

# The Workforce Routing and Scheduling Problem: solving real-world Instances

Gabriel Volte  
LIRMM, University of Montpellier,  
CNRS  
Montpellier, France  
gabriel.volte@lirmm.fr

Chloé Desdouits  
DecisionBrain  
Montpellier, France  
chloe.desdouits@decisionbrain.com

Rodolphe Giroudeau  
LIRMM, University of Montpellier,  
CNRS  
Montpellier, France  
rodolphe.giroudeau@lirmm.fr

## ABSTRACT

We propose an efficient method to solve a workforce routing and scheduling problem with working constraints, and a bounded execution time limit. This problem combines two fundamental problems in operations research: routing and scheduling. In such a context, we develop a column generation algorithm, as a set partitioning problem with side constraints, within a branch-and-price framework. The pricing sub-problem is an elementary shortest path with resource constraints modeled with constraint programming. In our branch-and-price framework, we first solve our problem using branch-and-price and a branch-and-bound strategy is proposed on the last restricted master problem, in order to obtain a feasible solution when the time limit is almost reached. However, we show that the developed method leads to better solutions than using constraint programming or large neighborhood search methods. We show the relevance of our method with various-size real instances.

## INTRODUCTION

We consider in this paper a hybrid problem in which it is necessary to associate the vehicle optimization problem with an assignment problem for employees to satisfy some specific technical constraints. The study of this problem is motivated by taking into account new business constraints for employees with specific skills. These problems are more and more present in the everyday life of maintenance companies. The main difficulty is to consider the various parameters to respond to real situations.

Workforce Scheduling and Routing Problem (hereafter *WSRP*) represents problems that mobilize workforce to perform tasks for customers. Given a set of employees and a set of tasks to be scheduled, *WSRP* consists in assigning tasks to employees in order to fulfill some constraints while minimizing operational costs.

*WSRP* combines the complexity of scheduling problems [2, 18]:

- Multi-skill Project Scheduling Problem, MPSP [6, 14, 21] (Technician and Task Scheduling Problem).
- Sequencing and Scheduling Problem, SSP [19],
- Project Scheduling with Resources Constrained Scheduling Problem,

and problems of vehicle routing [20, 25]:

- Vehicle Routing Problem with Time Windows [23],
- Vehicle Routing Problem with Time Windows and Dependencies,

Figure 1 represents the successive generalizations of basic scheduling and routing problems, such as the *TSP*, that lead to the *WSRP* class of problems.

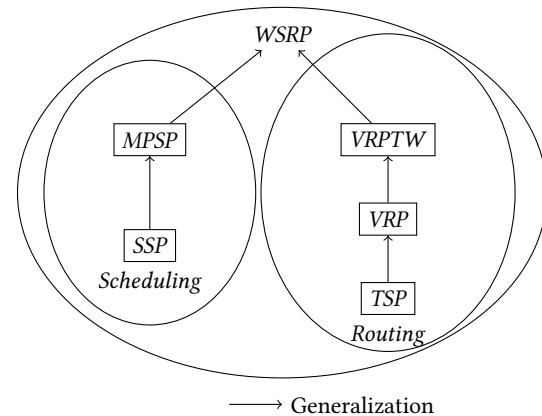


Figure 1: Description of hierarchical complexity class for *WSRP*.

This paper is organized as follows: the section **RELATED WORK** gives an overview of the previous works found in the literature on the *WSRP*, the section **MODELLING** formally describes our problem and a first compact model using integer linear programming (henceforth ILP) is given. The column generation decomposition and the branch-and-price scheme implemented in section **THE BRANCH-AND-PRICE FRAMEWORK**. The results and instances are presented in section **TESTS**. The last section concludes the paper and presents some future work.

## RELATED WORK

In the next section, we formally define the Workforce Scheduling and Routing Problem class, based on the survey [3]. This survey first presents the common characteristics of technicians and tasks, summarized in Table 1, then reviews known methods to solve problems considered as *WSRP*. The main method used to tackle these problems is a hybrid approach combining exact methods, integer linear programming or constraint programming, and heuristics/meta-heuristics methods, large neighborhood search or tabu search. The branch-and-price approach is also used since this approach is known to be efficient on routing problems and scheduling problems. This survey also gives a detailed computational study outlining the computational difficulties to solve these problems. This study has been carried out on different data sets with different integer linear programming formulations.

We describe some characteristics presented in Table 1. The processing time of the tasks is not negligible compared to the travel time and may depend on the employee. Tasks have required skills to filter employees who can perform them. A task can be processed by one or more employees, in which case all employees must be present before the starting time of the task. Castillo et

al. [4] and Rasmussen et al. [22] define temporal dependencies among tasks. Thus, some tasks admit a priority over others. Tasks can have priority meaning that a task should be performed before others.

Some tasks can be outsourced. In addition, the schedule of the employees can vary: it can be daily, weekly, etc. In general, *WSRP* instances are too big to be exactly solved. They are usually divided into smaller geographical areas to prevent an employee from working far from home but also to reduce the size of instances making it easier to solve.

