CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

# Branch-and-Price for a Multi-Attribute Technician Routing and Scheduling Problem

**Ines Mathlouthi**
**Michel Gendreau**
**Jean-Yves Potvin**

**August 2017**

**CIRRELT-2017-56**

**Bureaux de Montréal :**
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

**Bureaux de Québec :**
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

# Branch-and-Price for a Multi-Attribute Technician Routing and Scheduling Problem

**Ines Mathlouthi[1,2], Michel Gendreau[1,3], Jean-Yves Potvin[1,2,*]**

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[3] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

**Abstract.** In this paper, we present an exact branch-and-price algorithm for a multi-attribute technician routing and scheduling problem. This problem integrates a number of distinctive features from a real-world application, like the management of an inventory of parts and multiple time windows for service. A new ternary branching scheme is introduced within the branch-and-price algorithm, based on the fact that not all tasks need to be performed. The computational results show that our algorithm can solve instances with up to 45 tasks and greatly improves upon a commercial solver applied to an arc-based mixed integer program.

**Keywords**: Technician routing and scheduling problem, multiple time windows, inventory of parts, branch-and-price, column generation, elementary shortest path with resource constraints, ternary branching.

**Acknowledgements.** Financial support for this work was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC). This support is gratefully acknowledged.



---

* Corresponding author: Jean-Yves.Potvin@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

# 1 Introduction

The Technician Routing and Scheduling Problem (TRSP) is a problem found in many real-world applications, although it has received only limited attention in the scientific literature. In this paper, we tackle a TRSP previously introduced in [8]. This problem is motivated by an application for the maintenance and repair of electronic transaction equipments. It can be defined as follows. There is a set of technicians with different skills who are available to carry out a number of different tasks. Each task has a priority level, depending on its emergency or the customer's importance. One must then assign tasks to the technicians and build a route for each technician, starting and ending at his home base location, so as to minimize an objective that accounts for overtime, total traveled distance, and total gain over the performed tasks. The solution must also satisfy constraints related to the technicians' skills required to perform their assigned tasks, working hours and multiple service time windows associated with each task.

Two particular features distinguish our problem from other TRSPs. First, the route schedules must account for technicians' breaks. Second, a task may consume one or more parts. Thus, an inventory of spare parts is carried by each vehicle and the technician must replenish (at most once) at a pre-assigned depot if the number of parts in the inventory does not allow all his assigned tasks to be done. The visit at the pre-assigned depot does not need to take place at the start of the route, as long as no stock out occurs. This visit can also be used to get special parts if they are needed for some tasks (at most one special part per task). As opposed to spare parts, special parts are not carried by technicians unless they are needed.

In [8], a mixed integer programming (MIP) model was developed for this problem and solved with a commercial solver. But this approach only allowed very small instances to be solved. This work is now aimed at developing a branch-and-price algorithm to allow larger instances to be solved exactly in reasonable computation times.

The rest of this paper is organized as follows. In Section 2, we review some related work. In Section 3, the problem is precisely defined. Then, the solution method is described in Section 4. Results on test instances of different sizes are reported and compared with those obtained with a commercial solver in Section 5. Finally, Section 6 concludes the paper.

# 2 Related Work

The work reported in the literature on the TRSP can be divided into two broad classes: static and dynamic TRSPSs. They are reviewed in the following.

## 2.1 Static TRSPs

Research on static TRSPs began with the work of Tsang and Voudouris [15] in 1997. The authors address a problem faced by the technician work force of British Telecom, using guided local search and a fast local search. In this problem, the service time is related to the technician's experience.

Weigel and Coo [16] report a problem faced by an American retailer. In this paper, tasks are first assigned to technicians and routes are then optimized individually using Or-Opt exchanges [9].

Xu and Chiu [17] consider a problem faced by service providers in the telecommunications industry. The objective in this work is to maximize the number of served tasks while taking into account task priority, technician skills and overtime. Four heuristics based on local search and GRASP are reported. An extension of this problem is considered in [4] by taking into account new features, like time windows. A customized biased random key genetic algorithm meta-heuristic is proposed to solve the problem

Blakeley et al. tackle a problem faced by the Schindler Elevator Corporation [1]. Their aim is to schedule periodic maintenance operations while taking into account technician skills and work regulations. Their sophisticated system employs various operations research techniques to assign maintenance work to technicians and construct their routes. Tang et al. [14] address a similar application and solve it with a tabu search heuristic.

The French Operations Research Society (ROADEF) proposed a challenge based on a problem encountered by France Telecom where routes for technicians must be scheduled on a multiple day horizon and where each task requires one or more skills. In this problem, teams of technicians must be created to serve the maximum number of tasks. Based on this challenge, Hashimoto et al. [7], developed a GRASP-based problem-solving methodology, while Cordeau et al. [3] used an adaptive large neighborhood search.

Another multi-period TRSP for a maintenance provider specialized in electric forklifts is solved with a branch-and-price algorithm in Zamorano and Stolletz [18]. In this application, the technicians are paired into teams to perform tasks and the branching strategy exploits this particularity.

## 2.2 Dynamic TRSPs

In 2008, Bostel et al. [2] first addressed a dynamic TRSP faced by Veolia, a water treatment and distribution company. In this problem, technician routes must be planned over a period of one week for repair or maintenance. The tasks to be scheduled can either be known in advance (preventive maintenance) or can occur dynamically. Each task has a time window for service. The first proposed method is a memetic algorithm, which is first applied to static tasks to produce

tours for every day of the week. Dynamic tasks are then integrated into the solution as they occur, using the memetic algorithm. The second approach is based on a column generation algorithm which can only be applied to problem instances of small size.

