The iterated local search proposed by Van Den Eeckhout et al. (2019).

When comparing incumbent solutions from exact and heuristic procedures, attention should be paid to the size of the instances and the stopping criterium. A heuristic

procedure is more likely to perform well on larger instances whereas an optimal procedure is more likely to have a better performance on smaller instances. For this reason, we benchmark the different procedures only on the set of instances with 30 activities. The stopping criterion for all procedures is a time limit, which is set at 1800 seconds.

Figure 2 compares the performance of the different procedures as a function of the CPU time (in seconds). Figure 2(a) shows the behaviour over time of the average solution quality of the incumbent solution. Figure 2(b) shows the percentage of the instances for which the proposed procedure finds better results compared to the ILS. Note that the scale of the x-axis is adapted in both figures to better present the results.

Figure 2: Benchmarking with other dedicated optimisation approaches

Proposed procedure versus BP_{expl} and RBB_{expl}

Figure 2(a) reveals a stagnation of the quality of the incumbent solutions obtained by BP_{expl} after less than 300 seconds, while the proposed procedure (BP_{impl}) continuously finds better solutions within the postulated time limit. The former is explained by the large branching tree visited due to the incorporation of the original project scheduling variables in the restricted master problem and the corresponding branching decisions. The performance of the proposed procedure, BP_{impl} , is thus far superior at each moment in time, with a deviation of 18.5% after 1800 seconds.

In contrast to BP_{expl} , RBB_{expl} continuously leads to better solutions. However, RBB_{expl} only finds a solution with a quality similar to the average initial upper bound of BP_{impl} after 737 seconds. This shows that the initial upper bound found by the proposed procedure, based on a restricted branch-and-bound with workload pattern variables, has a far better performance both in terms of quality and computational time.

In conclusion, the proposed procedure (BP_{impl}), which takes the project scheduling constraints implicitly into account in the mathematical problem formulation, leads to a higher optimal LP solution value, a better initial upper bound value based on a restricted brand-and-bound and, finally, better incumbent solutions resulting in a smaller optimality gap.

Proposed procedure versus ILS

Figure 2(a) also displays the performance of the *ILS* heuristic. Due to the computational expensive population generation in the *ILS* algorithm, the initial incumbent solution for every instance can only be calculated after 305 seconds. Therefore, the average solution quality cannot be calculated until that moment and the initial behaviour of the *ILS* heuristic, i.e. the sharp decrease in the average solution quality before 305 seconds, cannot be displayed. The solution quality of the incumbent solution gradually improves but stagnates from 600 seconds onwards. Further, we observe that the *ILS* outperforms BP_{impl} when the runtime is smaller than 1256 seconds. After 1256 seconds, the proposed procedure (BP_{impl}) still finds better incumbent solutions, whereas the average quality of the *ILS* has stagnated. As a result, BP_{impl} outperforms *ILS* from this moment on. Figure 2(b) shows a more nuanced view on the comparison between the *ILS* heuristic and the proposed procedure. In this graph, we compare the results from individual instances leading to three observations. First, in the initial stages of both algorithms ($CPU \leq 30$ seconds), we notice that BP_{impl} is able to find better solutions for 91.1% of the instances. This is explained by the good performance of the initial upper bound of BP_{impl} and the quasi-random generation of the population in the *ILS* procedure. Second, when the runtime is between 30 and 120 seconds, the percentage of instances for which *ILS* finds a better solution increases, leading to the point where BP_{impl} is only for 40.9% of the instances better than *ILS*. Third, after 120 seconds, the percentage of instances for which BP_{impl} finds a better solution increases again. After 1800 seconds, BP_{impl} finds a better incumbent for 85.5% of the instances.