Employee	Task
Means of transport	Processing time
Starting Position	Position
Ending position	Temporal dependence
Working Hours	Opening hours
Team	Required Skills
Skills	Priority
	Outsourcing

**Table 1: Characteristics of employees and tasks.**

For example, we may consider a set of employees who have to execute a set of tasks. Employees can travel by car, bicycle or public transport to perform the tasks. Employees are allowed to start and end their day from home. In the literature, there are numerous works of surveys aimed at characterizing and classifying the various problems belonging to the *WSRP* class. Based on an extension of the classic notation scheme  $\alpha | \beta | \gamma$  proposed by [13], Desrochers [7] develop a classification of *WSRP*.

An extensive overview of time constraint routing and scheduling problems during the last decades is given in Desrosiers et al. [8]. They detail ILP models and algorithms (column generation and dynamic programming) for each variation of problems (TSPTW, SDVRPTW, MDVRPTW, etc), focus their work on optimization methods for practical size instances. Although, they also present heuristic methods to solve complex problems or large-scale instances when optimal solutions are too difficult to obtain. The survey [24] outlines the research on different routing problems with time windows (M-TSPTW, SPPTW, etc) and give hints for future works on these problems.

To solve problems belonging to *WSRP*, we observe in literature many methods such as exact methods (constraint programming or integer linear programming), meta-heuristics (simulated annealing, tabu search, genetics, ...) or hybrid methods. Regarding exact methods, one can find ILP models and column generation using Dantzig-Wolfe decomposition. The master problem corresponds to a set partitioning problem [1] and the sub-problem to an Elementary Shortest Path Problem with Time Windows [12, 15] which is known to be  $\mathcal{NP}$ -hard [10].

## MODELLING

The goal of the project is to assign maintenance tasks to technicians in order to build daily schedules while optimizing some criterion such as quality of service, travel time, productivity and efficiency. The time limit is bounded to at most one hour for the biggest instances. The number of tasks is too large to schedule all of them in one day, thus tasks can be postponed. Thus the set of tasks is updated every day according to previous schedules.

The problem can be stated as follows: let us define  $\mathcal{P}$  the set of technicians and  $\mathcal{T}$  the set of tasks. For each technician, we

add two artificial tasks: one for the starting point ( $0^p$ ) and the other for the ending point ( $n^p$ ). Therefore, we define the set  $\mathcal{T}^p = \mathcal{T} \cup \{0^p, n^p\}$  for each technician  $p$ .

Let  $p_j$  be the processing time of task  $j$ , and let  $d_j$  be the due date of task  $j$ ,  $\omega_j$  is the weight (or revenue) of task  $j$ .

Let  $q$  be the number of skills and  $l$  the number of level of skills. Consider  $\alpha^p = (\alpha_1^p, \alpha_2^p, \dots, \alpha_q^p)$  be the skill vector of technician  $p$  and  $\beta^j = (\beta_1^j, \beta_2^j, \dots, \beta_q^j)$  the skill vector of task  $j$ . For each  $i \in \{0, \dots, q\}$ ,  $\alpha_i^p$  and  $\beta_i^j$  indicate the level (value in  $\{0, \dots, l\}$ ) of the  $i$ th skill in the vector.

Each task possesses a location and each technician have a starting location and ending location. Let  $M$  be the distance matrix where  $m_{i,j}$  represents the distance between locations  $i$  and  $j$ .

Let  $\bar{\mathcal{K}}^p$  (resp.  $\mathcal{K}^p$ ) be the set of unavailable (resp. available) periods of technician  $p$ . The previous notation is extended to task  $j$  with  $\bar{\mathcal{K}}^j$  and  $\mathcal{K}_j$ .  $\mathcal{K}_j^p$  denotes the set of time windows where technician  $p$  and task  $j$  are both available,  $\mathcal{K}_j^p = \mathcal{K}_j \cap \mathcal{K}^p$ . We define  $[a_k, b_k] \in \mathcal{K}_j^p$  the  $k$ th time window of the set. A task cannot overlap unavailable period (no task should start or end during an unavailable period).

The beginning (resp. ending) of the workday of a technician is given by the starting time (resp. end time) of his working hours. Moreover, the technician cannot travel before his starting hours or after his ending hours. Lastly, if a technician arrives early to a customer waiting is allowed.

Our problem can be formulated as integer linear program given below. The routing variables  $x_{ijk}^p$  take value 1 if the technician  $p \in \mathcal{P}$  travels from task  $i \in \mathcal{T}^p$  to task  $j \in \mathcal{T}^p$  in the time window  $k \in \mathcal{K}_i^p$ , 0 otherwise; the scheduling variables  $t_i^p$  correspond to the time the technician  $p \in \mathcal{P}$  starts the task  $i \in \mathcal{T}^p$ ; the covering variables  $y_i$  take value 1 if the task  $i \in \mathcal{T}$  is unscheduled/uncovered and 0 if the task  $i$  is performed by a technician; the tardiness variables  $D_i$  correspond to the lateness of the task  $i \in \mathcal{T}$ . First, we introduce an ILP representing the backbone of our problem, then we will add the specific constraints (*same technician constraints* and *appointment constraints*).

$(\pi_1, \pi_2, \pi_3, \pi_4)$  are the weights of the different criteria in the objective function (Equation (1)). The first criterion corresponds to the number of unscheduled tasks, the second minimizes the technician's travel distances, the third computes the sum of the tasks tardiness and finally the fourth maximizes the skill gap between technicians and tasks.  $\mathcal{W}$  is the total of the weight of all tasks. The first criterion maximizes the weighted sum of tasks scheduled but we choose to minimize the weighted sum of unscheduled tasks, the weight of all tasks minus the weight of all unscheduled tasks gives the weight of all scheduled tasks.