Pillac et al. [10] also address a TRSP in which a fraction of the tasks occurs dynamically. A parallel architecture is proposed to speed up the calculations. An initial solution is first created for known tasks using a regret heuristic [12]. This solution, is improved by an adaptive large neighborhood search [11]. The latter successively destroys (removes tasks) and repairs the current solution (reinserts tasks). When a new task occurs, the part of the solution that is already executed is fixed and the new task in incorporated into the solution by running again the ALNS for a limited number of iterations.

In this paper, we consider a multi-attribute TRSP that belongs to the class of static TRSPs. It will be described in the next section.

# 3 Problem Definition

In this section, we introduce the new TRSP variant proposed by a company that provides maintenance and repair of electronic transaction equipments. The three main entities involved in this problem are the following:

- Technicians: Each technician has skills which allow him to perform certain tasks and not others. His normal workday extends from 9H00 AM to 5H00 PM, if there is no overtime, and his route starts and ends at his home base. Furthermore, his route should not exceed a maximum traveled distance. Each technician can take three breaks during the day: one break of 15 minutes in the morning and afternoon, respectively, and a mid-day break of 30 minutes (where morning, afternoon and mid-day are defined through time intervals). Also, it is not possible for a technician to take a break just before returning to his home base. Each technician starts with an initial inventory of four different types of spare parts. Along the route, a pre-assigned depot can be used to replenish the inventory of spare parts and also get any needed special parts. A fixed capacity is associated with each type of spare parts stored in the vehicle.
- Tasks: It is assumed that tasks can be postponed, so there is no need to serve all tasks. Each task has a gain which stands for its priority. A task also has a service time and multiple possible time windows for its service. Finally, it is characterized by the types and number of spare parts of each type, as well as (possibly) one special part, needed by the technician to perform it;
- Depots: there are many depots with a (virtually infinite) number of parts.

The problem is to design technician routes for one day, where each route starts and ends at the technician's home location and serves a subset of tasks, including one possible stop at a preassigned depot, while satisfying the required skills for each task, the multiple time windows at each location, the time window for each break, the maximum traveled distance of each route, the required number of spare parts and, possibly, the requirement of a special part for each task. The objective is to maximize a weighted sum of the total gain over all performed tasks, (minus) the total traveled distance and overtime. An arc-based MIP for this problem can be found in [8]. In this model, a penalty term is also added in the objective to favor the insertion of breaks into the route schedules. In the problem considered here, a technician is forced to take a break if his schedule covers the time window associated with that break (which corresponds to the policy used in practice).

The number of variables in the MIP quickly increases with problem size. To alleviate this scalability issue, we propose here an alternative path-based formulation which is addressed through a branch-and-price algorithm, as explained in the next section.

# 4 Branch-and-Price

The branch-and-price algorithm for solving our TRSP is described below, by first introducing the column generation scheme and then the branching scheme(s).

## 4.1 Column generation

Column generation first requires the definition of a master problem. This is the topic of the first subsection. Then, we explain how column generation is realized through the definition of an appropriate pricing problem.

### 4.1.1 Master problem

The master is formulated as a set packing problem given that not all tasks need to be served. The decision variables (columns) in this model correspond to possible feasible routes for each technician. Note that there is a separate subset of columns for each technician because the home location, the skills and the preassigned depot are not necessarily the same. With each route is associated a cost. If $\delta_r$, $d_r$, $g_r$ and $w_r$ denote the overtime, distance, gain over served tasks and penalty for the exclusion of breaks, respectively, for route $r$ then the cost $c_r$ of this route can be written as:

$$c_r = \psi\delta_r + \vartheta d_r - \varphi g_r + \upsilon w_r \tag{1}$$

where $\psi$, $\vartheta$, $\varphi$ and $\upsilon$ are (normalized) weighting parameters.

To define the master problem, let us denote $I$ the set of tasks, $K$ the set of technicians, $R$ the set of all feasible routes and $R_k \subseteq R$ the subset of feasible routes for technician $k$. Also, parameter $a_{ir}$ is 1 if task $i$ is part of route $r$, 0 otherwise. The decision variables are $x_r$, $r \in R$, where $x_r$ is 1 if route $r$ is in the solution, and 0 otherwise. The model is then the following:

$$(MP) \quad \text{Min} \sum_{r \in R} c_r x_r \tag{2}$$

subject to

$$\sum_{r \in R} a_{ir} x_r \leq 1 \qquad i \in I \tag{3}$$

$$\sum_{r \in R_k} x_r \leq 1 \qquad k \in K \tag{4}$$

$$x_r \in \{0, 1\} \qquad r \in R \tag{5}$$

In this model, the objective is to minimize the total route cost. Constraints (3) ensure that each task is served by at most one route and constraints (4) ensure that each technician performs at most one route.

### 4.1.2 Pricing problem

In practice, the number of feasible routes $R$ is so large that it is not possible to directly solve the master problem presented above. Thus, we have to resort to column generation, where only a subset of feasible routes (columns) is considered in a so-called restricted master problem (RMP). Starting with an initial set of columns, the linear relaxation of the corresponding RMP is first solved to obtain values for the dual variables. With these dual values, new columns of negative reduced costs are identified by solving a pricing problem and are added to the RMP. The linear relaxation of the augmented RMP is then solved again to obtain new dual values. This is repeated until no column with negative reduced cost can be found. At this point, we have the optimal solution of the linear relaxation of the master problem, which is not necessarily integer.