6 Conclusion

In this paper, a project staffing problem is discussed that incorporates the project scheduling problem with discrete time/resource trade-offs to lower the staffing budget. In order to determine accurately the minimum required staffing budget, we construct a baseline personnel roster for regular and temporary workers taking time-related schedul-

ing constraints into account. We propose a branch-and-price methodology thriving on a column generation procedure that is defined by a restricted master problem with a limited number of personnel and workload patterns, a personnel pricing problem and a workload pricing problem. In addition, the search space is decomposed in order to accelerate the search for the optimal integer solution. The main idea is to translate the strategic budgeting problem into multiple tactical scheduling subproblems with a stipulated workforce size. For each subproblem, a branch-and-price is applied to search for a feasible baseline personnel roster. The stipulation of the staffing composition in advance allows the inclusion of resource capacity cuts in the workload pricing problem to find highly relevant workload patterns. The computational experiments show the contribution of both the search space decomposition and the defined resource capacity cuts. Furthermore, we present a search strategy which gradually increases the lower bound besides searching for a better upper bound. In addition, we show the superior performance of the proposed procedure by providing a benchmark with different other procedures.

To summarise, our contribution is fourfold. First, we propose a new mathematical formulation thriving on a personnel subproblem and a workload subproblem for the integrated project and project staffing problem with discrete time/resource trade-offs, leading to a tighter linear programming relaxation. Second, we are the first to propose an exact procedure for the problem under study outperforming existing approaches. Third, via well-devised experiments we show the contribution of the search space decomposition and the devised cuts introduced in the workload pricing problem to accelerate the search for the optimal solution. Fourth, by defining a search strategy based on the lower bound values, we obtain knowledge about the final optimality gap.

Adding the proposed cuts increases the complexity of the workload pricing problem and therefore the algorithm spends most time in solving these problems. Future research can focus on heuristics or a reformulation of the workload pricing problem to exploit a special underlying structure in order to improve the performance of the overall procedure when the number of activities increases. In addition, we aim to verify if the proposed methodology is effective in other environments where demand planning needs to be integrated into a staffing problem.

Acknowledgements The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO and the Flemish Government - department EWI.

References

- Alfares, H. K. (1998). An efficient two-phase algorithm for cyclic days-off scheduling. *Computers & Operations Research*, 25(11):913–923.
- Alfares, H. K. (2004). Survey, categorization, and comparison of recent tour scheduling literature. *Annals of Operations Research*, 127(1-4):145–175.
- Alfares, H. K. and Bailey, J. E. (1997). Integrated project task and manpower scheduling. *IIE transactions*, 29(9):711–717.
- Alfares, H. K., Bailey, J. E., and Lin, W. Y. (1999). Integrated project operations and personnel scheduling with multiple labour classes. *Production Planning & Control*, 10(6):570–578.
- Artigues, C., Gendreau, M., Rousseau, L.-M., and Vergnaud, A. (2009). Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound. *Computers & Operations Research*, 36(8):2330–2340.
- Baker, K. R. (1976). Workforce allocation in cyclical scheduling problems: A survey. *Journal of the Operational Research Society*, 27(1):155–167.
- Baker, K. R. and Magazine, M. J. (1977). Workforce scheduling with cyclic demands and day-off constraints. *Management Science*, 24(2):161–167.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329.
- Beaumont, N. (1997). Using mixed integer programming to design employee rosters. *Journal of the Operational Research Society*, 48(6):585–590.
- Bechtold, S. E., Brusco, M. J., and Showalter, M. J. (1991). A comparative evaluation of labor tour scheduling methods. *Decision Sciences*, 22(4):683–699.
- Beliën, J. and Demeulemeester, E. (2008). A branch-and-price approach for integrating nurse and surgery scheduling. *European journal of operational research*, 189(3):652–668.
- Beliën, J., Demeulemeester, E., De Bruecker, P., Van den Bergh, J., and Cardoen, B. (2013). Integrated staffing and scheduling for an aircraft line maintenance problem. *Computers & Operations Research*, 40(4):1023–1033.
- Billionnet, A. (1999). Integer programming to schedule a hierarchical workforce with variable demands. *European Journal of Operational Research*, 114(1):105–114.
- Brucker, P., Drexl, A., Möhring, R., Neumann, K., and Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European journal of operational research*, 112(1):3–41.
- Brunner, J. O., Bard, J. F., and Köhler, J. M. (2013). Bounded flexibility in days-on and days-off scheduling. *Naval Research Logistics (NRL)*, 60(8):678–701.
- Burns, R. N. and Carter, M. (1985). Work force size and single shift schedules with variable demands. *Management Science*, 31(5):599–607.
- Burns, R. N. and Koop, G. (1987). A modular approach to optimal multiple-shift manpower scheduling. *Operations Research*, 35(1):100–110.
- Caprara, A., Monaci, M., and Toth, P. (2003). Models and algorithms for a staff scheduling problem. *Mathematical programming*, 98(1):445–476.
- De, P., Dunne, E. J., Ghosh, J. B., and Wells, C. E. (1995). The discrete time-cost tradeoff problem revisited. *European Journal of Operational Research*, 81(2):225–238.
- Demeulemeester, E., REYCK, B. D., and Herroelen, W. (2000). The discrete time/resource trade-off