We denote by *Prec*, *Same* and *App*, the set of pairs  $(i, j) \in \mathcal{T} \times \mathcal{T}$  for which a *precedence constraint*, a *same technician constraint*, and *appointment constraint* exists, respectively. *Precedence constraints* are defined below. *Same technician constraint* corresponds to a pair  $(i, j) \in \text{Same}$ , if technician  $p$  executes task  $i$  (resp  $j$ ) thus he is the only one who can perform task  $j$  (resp  $i$ ).

*Appointment constraints* enforce a task to be performed by a technician at a fixed time. These constraints appear when the customers require a specific technician or a specific time to perform a job. There are three kinds of appointment constraints:

- When task  $j$  is assigned to technician  $p$  (even if he does not have the required skills to perform it): when  $p$  should perform  $j$ ?

- When task  $j$  is assigned to time  $t$ : which technician should perform it?
- When task  $j$  is assigned to technician  $p$  and to time  $t$ : is  $j$  scheduled for  $p$  at time  $t$ ?

Model M1: Compact formulation

$$\begin{aligned} \text{Maximize } & \pi_1(\mathcal{W} - \sum_{t \in \mathcal{T}} \omega_t y_t) - \pi_2 \sum_{p \in \mathcal{P}} \sum_{i, j \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_i^p} x_{ijk}^p m_{i,j} \\ & - \pi_3 \sum_{j \in \mathcal{T}} \omega_j D_j + \pi_4 \sum_{p \in \mathcal{P}} \sum_{i, j \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_i^p} \sum_{s=1}^q x_{ijk}^p (\alpha^p(s) - \beta^i(s)) \end{aligned} \quad (1)$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} \sum_{j \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_i^p} x_{ijk}^p + y_i = 1 \quad \forall i \in \mathcal{T} \quad (2)$$

$$\sum_{j \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_{0p}^p} x_{0pj}^p = 1 \quad \forall p \in \mathcal{P} \quad (3)$$

$$\sum_{j \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_j^p} x_{jnp}^p = 1 \quad \forall p \in \mathcal{P} \quad (4)$$

$$\sum_{i \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_i^p} x_{ihk}^p - \sum_{j \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_h^p} x_{hjk}^p = 0 \quad \forall p \in \mathcal{P}, \forall h \in \mathcal{T} \quad (5)$$

$$\sum_{h \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_j^p} x_{jhk}^p (\alpha^p(i) - \beta^j(i)) \geq 0 \quad \forall j \in \mathcal{T}, \forall p \in \mathcal{P}, \forall i \in [1..q] \quad (6)$$

$$B_j y_i + \sum_{p \in \mathcal{P}} t_i^p + p_i \leq \sum_{p \in \mathcal{P}} t_j^p + B_i y_j \quad \forall (i, j) \in \text{Prec} \quad (7)$$

$$\sum_{j \in \mathcal{T}} \sum_{k \in \mathcal{K}_i^p} x_{ijk}^p a_k \leq t_i^p \leq \sum_{j \in \mathcal{T}} \sum_{k \in \mathcal{K}_i^p} x_{ijk}^p b_k \quad \forall i \in \mathcal{T}, \forall p \in \mathcal{P} \quad (8)$$

$$\begin{aligned} \sum_{j \in \mathcal{T}} \sum_{k \in \mathcal{K}_i^p} x_{ijk}^p a_k \leq t_i^p + \left( \sum_{j \in \mathcal{T}} \sum_{k \in \mathcal{K}_i^p} x_{ijk}^p \right) \cdot p_i \leq \sum_{j \in \mathcal{T}} \sum_{k \in \mathcal{K}_i^p} x_{ijk}^p b_k \\ \forall i \in \mathcal{T}, \forall p \in \mathcal{P} \end{aligned} \quad (9)$$

$$t_i^p + x_{ijk}^p (m_{i,j} + p_i) \leq t_j^p + (1 - x_{ijk}^p) \cdot B_k \quad \forall p \in \mathcal{P}, \forall i, j \in \mathcal{T}^p, \forall k \in \mathcal{K}_i^p \quad (10)$$

$$D_j \geq \sum_{p \in \mathcal{P}} t_j^p + p_j - d_j \quad \forall j \in \mathcal{T} \quad (11)$$

$$x_{ijk}^p \in \{0, 1\}, \quad \forall p \in \mathcal{P}, \forall i, j \in \mathcal{T}^p, \forall k \in \mathcal{K}_i^p \quad (12)$$

$$t_i^p \in \mathbb{N}, \quad \forall p \in \mathcal{P}, \forall i \in \mathcal{T}^p \quad (13)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \mathcal{T} \quad (14)$$

$$D_j \in \mathbb{N}, \quad \forall j \in \mathcal{T} \quad (15)$$

The Model M1 is inspired by the MIP model given by a home nursing problem [22] and VRPTW [9]. Our problem differs on a few constraints: the skills/qualifications of employees, the working hours of employees and precedence constraints (they have temporal dependencies constraints). In addition, one additional dimension is needed on the routing decision variables ( $x_{ijk}^p$ ) because it is necessary to know on which time window the task is scheduled. The relevance of this compact formulation is discussed in Section TESTS and presented in Table 3.