The initial set of columns is obtained by solving the TRSP with a simple greedy insertion heuristic. The routes are constructed for each technician in turn. At each iteration, the route of the current technician is augmented by adding the unserved task that leads to the smallest increase to the objective value. This is repeated until no more feasible insertion in the route is possible.

Now, let us now denote $\lambda_i$ and $\mu_k$ the optimal values of the dual variables for constraints (3) and (4), respectively, and $I_r = \{i \in I : a_{ir} = 1\}$. Then, the

reduced cost $c'_r$ of route (column) $r$ is

$$c'_r = c_r - \mu_k - \sum_{i \in I_r} \lambda_i \tag{6}$$

In our column generation algorithm we have one pricing problem per technician to identify a route of negative reduced cost for this technician. The pricing problem corresponds to an elementary shortest path problem with resource constraints (ESPPRC) starting from the home location of the technician and back to it. Apart from the starting and ending home location of the technician, the network also comprises nodes for the pre-assigned depot, the subset of tasks that the technician can perform and the three breaks. The shortest path problem is solved using the well-known dynamic programming approach reported in [5]. In this algorithm, the nodes are assigned labels that represent partial paths used to reach them from the starting node. A label indicates the cost of the path as well as the consumption of resources (e.g., distance, time). More precisely, each label stores the following information on a partial path:

- $C$: cost;
- $T$: time (duration);
- $D$: distance;
- $SP^t$: number of remaining spare parts of type $t = 1, 2, 3, 4$;
- $F$: indicator set to 0 if the depot has already been visited, 1 otherwise.
- $V$: set of unreachable nodes;

It should be noted that a node is said to be unreachable if it has already been visited along the partial path, or if there is a resource such that its consumption prevents the node to be reached. In our case, a node (task) may be unreachable due to (1) the time windows (2) the maximum traveled distance or (3) some required part(s) are unavailable and the depot has already been visited.

The labels are used to reduce the number of partial paths that need to be considered during the execution of the dynamic programming algorithm. Basically, a dominance relation is established among the partial paths at a given node through their corresponding labels. If a path $p$ is dominated at some node $i$, then any extension to an immediate successor of $i$ will produce a path that will also be dominated. Thus, only non-dominated paths need to be considered. For two partial paths $p_1$ and $p_2$ from the home location of a technician to a given node $i$ with different labels $(C_1, T_1, D_1, SP_1, F_1, V_1)$ and $(C_2, T_2, D_2, SP_2, F_2, V_2)$, respectively, $p_1$ dominates $p_2$ if :

- $C_1 \leq C_2$;

- $T_1 \leq T_2$;
- $D_1 \leq D_2$;
- $SP_1^t \geq SP_2^t$ for spare parts of type $t = 1, 2, 3, 4$;
- $F_1 \geq F_2$
- $V_1 \subseteq V_2$

From an implementation point of view, we also store in the label the number of unreachable nodes $s$. It is used as a quick filter, since $s_1 \leq s_2$ is required for the condition $V_1 \subseteq V_2$ to be satisfied. Also, the subset of unreachable nodes is represented by a binary vector whose size corresponds to the number of nodes and each entry in the vector is one if the associated node is in the subset, 0 otherwise. Then, we have $V_1 \subseteq V_2$ if each entry in the vector for path $p_1$ is less than or equal than the corresponding entry in the vector for path $p_2$.

We regard to the technicians' breaks, the path extension goes as follows when going from $i$ to $j$. If we are in the time window of a given break at $j$ and this break has not been taken yet then (1) if it is not possible to insert the break after $j$ without exceeding the break's upper bound, then the break is inserted between $i$ and $j$ (if feasible) and (2) if the break can be inserted after $j$, the two following cases are accounted for: the technician goes directly from $i$ to $j$ or the break is inserted between $i$ and $j$ (if feasible). The penalty for not taking breaks prevents a route with breaks to be dominated by the same route without breaks (i.e., taking a break consumes time but it also reduces cost). Since a technician has to take a break when his schedule intersects with the break's time window, all non-dominated routes of negative reduced cost are filtered out at the end of the procedure to keep only those where the required breaks are taken by the technician.

#### 4.1.3 Decremental State-Space Relaxation

The column generation algorithm presented above can be improved by relaxing the elementary requirement in the ESPPRC pricing problem. This is known as Decremental State-Space Relaxation (DSSR) [13]. At each execution of the dynamic programming algorithm, the relaxation is tightened by considering nodes involved in a cycle as critical and by forbidding them to be visited more than once. The set of critical nodes thus grows from one execution to the next until an elementary path is obtained. With regard to the implementation, we have a vector $\varphi$ in the label for the set of unreachable and critical nodes. Then, the last condition $V_1 \subseteq V_2$ for two paths $p_1$ and $p_2$ in the previous dominance rule becomes:

- $\varphi_1 \subseteq \varphi_2$

As before, we also maintain the number of unreachable and critical nodes for quick filtering purposes.