- problem in project networks: a branch-and-bound approach. *IIE transactions*, 32(11):1059–1069.
- Dohn, A. and Mason, A. (2013). Branch-and-price for staff rostering: An efficient implementation using generic programming and nested column generation. *European Journal of Operational Research*, 230(1):157–169.
- Eitzen, G., Panton, D., and Mills, G. (2004). Multi-skilled workforce optimisation. *Annals of Operations Research*, 127(1-4):359–372.
- Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1):3–27.
- Hartmann, S. and Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 127(2):394–407.
- Kolisch, R. and Sprecher, A. (1997). Psplib-a project scheduling problem library: Or software-orsep operations research software exchange program. *European journal of operational research*, 96(1):205–216.
- Maenhout, B. and Vanhoucke, M. (2016). An exact algorithm for an integrated project staffing problem with a homogeneous workforce. *Journal of Scheduling*, 19(2):107–133.
- Maenhout, B. and Vanhoucke, M. (2017). A resource type analysis of the integrated project scheduling and personnel staffing problem. *Annals of Operations Research*, 252(2):407–433.
- Mercier, A., Cordeau, J.-F., and Soumis, F. (2005). A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451–1476.
- Mesquita, M., Moz, M., Paias, A., and Pato, M. (2015). A decompose-and-fix heuristic based on multi-commodity flow models for driver rostering with days-off pattern. *European Journal of Operational Research*, 245(2):423–437.
- Mesquita, M. and Paias, A. (2008). Set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem. *Computers & Operations Research*, 35(5):1562–1575.
- Möhring, R. H. (1984). Minimizing costs of resource requirements in project networks subject to a fixed completion time. *Operations Research*, 32(1):89–120.
- Ranjbar, M., De Reyck, B., and Kianfar, F. (2009). A hybrid scatter-search for the discrete time/resource trade-off problem in project scheduling. *European Journal of Operational Research*, 193:35–48.
- Ranjbar, M. R. and Kianfar, F. (2007). Solving the discrete time/resource trade-off problem in project scheduling with genetic algorithms. *Applied Mathematics and Computation*, 191(2):451–456.
- Rasmussen, M. S., Justesen, T., Dohn, A., and Larsen, J. (2012). The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3):598–610.
- Ryan, D. M. and Foster, B. A. (1981). An integer programming approach to scheduling. *Computer scheduling of public transport urban passenger vehicle and crew scheduling*, pages 269–280.
- Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385.
- Van Den Eeckhout, M., Maenhout, B., and Vanhoucke, M. (2019). A heuristic procedure to solve the project staffing problem with discrete time/resource trade-offs and personnel scheduling constraints. *Computers & Operations Research*, 101:144–161.
- Vanderbeck, F. and Wolsey, L. A. (1996). An exact algorithm for ip column generation. *Operations research letters*, 19(4):151–159.

- Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., and Tavares, L. V. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research*, 187(2):511–524.
- Vanhoucke, M. and Debels, D. (2007). The discrete time/cost trade-off problem: extensions and heuristic procedures. *Journal of Scheduling*, 10(4-5):311–326.
- Verbeeck, C., Van Peteghem, V., Vanhoucke, M., Vansteenwegen, P., and Aghezzaf, E.-H. (2017). A metaheuristic solution approach for the time-constrained project scheduling problem. *OR spectrum*, 39(2):353–371.
- Weglarz, J., Józefowska, J., Mika, M., and Waligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes—a survey. *European Journal of Operational Research*, 208(3):177–205.
- Zamorano, E. and Stolletz, R. (2017). Branch-and-price approaches for the multiperiod technician routing and scheduling problem. *European Journal of Operational Research*, 257(1):55–68.