Constraints (2) ensure that each task is scheduled at most once. If a task  $i$  is not scheduled (i.e.  $\sum_{p \in \mathcal{P}} \sum_{j \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_i^p} x_{ijk}^p = 0$ ) then to satisfy the constraint, task  $i$  must be covered by  $y_i = 1$  and

the task is postponed to a another day. The flow constraints for each technician depicted in (3), (4) and (5) control that technicians must start (resp. finish) their shift at their start (resp. end) location and that the flow conservation is respected (i.e. if a technician arrives at a customer he musts leave it). The skill constraints (6) restrict the set of tasks allowed to be performed by a given technician. A task can only be performed by a technician who has the required skills. In the current implementation of these constraints, we force variables  $x_{ijk}^p$  to 0 when  $\alpha^p(s) - \beta^i(s) > 0$ ,  $s \in [0, \dots, q]$ . Constraints (7) give the precedence constraints among the tasks: let  $(i, j) \in \text{Prec}$ , task  $j$  can be performed only if task  $i$  is executed before. The temporal constraints (8) and (9) and (10) verify that the availability periods of tasks and technicians are respected in the schedule (no overlap with unavailability or travel periods). If a task  $j$  is not executed by technician  $p$ , constraint (8) forces  $t_j^p$  to 0.

Constraints (16)-(19) correspond to problem-specific constraints.

Constraints (16) model the same technician constraints: they ensure that if a technician performs one of the two tasks, then the second is either performed by the same technician or the task is not scheduled. Constraints (17), (18) and (19) represent pre-assignment constraints describe above. Constraints (17) ensure that the right technician performs the task or the task is not scheduled. Constraints (18) ensure that the task is scheduled at the right time or not at all. Constraints (19) ensure that the task is assigned to the right technician at the right time or that otherwise task is not performed.

$$\begin{aligned} \sum_{h \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_j^p} x_{ihk}^p + y_i = \sum_{h \in \mathcal{T}^p} \sum_{k \in \mathcal{K}_j^p} x_{jhk}^p + y_j \\ \forall (i, j) \in \text{Same}, \forall p \in \mathcal{P} \end{aligned} \quad (16)$$

$$\sum_{j \in \mathcal{T}} \sum_{k \in \mathcal{K}_j^p} x_{ijk}^p - (1 - y_i) = 0 \quad \forall (i, p) \in \text{App} \quad (17)$$

$$\sum_{p \in \mathcal{P}} t_i^p - (1 - y_i) \cdot t_i = 0 \quad \forall (i, t_i) \in \text{App} \quad (18)$$

$$t_i^p - (1 - y_i) \cdot t_i = 0 \quad \forall (i, p, t_i) \in \text{App} \quad (19)$$

## THE BRANCH-AND-PRICE FRAMEWORK

In this section, we will introduce a branch-and-price framework. First, we use a Dantzig-Wolfe decomposition on the compact formulation in order to model it as a set partitioning problem with side constraints. In a branch-and-price framework the problem is split into a master problem (hereafter MP) and a pricing sub-problem (henceforth PSP). The PSP generates new feasible schedules/routes for each technician. Given the set of all feasible technician schedules, the MP assigns a schedule to each technician such that a maximum of tasks is processed (c.f. the first criterion of the objective function). Since the set of feasible technician schedules can be very large, we restrict the MP to a subset of schedules to obtain a reasonable size problem (called Restricted Master Problem denoted by RMP). A feasible route for a technician begins at his starting location and ends at his ending location, and respects all constraints mentioned in Model M1.

### Master Problem

Consider  $\mathcal{S}^p$  the set of all feasible schedules for technician  $p$  (this set will be generated successively by the PSP). Let  $a_{is}^p = 1$  if task  $i$  is in Schedule  $s$  of technician  $p$  and 0 otherwise; let  $t_{is}^p$  be the starting time of Task  $i$  in schedule  $s$  of technician  $p$ . The

constant  $c_s^p$  represents the cost of the schedule  $s$  for technician  $p$ . In the compact formulation (cf. Model M1), the aim is to minimize delays, distances and maximize the skill gap between tasks and technicians. This cost is a variation of the objective function of the compact formulation (Equation (1)). The first criterion of the compact formulation objective function is separated from the others in the RMP objective function to enhance the linear relaxation of the RMP.

$$c_s^p = -\pi_2 \sum_{i,j \in \mathcal{T}} \sum_{k \in \mathcal{K}_j^p} x_{ijk}^p m_{i,j} - \pi_3 \sum_{j \in \mathcal{T}} \omega_j D_j + \pi_4 \sum_{i,j \in \mathcal{T}} \sum_{k \in \mathcal{K}_j^p} \sum_{s=1}^q x_{ijk}^p (\alpha^p(s) - \beta^i(s))$$

We introduce binary variables for the RMP: the scheduling variables  $\lambda_s^p$  take value 1 if schedule  $s$  is chosen for technician  $p$  and 0 otherwise; the covering variables  $\gamma_i$  take value 1 if task  $i$  is uncovered/unscheduled and 0 otherwise.