## 4.2 Branching

As mentioned above, the column generation algorithm does not necessarily lead to an integer solution, given that a linear relaxation of the current RMP is solved at each iteration. Thus, branching may be required to identify an optimal integer solution. Column generation is first applied at the root node of the search tree. If the solution obtained is integer, then we have the optimal solution of our TRSP and we stop here. Otherwise, we must branch on a fractional variable to create two child nodes, each node being associated with a different subproblem. The search is then executed by selecting one open node (i.e., a node already generated but not solved yet) and by applying column generation to the corresponding subproblem. The algorithm stops when all open nodes are done. The best integer solution found during the search is then the optimal solution of our TRSP.

In the literature, a popular approach is to branch on an arc variable whose (fractional) value corresponds to the flow on that arc. By setting the variable to either 0 or 1, two new child nodes are generated. This additional constraint can be easily integrated into the master and pricing problems at each child node [6]. Unfortunately, the branch where the variable is set to 0 is very weak since it forbids only one arc. In this paper, we exploit the fact that not all tasks need to be performed.

Let us define $y_{ik} = 1$ if task $i$ is assigned to technician $k$, 0 otherwise. It should be noted that the value of $y_{ik}$ can be obtained by summing the variables $x_r$ in the RMP for which task $i$ is served in route $r \in R_k$. Two different branching strategies based on these variables are tested in the computational results:

*Binary branching*. Here, two child nodes are created when there is one or more fractional variables $y_{ik}$ in the solution. We first select the variable closest to 0.5 and set it to 1 on one branch and to 0 on the other branch. In the first case, when $y_{ik} = 1$, task $i$ is performed by technician $k$. Thus, after solving the pricing problem of technician $k$, we consider only non dominated routes of negative reduced cost that cover task $i$. In the second case, when $y_{ik} = 0$, task $i$ is not performed by technician $k$ (i.e., the task is either performed by another technician or not performed at all). Accordingly, there is no node for task $i$ in the pricing problem of technician $k$, which is much stronger than forbidding a single arc.

*Ternary branching*. Here, three child nodes are created when there is one or more fractional variables $y_{ik}$ in the solution. One branch stands for $y_{ik} = 1$, as in the binary branching. The second branch stands for $\sum_{k' \in K} y_{ik'} = 0$, that is, neither technician $k$ nor any other technician performs task $i$. Accordingly, there

is no node for task $i$ in the pricing problem of each technician. The third branch stands for $y_{ik} = 0$ and $\sum_{k' \in K \setminus \{k\}} y_{ik'} = 1$. Here, although technician $k$ does not perform task $i$, some other technician does. In this case, there is no node for task $i$ in the pricing problem of technician $k$ and the constraint $\sum_{r \in R \setminus R_k} a_{ir} x_r = 1$ replaces the constraint $\sum_{r \in R} a_{ir} x_r \leq 1$ in the master problem to force some other technician to perform the task.

One could wonder if an integer solution of our TRSP is guaranteed when all variables $y_{ik}$ are integer. This is indeed the case as demonstrated below.

**Proposition.** Let $X$ be a solution at a node of the branching tree. If there is no task shared by two or more technicians (i.e., $y_{ik}$ is either 0 or 1, $i \in I$, $k \in K$), then $X$ is integer.

**Proof.** Let us define $I_k$ the set of tasks served by technician $k \in K$ in solution $X$ and $R_k^* = \{r \in R_k : x_r > 0\}$ the set of routes of technician $k$ used in solution $X$. Since, by hypothesis, no task is shared by two or more technicians for each task $i \in I_k$, we have:

$$\sum_{r \in R_k^*} a_{ir} x_r = 1 \quad i \in I_k,\, k \in K; \tag{7}$$

$$\sum_{r \in R_k^*} a_{ir} x_r = 0 \quad \text{if } i \notin I_k,\, k \in K; \tag{8}$$

To prove that $X$ is integer, we show that all routes (columns) in $R_k^*$ are the same for any given technician $k \in K$, that is:

$$a_{ir} = 1 \quad i \in I_k,\, r \in R_k^*; \tag{9}$$

$$a_{ir} = 0 \quad i \notin I_k,\, r \in R_k^*; \tag{10}$$

If this is true, $R_k^*$ contains a single route $r$ for technician $k$ and the corresponding variable $x_r = 1$ (otherwise, we would have two identical columns in the basis). Hence, $X$ is integer.

By contradiction, let us suppose that equations (9) and (10) do not hold for some technician $k$. Thus, there is at least two routes, $r'$, $r'' \in R_k^*$, $r' \neq r''$, such that $a_{i^* r'} \neq a_{i^* r''}$ for at least one task $i^*$. Two cases are possible:

If $i^* \notin I_k$ then $a_{i^* r'} = 1$ or (exclusive) $a_{i^* r''} = 1$. Since $r'$, $r'' \in R_k^*$, we have $x_{r'} > 0$, $x_{r''} > 0$ and equation (8) cannot hold.

If $i^* \in I_k$ then, without loss of generality, let us suppose that $a_{i^* r'} = 1$ and $a_{i^* r''} = 0$. We define $R_k^{i^*} = \{r \in R_k^* | a_{i^* r} = 1\}$. By equation (7):

$$\sum_{r \in R_k^{i^*}} x_r = 1$$

and

$$x_{r''} + \sum_{r \in R_k^{i^*}} x_r > 1$$

because $r'' \in R_k^*$ and, consequently, $x_{r''} > 0$. But, it contradicts the fact that:

$$\sum_{r \in R_k^*} x_r \leq 1 \quad k \in K$$

since each technician can serve at most one route, QED.

# 5 Computational results

In the following, we report computational results obtained with our branch-and-price algorithm. First, the test instances are described. Then, the computational behavior of our algorithm on different types of test instances is reported. Finally, a comparison with the mathematical model in [8], when solved with CPLEX, is presented. It should be noted that the latter model was slightly modified to comply with the inventory constraints stated in Section 3, where a fixed vehicle capacity for each type of special parts should not be exceeded (in [8], the parts collected at the depot were simply added to those already present in a vehicle).

## 5.1 Test instances

The test instances were generated as in [8] and their characteristics are presented in Table 1. We have considered every possible combination of the basic parameter values shown in Table 2, for a total of $2 \times 2 \times 8 = 32$ subsets of instances, with 5 instances in each subset. To evaluate the performance of the tested problem-solving methods along various dimensions, other values were considered for the parameters Skills, Service time, Special part and # Technicians, as indicated in Table 3. When a new value is tested for one of these parameters, the other parameters remain fixed at their basic value. With regard to the skill configuration 50% (2 technicians) and 25% (1 technician), it should be noted that the second technician with 50% of all tasks automatically takes the tasks that cannot be performed by the first technician. In this way, every task can be served by at least one technician. Also, when the number of technicians is 4,

the fourth technician can perform 25% of all tasks. Thus, the skill configuration is 100% (1 technician), 50% (1 technician) and 25% (2 technicians).

Table 1: Characteristics of the test instances

| | |
|---|---|
| *Service area* | The service area corresponds to a 40km × 40km or 50km × 50km squared area. |
| *Depot location* | There are 3 depots randomly located within the service area. |
| *Task location* | Each task is randomly located in the service area. |
| *Task service time* | The service time of each task is randomly chosen between 30 and 45 minutes. |
| *Task gain* | The gain of each task is randomly chosen between 1 and 10. |
| *Technician home base* | The first two technicians are located at the opposite ends of the service area (along the diagonal). The other technicians are randomly located within the service area. |
| *Technician skills* | With each technician is associated the percentage of tasks that he can perform, which is chosen from 100%, 50% and 25%. |
| *Parts* | The number of spare parts needed to perform a task is randomly chosen between 0 and 3. Then, each spare part is assigned a type, among 4 different types. A special part is needed with a probability of 0.125 |
| *Time windows* | Both narrow and wide time windows are considered. The latter are twice as wide as the former on average. The length of a narrow time window is randomly generated between 60 and 90 minutes. The lower bound of the first time window is chosen randomly between 9H00 AM and noon. The lower bounds of the remaining time windows are randomly chosen from 2 to 3 hours after the upper bound of the previous time window until a maximum of 3 time windows are obtained. |

In all test instances, the crow fly distance is assumed between each pair of locations, the speed of the vehicles is set at 50km/h and the maximum distance of each technician route is set at 125km. With regard to the objective function, the weights are $\psi = 1$ (overtime in seconds), $\vartheta = 5$ (distance in kilometers), $\varphi = 500$ (gain), $\upsilon = 1000$ (number of excluded breaks, where the morning and afternoon breaks of 15 minutes count for 1 and the mid-day break of 30 minutes counts for 2). These weights are then normalized between 0 and 1. In particular,

this setting allows a technician to do some overtime to perform a high gain task and leads to more challenging instances than the weighting scheme proposed in [8].

Table 2: Basic parameter values

| | |
|---|---|
| Time windows | Narrow, Wide |
| Service Area | 40km × 40km, 50km × 50km |
| # Tasks | 10, 15, 20, 25, 30, 35, 40, 45 |
| # Technicians | 3 |
| Skills (% Tasks) | 100% (1 technician), 50% (1 technician), 25 % (1 technician) |
| Service time | 30-45 minutes |
| Special part (prob.) | 0.125 |

Table 3: New parameter values

| | |
|---|---|
| Skills 1 | 100% (3 technicians) |
| Skills 2 | 50% (2 technicians), 25% (1 technician) |
| Service time 1 | 15-30 minutes |
| Service time 2 | 10-20 minutes |
| Special part 1 (prob.) | 0 |
| Special part 2 (prob.) | 0.25 |
| # Technicians | 4 |

## 5.2 Experiments

This section reports the results obtained on the instances introduced in the previous section. Our branch-and-price algorithm was given a maximum of 24 hours of computation time on a 3.07GHz Intel Xeon X5675 processor. The linear relaxation of the restricted master problems were solved with CPLEX 12.6. The tables of results provide the following information :

- *Name*: Name of each subset of 5 instances using the format $X$-$Y$-$Z$, where $X$ is the size of the time windows (either $N$ for narrow or $W$ for wide), $Y$ the size of the service area (either 40 for 40km × 40km or 50 for 50km × 50 km) and $Z$ the number of tasks (either 10, 15, 20, 25, 30, 35, 40 or 45).
- *Opt*: Number of instances solved to optimality at the root (number of instances solved to optimality after branching);

- *CPU*: Average computation time in seconds (over instances solved to optimality); when the time limit of 24 hours is reached on every instance in a subset, the entry is marked with a "-" sign.
- *Nodes*: Average number of nodes in the branching tree (over instances solved to optimality), excluding the root; when the time limit of 24 hours is reached on every instance in a subset, the entry is marked with a "-" sign.

Table 4 reports the results obtained on the test instances generated with the basic parameter values shown in Table 2. We tested the branch-and-price algorithm with the standard pricing problem (ESPPRC) and the relaxed one (DSSR), using the binary and ternary branching schemes.