Model M2: Restricted Master Problem

$$\text{Max} \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}^p} \lambda_s^p c_s^p + \pi_1 \sum_{i \in \mathcal{T}} (1 - \gamma_i) w_i \quad (20)$$

$$\text{s.t.} \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}^p} a_{is}^p \lambda_s^p + \gamma_i = 1 \quad \forall i \in \mathcal{T} \quad (21)$$

$$\sum_{s \in \mathcal{S}^p} \lambda_s^p \leq 1 \quad \forall p \in \mathcal{P} \quad (22)$$

$$\gamma_i + \sum_{s \in \mathcal{S}^p} a_{is}^p \lambda_s^p = \sum_{s \in \mathcal{S}^p} a_{js}^p \lambda_s^p + \gamma_j \quad (23)$$

$$\forall (i,j) \in \text{Same}, \forall p \in \mathcal{P}$$

$$B_j \gamma_i + \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}^p} t_{is}^p \lambda_s^p + p_i \leq \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}^p} t_{js}^p \lambda_s^p + B_i \gamma_j \quad (24)$$

$$\forall (i,j) \in \text{Prec}$$

$$\lambda_s^p \in [0, 1] \quad \forall p \in \mathcal{P}, \forall s \in \mathcal{S}^p \quad (25)$$

$$\gamma_i \in [0, 1] \quad \forall i \in \mathcal{T} \quad (26)$$

Constraints (21) express the fact that each task must be executed or covered. Constraints (22) ensure that only one schedule is associated with a technician. The *same technician constraints* are modeled by constraints (23). The *same technician constraints* are only in the RMP because in the PSP these constraints are always checked (a PSP is solved for each technician). Constraints (24) model the *precedence constraints*. As *precedence constraints* among tasks are independent of the set of technicians, these constraints must be present in the RMP and the PSP. Constraints (25) (resp. (26)) indicate the domain of  $\lambda_s^p$  variables (resp.  $\gamma_i$ ).

For any primal solution of the RMP, we obtain a dual solution  $[\mathbf{u}, \mathbf{z}, \mathbf{l}, \mathbf{w}]$ , where  $\mathbf{u} = (u_i)_{i \in \mathcal{T}}$ ;  $\mathbf{z} = (z_p)_{p \in \mathcal{P}}$ ;  $\mathbf{l} = ((l_i, l_j))_{(i,j) \in \text{Prec}}$ ;  $\mathbf{w} = ((w_{ip}, w_{jp}))_{(i,j) \in \text{Same}, p \in \mathcal{P}}$  are the dual variables of constraints (21), (22), (24), (23) respectively. These dual variables are used in the PSP (cf. Equation (36)) to generate new improving routes for each technician.

## Pricing subproblem

In our case, the sub-problem generates feasible schedules/routes (that respect the constraints) for each technician, thus these routes are added to the RMP. The sub-problem aims to find feasible routes for a technician which improve the solution obtained in the RMP. We cannot consider technicians as a fleet of vehicles (they have almost no similar characteristics), thus we must solve

a sub-problem for each technician. The PSP is solved using constraint programming with the ILOG IBM Scheduler constraints and variables (for more information on those constraints and variables please refer to [16, 17]). The Pricing Sub-Problem (hereafter PSP) is the elementary shortest path problem with time windows (ESPPTW). It focuses on finding an improved schedule for a particular technician. Recall that ESPPTW is  $\mathcal{NP}$ -hard in a strong sense [10] (there is no hope to develop dynamic programming).

Since our problem is a maximization problem if the PSP objective function  $Z_{PSP} < 0$  (cf. Equation (35)) then the corresponding route is not improving the current solution. Adding it in the solution of the RMP would decrease the value of the objective function. So we add in the RMP all tours with a reduced cost ( $Z_{PSP}$ ) strictly positive to potentially increase the value of the objective function. Since the PSP is hard to solve, the optimization is terminated as soon as a tour with a strictly positive reduced cost is found. Thanks to the constraints propagators, constraint programming is effective to find a good feasible solution in a short time.

For any technician  $p$ , we construct the following constraint programming model. We introduce the interval variables  $X_i^I$ ,  $\forall i \in \mathcal{T}^p$  to model tasks scheduling time. The domain of these variables is either  $\{\perp\}$  (task is not processed) or the scheduling horizon (the scheduling time of the task). Let  $X_p^S$  refers to the sequence variable of technician  $p$ , the domain of this variable is a permutation of tasks interval variables:  $D(X_p^S) = \text{perm}(\{X_{p-i}^I \mid \forall i \in \mathcal{T}^p \cup \{X_{p0}^I, X_{pn}^I\}\})$ .