We first note that instances with up to 45 tasks can be solved to optimality with our algorithms. Also, the instances with narrow time windows and a service area of 50 km $\times$ 50 km are easier to solve because they are more constrained. In the latter case, it should be noted that the maximum distance constraint on each route plays a more significant role when the service area is larger. In general, DSSR provides a substantial improvement over ESPPRC by allowing more instances to be solved to optimality and by requiring less computation time. Although we do not show the details here, the improvement provided by DSSR with ternary branching over the other tested variants increases on more difficult instances obtained by using the parameter values shown in Table 3, for example when all technicians have the required skills to perform all tasks (see below).

Based on these results, we will focus in the following on the branch-and-price algorithm with DSSR and ternary branching. We will first examine the results obtained when a parameter, either Skills, Service time, Special part or # Technicians is set to a new value in Table 3, while fixing the other parameters to their basic value. In the corresponding tables of results, the following additional information is provided:

- *Tasks*: Average number of tasks performed by the technicians;
- *Idle*: Average number of idle (unassigned) technicians.

*Impact of special parts*

Table 5 reports the results obtained when the probability that a special part is required to perform a task is reduced from the basic value .125 to 0 or increased from .125 to .25. When no special part is required, there is no need to visit the depot before serving tasks and the solution structure is less constrained. Hence, the problem becomes more difficult. Accordingly, we observe a substantial increase in computation times in this case, although the number of instances solved to optimality does not change much.

*Impact of skills*

Table 6 reports the results obtained when we restrict or enlarge the subset of tasks that each technician can perform. We observed that this characteristic has the largest impact on the performance of the branch-and-price algorithm. For example, if we assume that every technician has all the required skills to perform all tasks, which corresponds to *Skills 1* in Table 3, the algorithm cannot solve instances with more than 30 tasks for the subsets with wide time windows and 35 tasks for the subsets with narrow time windows. Conversely, when the skills are reduced according to *Skills 2* in Table 3, all instances are solved to optimality in much less computation time. It is even possible to solve instances with 50 tasks, although these results are not reported here for brevity purposes.

*Impact of service times*

Table 7 reports the results when the service time is decreased with regard to the basic configurations. Instead of randomly choosing the time required to perform a task in the interval 15-30 minutes, the configurations *Service Time 1* and *Service Time 2* in Table 3 choose a time in the intervals 15-30 minutes and 10-20 minutes, respectively. It should be noted that reducing the service time increases the combinatorial complexity because each technician has now more flexibility for serving tasks, which is particularly helpful with narrow time windows. This is indicated in Table 7 by an increase in the number of performed tasks when going from 15-45 minutes to 10-20 minutes. Although there is no significant impact on the number of instances solved to optimality, the computation times also tend to increase with reduced service times.

*Impact of number of technicians*

Table 8 shows the results obtained when the number of technicians is increased from 3 to 4. The problem becomes more difficult, as shown by the computation times and the number of instances solved to optimality (19 fewer instances solved to optimality with 4 technicians). As expected, both the number of tasks performed by the technicians and the number of idle technicians increase.

Overall, the four parameters considered above have an impact on the performance of our branch-and-price algorithm, with the technicians' skills being the most sensitive.