Model M3: Constraint programming

$$\text{NoOverlap}(X_p^S, M) \quad (27)$$

$$\text{first}(X_p^S, X_{p0}^I) \quad (28)$$

$$\text{last}(X_p^S, X_{pn}^I) \quad (29)$$

$$\begin{cases} pOf(X_j^I) \leq pOf(X_i^I) \\ \text{EndBeforeStart}(X_i^I, X_j^I) \end{cases} \quad \forall (i,j) \in \text{Prec} \quad (30)$$

$$\begin{cases} \text{ForbidExtent}(X_i^I, \mathcal{K}_p^i) \\ \text{ForbidStart}(X_{p0}^I, \mathcal{K}^p) \\ \text{ForbidEnd}(X_{pn}^I, \mathcal{K}^p) \end{cases} \quad \forall i \in \mathcal{T}, \quad (31)$$

$$X_T = \sum_{i \in \mathcal{T}} \max(0, d_i - \text{EndOf}(X_i^I)) \quad (32)$$

$$X_D = \sum_{(i,j) \in X_p^S} m_{i,j} \quad (33)$$

$$X_{SG} = \sum_{i \in \mathcal{T}} pOf(X_i^I) \times \sum_{s \in [1..q]} (\alpha^p(s) - \beta^i(s)) \quad (34)$$

$$\text{Max } Z_{PSP} = X_{SG} - X_T - X_D - f(\mathbf{u}, \mathbf{z}, \mathbf{l}, \mathbf{w}) \quad (35)$$

Equation (27) ensures that the tasks performed by  $p$  are not overlapping and respects the travel time matrix  $M$ . Equations (28) and (29) enforce the route to begin (resp. end) at the starting (resp. ending) location. The constraint  $pOf$  (meaning *presenceOf*, is used to know if a task is executed) and the constraint  $\text{EndBeforeStart}$  are both used to assure that precedence constraints (30) are satisfied. Equations (31) prevent tasks to be performed outside the technician and task time windows. The constraint  $\text{ForbidExtent}$  ensures that tasks are not overlapping an unavailability period (given by  $\mathcal{K}_p^i$  or  $\mathcal{K}^p$ ). The constraint  $\text{ForbidStart}$  (resp.  $\text{ForbidEnd}$ ) ensures that tasks begin before (resp. end after) an unavailability period. We restrict the domain of the variables to satisfy appointment constraints. In the objective function (cf. Equation (35)), the

variable  $X_{SG}$  computes the skill gap between technicians and tasks (cf. Equation (34)), the variable  $X_T$  computes the tasks tardiness (cf. Equation (32)) and the variable  $X_D$  computes the travel time/distance (cf. Equation (33)). The function  $f(\mathbf{u}, \mathbf{z}, \mathbf{l}, \mathbf{w})$  is dedicated to the cost associated with the dual variables  $[\mathbf{u}, \mathbf{z}, \mathbf{l}, \mathbf{w}]$  defined above.

$$f(\mathbf{u}, \mathbf{z}, \mathbf{l}, \mathbf{w}) = \sum_{i \in \mathcal{T}} pOf(X_i^I) \times u_i + z_p + \sum_{(i,j) \in Prec} (l_j \beta_i - l_i \beta_j) - \sum_{(i,j) \in Same} (w_{jp} \times pOf(X_j^I) - w_{ip} \times pOf(X_i^I)) \quad (36)$$

## Branching strategies

We based our branching strategy on the ones presented in [11]. This paper presents two rules for branching. The first one, «standard strategy» consists in branching on decision variables  $\lambda_s^p$ , this branching is not effective because it leads to an unbalanced branching tree. The second one, is the «natural strategy» consists in branching on flow variables  $x_{ij}$  (cf. Model M1) and decision variables  $y_i$ , we opt for this strategy because it leads to a more balanced tree and an easier PSP.

Branch-and-price is usually used to obtain optimal solution but with a lack of resources and because of the large-scale instances this method is neglected. Because of the time limit and the large-scale highly constrained instances, finding an optimal solution can be difficult. Our algorithm is based on the one used in [5]. The authors propose a column generation to obtain an optimal non-integer solution. Therefore, a branch-and-bound algorithm is applied to obtain an integer solution.

We enhance this method adding the branch-and-price framework to generate more heterogeneous routes. We solve the problem with the following branch-and-price scheme (cf. Figure 2). We first solve the problem using a standard branch-and-price framework, at each node of the branching tree RMP is solved using column generation. If an integer solution is found, the bound (the best feasible solution found) is updated, else we add a branching node. At the end of the time limit, if we reach it, we use branch-and-bound method on the last RMP to obtain an integer solution. If this solution is better than the best one found in the branch-and-price algorithm we keep it.

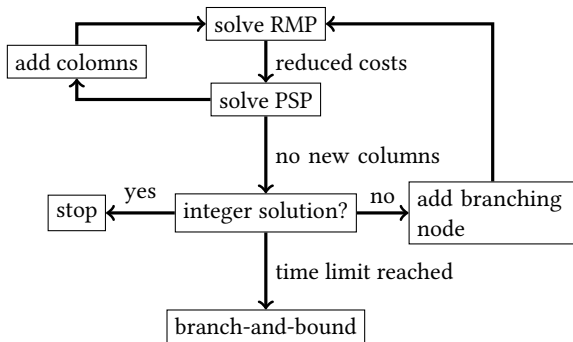


Figure 2: Illustration of our branch-and-price framework.

## TESTS

We have access to many instances of two Decisionbrain customers. Table 2 gives statistics for each instance. The name #30#256 (first column) means that the instance has 30 technicians

and 256 tasks, the column  $|Prec|$  gives the number of precedence constraints,  $|Same|$  shows the number of same technician constraints,  $|App|$  represents the number of precedence constraints,  $q$  denotes the number of skills (length of the skill vector),  $loc$  indicate the number of task and technician locations,  $\mathcal{K}^P$  (resp.  $\mathcal{K}_T$ ) shows the mean of technician time windows (resp. task time windows).

instance	$ Prec $	$ Same $	$ App $	$q$	loc	$\mathcal{K}^P$	$\mathcal{K}_T$
#30#256	0	0	0	154	162	0.96	1
#30#305	30	5	0	137	162	0.9	1
#30#2781	341	101	37	137	514	0.9	0.92
#144#1377	0	0	0	154	163	0.875	1
#145#4568	0	0	0	183	544	0.87	1

Table 2: The set of instances.