| Name | ESPPRC | | | | | | DSSR | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Binary Branching | | | Ternary Branching | | | Binary Branching | | | Ternary Branching | | |
| | Opt | CPU | Nodes | Opt | CPU | Nodes | Opt | CPU | Nodes | Opt | CPU | Nodes |
| N-40-10 | 5 | 0.3 | 0 | 5 | 0.3 | 0 | 5 | 0.3 | 0 | 5 | 0.4 | 0 |
| N-40-15 | 2(3) | 0.9 | 4.7 | 2(3) | 0.9 | 9 | 2(3) | 1 | 3.3 | 2(3) | 1.1 | 93 |
| N-40-20 | 3(2) | 5.8 | 14 | 3(2) | 5.1 | 12 | 3(2) | 4.8 | 10 | 3(2) | 3.4 | 120 |
| N-40-25 | 2(3) | 1312.5 | 104.7 | 2(3) | 1289.6 | 12 | 2(3) | 53.6 | 24.7 | 2(3) | 44.8 | 282 |
| N-40-30 | 1(1) | 1223.5 | 1022 | 1(1) | 1189.2 | 282 | 2(1) | 85.1 | 30 | 2(1) | 69.4 | 363 |
| N-40-35 | 0(2) | 1518.3 | 1534 | 0(2) | 1498.2 | 363 | 0(2) | 334.2 | 126 | 0(2) | 241.4 | 2185.5 |
| N-40-40 | 1(1) | 1813.2 | 2046 | 1(1) | 1066.6 | 120 | 1(1) | 428 | 254 | 1(1) | 280.1 | 3279 |
| N-40-45 | 0(0) | - | - | 0(0) | - | - | 0(0) | - | - | 0(0) | - | - |
| N-50-10 | 4(1) | 0.5 | 2 | 4(1) | 0.6 | 3 | 4(1) | 0.5 | 2 | 4(1) | 0.7 | 3 |
| N-50-15 | 4(1) | 0.6 | 6 | 4(1) | 0.6 | 3 | 4(1) | 0.4 | 6 | 4(1) | 0.7 | 12 |
| N-50-20 | 5 | 1 | 0 | 5 | 1.1 | 0 | 5 | 0.5 | 0 | 5 | 0.3 | 0 |
| N-50-25 | 4(1) | 1.2 | 6 | 4(1) | 0.7 | 3 | 4(1) | 0.6 | 6 | 4(1) | 0.4 | 39 |
| N-50-30 | 4(0) | 20.3 | 0 | 4(1) | 5.2 | 12 | 4(1) | 6.7 | 14 | 4(1) | 4.6 | 39 |
| N-50-35 | 2(2) | 35.1 | 30 | 2(2) | 29.2 | 25.5 | 2(2) | 26.1 | 22 | 2(2) | 22.9 | 241.5 |
| N-50-40 | 3(0) | 35.1 | 0 | 3(0) | 34.8 | 0 | 3(1) | 33.6 | 30 | 3(1) | 30.4 | 363 |
| N-50-45 | 1(2) | 110.6 | 46 | 1(3) | 105.2 | 39 | 1(3) | 73.5 | 40.7 | 1(3) | 67.5 | 606 |
| W-40-10 | 4(1) | 0.5 | 2 | 4(1) | 0.5 | 3 | 4(1) | 0.5 | 2 | 4(1) | 0.3 | 12 |
| W-40-15 | 2(3) | 2.6 | 8.7 | 2(3) | 2.2 | 6 | 2(3) | 1 | 4.7 | 2(3) | 1 | 12 |
| W-40-20 | 4(1) | 14.9 | 14 | 4(1) | 14.4 | 12 | 4(1) | 14.5 | 14 | 4(1) | 13.8 | 39 |
| W-40-25 | 1(4) | 1557.6 | 1790 | 1(4) | 1408.7 | 302.3 | 1(4) | 95.2 | 38 | 1(4) | 91.8 | 909.8 |
| W-40-30 | 1(1) | 57.2 | 30 | 1(1) | 50.4 | 39 | 1(1) | 27.9 | 30 | 1(1) | 25.2 | 120 |
| W-40-35 | 1(0) | 3221.9 | 0 | 1(1) | 3422.4 | 3279 | 1(1) | 197 | 62 | 1(1) | 177.7 | 1092 |
| W-40-40 | 0(0) | - | - | 0(1) | 3554.7 | 3279 | 0(1) | 242.8 | 126 | 0(1) | 239.9 | 3279 |
| W-40-45 | 0(0) | - | - | 0(0) | - | - | 0(0) | - | - | 0(0) | - | - |
| W-50-10 | 4(1) | 0.5 | 2 | 4(1) | 0.6 | 3 | 4(1) | 0.6 | 2 | 4(1) | 0.6 | 3 |
| W-50-15 | 3(2) | 0.5 | 2 | 3(2) | 0.7 | 3 | 3(2) | 0.4 | 6 | 3(2) | 0.5 | 12 |
| W-50-20 | 4(1) | 1.7 | 6 | 4(1) | 1.9 | 12 | 4(1) | 1.7 | 14 | 4(1) | 1.7 | 12 |
| W-50-25 | 5 | 7.6 | 0 | 5 | 6.6 | 0 | 5 | 5.2 | 0 | 5 | 5.8 | 0 |
| W-50-30 | 4(0) | 18.8 | 0 | 4(0) | 17.7 | 0 | 4(0) | 17.6 | 0 | 5 | 15.1 | 0 |
| W-50-35 | 3(0) | 146.7 | 0 | 3(0) | 148.5 | 0 | 3(0) | 27.3 | 0 | 3(0) | 28.3 | 0 |
| W-50-40 | 0(2) | 293.6 | 94 | 0(3) | 280.8 | 66 | 1(2) | 180.6 | 94 | 1(3) | 178.7 | 2550 |
| W-50-45 | 2(0) | 440 | 0 | 2(1) | 354.8 | 120 | 2(0) | 237.9 | 0 | 2(1) | 233.8 | 3279 |