We are going to compare the branch-and-price method described here with the proposed ILP model solved using the software CPLEX and four other methods developed by Decision-Brain. The «ILP» corresponds to the integer linear program implemented and tested with Cplex. The «CP» corresponds to constraint programming using ILOG IBM Scheduler constraints and variables and tested with CPOptimizer. The «H» corresponds to a heuristic, kept confidential. The «H+X» method corresponds to start the X optimization with a first solution computed by the heuristic. The «LNS» corresponds to Large Neighborhood Search using the best insert algorithm on different neighborhood operators. The «BP» column corresponds to our branch-and-price scheme.

Instance	Model	Obj.	CPU(s)	Gap
#30#256	ILP	1.1379E7	1802	105%
#30#256	CP	2.1393E7	1803	9.9%
#30#256	H	2.1475E7	17	9.5%
#30#256	H+ILP	2.1475E7	1806	9.5%
#30#256	H+CP	2.1475E7	1803	9.5%
#30#256	H+LNS	2.1687E7	1643	8.5%
#30#305	ILP	2364770.0	1801	525%
#30#305	CP	1.3094E7	1802	17%
#30#305	H	1.1269E7	21	35%
#30#305	H+ILP	1.1278E7	1807	35%
#30#305	H+CP	1.3102E7	1802	16.9%
#30#305	H+LNS	1.3314E7	1678	15.1%

Table 3: Results for instances with 30 technicians and 256 tasks and 30 technicians and 305 tasks with a resolution time of 30 minutes.

The ILP model does not scale for the medium and large size instances, we obtain a high gap on the medium size instances (100% for instance#30#256 and 525% for instance#30#305). The CP model scales, and in some cases, achieves better results than the heuristic and meta-heuristic resolution method (H + LNS). One can see that the behavior of the CP is very close to the behavior of heuristics. Indeed, the CP obtains a good quality solution in a short time thanks to solver constraint propagators by cutting non-solution domain values. It is interesting to note that the heuristic gives a good solution in just a few seconds.

Now, we display the results obtained with the branch-and-price scheme. Table 5 gives the results for the different instances. In this table, CP is used to solve the PSP and the adopted tree traversal strategy is the Best-first search strategy in order to converge quickly towards a good solution.

Instance	Model	Obj.	CPU(s)	Status
#144#1377	CP	1.0995E7	3610	Feasible
#144#1377	H	1.4247E7	58	Feasible
#144#1377	H+CP	1.4360E7	3591	Feasible
#144#1377	H+LNS	1.4738E7	2877	Feasible
#145#4568	H	1.0421E8	221	Feasible
#145#4568	H+LNS	1.0645E8	3472	Feasible
#30#2781	CP	1.7181E7	3650	Feasible
#30#2781	H	2.3314E7	20	Feasible
#30#2781	H+CP	2.3343E7	3584	Feasible
#30#2781	H+LNS	2.3314E7	3636	Feasible

**Table 4: Results for large instances with a resolution time of 1 hour.**

The column «nodes» refers to the number of nodes browsed in the branch-and-price tree. The column «#col» represents the total number of columns in the master problem.

Instance	Model	Obj.	CPU(s)	nodes	#col	gap
#30#256	BP	2.1690E7	607.0	7	436	8.4%
#30#256	H+BP	2.1599E7	600.0	9	421	8.9%
#30#305	BP	1.3322E7	612.0	13	484	15%
#30#305	H+BP	1.3232E7	601.0	14	424	15,8%
#30#256	BP	2.0802E7	1812.0	15	565	13%
#30#256	H+BP	2.1712E7	1808.0	17	544	8.3%
#30#305	BP	1.3263E7	1809.0	13	606	15.5%
#30#305	H+BP	1.3251E7	1816.0	23	558	15.6%

**Table 5: Results for branch-and-price on medium-sized instances with a resolution time limit of 10 and 30 minutes using constraint programming for the PSP.**

One can observe that solutions obtained with the branch-and-price are better than solutions obtained with the constraint programming model and even than solutions computed by heuristic followed by the local search. However, as the ILP model, the branch-and-price does not scale. Large instances are too substantial to be treated by our branch-and-price scheme in a reasonable time. These results nevertheless show the interest in using the hybridization between column generation and constraint programming.

## CONCLUSION

In this paper, we propose a branch-and-price scheme dedicated to solving a *WSRP* problem in the presence of large-scale highly constrained real-world instances when the time limit is bounded. With this method, we were able to obtain good results, better than LNS or CP in some instances. However this method is not scalable, therefore results for large instances are missing.

Using a dynamic programming label algorithm to solve the sub-problem should speed the solving process up by adding multiples improving routes in the master problem at each step of the column generation algorithm while decreasing memory usage of each sub-problem. On the column generation phase, we solve a sub-problem for each technician, therefore solving all the sub-problem is time-consuming. One could try to group technicians that have some similar characteristics to reduce the time spent solving sub-problem.

## REFERENCES

- [1] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations research* 46, 3 (1998), 316–329.
- [2] J. Blazewicz, J.K. Lenstra, and A.H.G.R. Kan. 1983. Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics* 5, 1 (1983), 11–24. DOI: [http://dx.doi.org/10.1016/0166-218X\(83\)90012-4](http://dx.doi.org/10.1016/0166-218X(83)90012-4)
- [3] J.A. Castillo-salazar. 2014. Qu , Rong ( 2014 ) Workforce scheduling and routing problems : literature survey and computational study . *Annals of Operations Research* . pp . 1-29 . ISSN 1572- 239 (2014), 1–29.
- [4] J.A. Castillo-Salazar, D. Landa-Silva, and R. Qu. 2015. A greedy heuristic for workforce scheduling and routing with time-dependent activities constraints. *International Conference on Operations Research and Enterprise Systems ( ICORES 2015 )* (2015).
- [5] E. Choi and D.W. Tcha. 2007. A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research* 34, 7 (2007), 2080–2095.
- [6] J.F. Cordeau, G. Laporte, F. Pasin, and S. Ropke. 2010. Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling* 13, 4 (2010), 393–409. DOI: <http://dx.doi.org/10.1007/s10951-010-0188-7>
- [7] M. Desrochers, J.K. Lenstra, and M.W.P. Savelsbergh. 1990. A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research* 46, 3 (1990), 322–332. DOI: [http://dx.doi.org/10.1016/0377-2217\(90\)90007-X](http://dx.doi.org/10.1016/0377-2217(90)90007-X)
- [8] J. Desrosiers, Y. Dumas, M. Solomon, and F. Soumis. 1995. Time Constrained Routing and Scheduling. *Handbooks in Operations Research and Management Science* 8, C (1995), 35–139. DOI: [http://dx.doi.org/10.1016/S0927-0507\(05\)80106-9](http://dx.doi.org/10.1016/S0927-0507(05)80106-9)
- [9] A. Dohn, M.S. Rasmussen, and J. Larsen. 2011. The vehicle routing problem with time windows and temporal dependencies. *Networks* 58, 4 (2011), 273–289.
- [10] M. Dror. 1994. Note on the Complexity of the Shortest Path Models for Column Generation in VRPTW. 42 (10 1994), 977–978.
- [11] D. Feillet. 2010. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR: A Quarterly Journal of Operations Research* 8, 4 (2010), 407–424. <https://hal-emse.ccsd.cnrs.fr/emse-00505959>
- [12] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* 44, 3 (2004), 216–229.
- [13] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. 1979. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. *Annals of Discrete Mathematics* 5 (1979), 287–326.
- [14] C. A. J. Hurkens. 2009. Incorporating the strength of MIP modeling in schedule construction. *RAIRO - Operations Research* 43 (2009), 409–420. DOI: <http://dx.doi.org/10.1051/ro/2009026>
- [15] S. Irnich and G. Desaulniers. 2005. Shortest path problems with resource constraints. In *Column generation*. Springer, 33–65.
- [16] P. Laborie and J. Rogerie. 2008. Reasoning with Conditional Time-Intervals. *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference, May 15-17, 2008, Coconut Grove, Florida, USA* (2008), 555–560.
- [17] P. Laborie, J. Rogerie, P. Shaw, and P. Vilim. 2009. Reasoning with Conditional Time-Intervals. Part II: An Algebraical Model for Resources. *FLAIRS Conference* (2009), 201–206. <http://www.aaai.org/ocs/index.php/FLAIRS/2009/paper/viewPDFInterstitial/60>
- [18] B. J. Lageweg, J.K. Lenstra, E.L. Lawler, and A.H.G. Rinnooy Kan. 1982. Computer-Aided complexity classification of combinatorial problems. *Commun. ACM* 25, 11 (1982), 817–822. DOI: <http://dx.doi.org/10.1145/358690.363066>
- [19] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. 1993. Sequencing and scheduling: Algorithms and complexity. (1993), 445–522. DOI: [http://dx.doi.org/10.1016/S0927-0507\(05\)80189-6](http://dx.doi.org/10.1016/S0927-0507(05)80189-6)
- [20] J.K. Lenstra. 2013. Personnel scheduling: A literature review. (2013). DOI: <http://dx.doi.org/10.1016/j.cam.2008.10.038>
- [21] S. Pokutta and G. Stauffer. 2009. France Telecom workforce scheduling problem: A challenge. *RAIRO - Operations Research* 43, 4 (2009), 375–386. DOI: <http://dx.doi.org/10.1051/ro/2009025>
- [22] M.S. Rasmussen. 2010. The Home Care Crew Scheduling Problem: Preference-Based Visit Clustering and Temporal Dependencies. May (2010).
- [23] M. Solomon. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research* 35, 2 (1987), 254–265.
- [24] M. Solomon and J. Desrosiers. 1988. Survey Paper — Time Window Constrained Routing and Scheduling Problems. March 2014 (1988).
- [25] J.N. Tsitsiklis. 1992. Special cases of traveling salesman and repairman problems with time windows. *Networks* 22, 3 (1992), 263–282. DOI: <http://dx.doi.org/10.1002/net.3230220305>