Table 4: Basic Configurations

| Name | No special parts (0) | | | | Basic configurations (.125) | | | | More special parts (.25) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Opt | CPU | Nodes | Tasks | Idle | Opt | CPU | Nodes | Tasks | Idle | Opt | CPU | Nodes | Tasks | Idle |
| N-40-10 | 4(1) | 0.5 | 2 | 9.8 | 1 | 5 | 0.4 | 0 | 9.8 | 1 | 5 | 0.3 | - | 9.8 | 1 |
| N-40-15 | 2(3) | 2.1 | 9 | 13.4 | 0.4 | 2(3) | 1.1 | 93 | 14 | 0.2 | 2(3) | 0.6 | 9 | 14.4 | 0.4 |
| N-40-20 | 2(3) | 11 | 21 | 16.4 | 0 | 3(2) | 3.4 | 120 | 17 | 0 | 4(1) | 3.8 | 12 | 17.6 | 0 |
| N-40-25 | 1(3) | 174.6 | 30 | 19.8 | 0 | 2(3) | 44.8 | 282 | 19.6 | 0 | 3(2) | 30.3 | 25.5 | 19 | 0 |
| N-40-30 | 2(3) | 386.1 | 174 | 20.8 | 0 | 2(1) | 69.4 | 363 | 20.7 | 0 | 2(3) | 50.7 | 66 | 20.8 | 0 |
| N-40-35 | 3(2) | 1163.6 | 241.5 | 22 | 0 | 0(2) | 241.4 | 2185.5 | 23 | 0 | 3(2) | 151.8 | 241.5 | 22 | 0 |
| N-40-40 | 1(1) | 1168.1 | 363 | 23.5 | 0 | 1(1) | 280.1 | 3279 | 23.3 | 0 | 1(1) | 246.6 | 363 | 23.5 | 0 |
| N-40-45 | 1(1) | 1223.8 | 363 | 25.5 | 0 | 0(0) | - | - | - | 3 | 1(1) | 945.6 | 1092 | 25.5 | 0 |
| N-50-10 | 4(1) | 0.8 | 3 | 8.2 | 1 | 4(1) | 0.7 | 3 | 8.4 | 1 | 5 | 0.4 | - | 8.4 | 1 |
| N-50-15 | 4(1) | 0.4 | 3 | 12 | 0.4 | 4(1) | 0.7 | 12 | 12 | 0.2 | 3(2) | 0.4 | 7.5 | 12.2 | 0.2 |
| N-50-20 | 5 | 1.8 | - | 15 | 0.2 | 5 | 0.3 | 0 | 15.6 | 0.2 | 3(2) | 0.3 | 12 | 15 | 0.2 |
| N-50-25 | 3(2) | 14.7 | 25.5 | 18.4 | 0 | 4(1) | 0.4 | 39 | 18.6 | 0 | 2(3) | 0.4 | 9 | 17.2 | 0 |
| N-50-30 | 4(1) | 15 | 39 | 20 | 0 | 4(1) | 4.6 | 39 | 19.7 | 0 | 4(1) | 4.1 | 12 | 19.6 | 0 |
| N-50-35 | 2(3) | 96.7 | 57 | 21.2 | 0 | 2(2) | 22.9 | 241.5 | 20.5 | 0 | 2(3) | 11.4 | 21 | 20.8 | 0 |
| N-50-40 | 1(3) | 139 | 174 | 22.5 | 0 | 3(1) | 30.4 | 363 | 22.5 | 0 | 3(2) | 22.1 | 39 | 21.4 | 0 |
| N-50-45 | 2(2) | 146.1 | 241.5 | 24.5 | 0 | 1(3) | 67.5 | 606 | 23.5 | 0 | 3(1) | 52.8 | 120 | 23.5 | 0 |
| W-40-10 | 4(1) | 0.5 | 3 | 9.8 | 1 | 4(1) | 0.3 | 12 | 10 | 1 | 4(1) | 0.4 | 3 | 9.8 | 1 |
| W-40-15 | 1(4) | 1.5 | 9.8 | 13.4 | 0.6 | 2(3) | 1 | 12 | 14 | 0.4 | 2(3) | 0.8 | 12 | 13.6 | 0.4 |
| W-40-20 | 3(2) | 60.2 | 79.5 | 17 | 0 | 4(1) | 13.8 | 39 | 17 | 0 | 4(1) | 8.1 | 12 | 16.8 | 0 |
| W-40-25 | 1(2) | 109.2 | 120 | 20.7 | 0 | 1(4) | 91.8 | 909.8 | 21 | 0 | 3(0) | 25.6 | - | 19.3 | 0 |
| W-40-30 | 3(1) | 127 | 120 | 21.3 | 0 | 1(1) | 25.2 | 120 | 22 | 0 | 3(1) | 36.7 | 120 | 21.3 | 0 |
| W-40-35 | 2(1) | 782.4 | 1092 | 23 | 0 | 1(1) | 177.7 | 1092 | 23 | 0 | 2(1) | 105.9 | 1092 | 23 | 0 |
| W-40-40 | 1(1) | 1010.1 | 3279 | 24 | 0 | 0(1) | 239.9 | 3279 | 25 | 0 | 1(1) | 138 | 1092 | 24 | 0 |
| W-40-45 | 1(0) | 1192.7 | - | 25 | 0 | 0(0) | - | - | - | 0 | 1(0) | 236.5 | - | 25 | 0 |
| W-50-10 | 5 | 0.7 | - | 8.4 | 1 | 4(1) | 0.6 | 3 | 8.4 | 1 | 4(1) | 0.3 | 3 | 8.4 | 1 |
| W-50-15 | 2(3) | 1.2 | 39 | 12 | 0.4 | 3(2) | 0.5 | 12 | 12.2 | 0.2 | 2(3) | 0.6 | 12 | 11.8 | 0.2 |
| W-50-20 | 4(1) | 1.1 | 12 | 14.8 | 0.2 | 4(1) | 1.7 | 12 | 15.6 | 0.2 | 3(2) | 1.1 | 7.5 | 15 | 0.2 |
| W-50-25 | 4(1) | 52.5 | 120 | 18.2 | 0 | 5 | 5.8 | 0 | 19.3 | 0 | 3(2) | 1.9 | 12 | 17.8 | 0 |
| W-50-30 | 4(1) | 149.6 | 363 | 20.4 | 0 | 5 | 15.1 | 0 | 19.8 | 0 | 4(1) | 8.4 | 39 | 19.8 | 0 |
| W-50-35 | 2(2) | 162.4 | 727.5 | 22.8 | 0 | 3(0) | 28.3 | 0 | 22 | 0 | 3(2) | 17.7 | 25.5 | 21.6 | 0 |
| W-50-40 | 1(3) | 239.6 | 849 | 23.8 | 0 | 1(3) | 178.7 | 2550 | 25 | 0 | 2(2) | 128.5 | 79.5 | 22.8 | 0 |
| W-50-45 | 2(2) | 262.1 | 1092 | 24.8 | 0 | 2(1) | 233.8 | 3279 | 23.5 | 0 | 2(2) | 143.5 | 120 | 24.5 | 0 |

Table 5: Special parts

| Name | Reduced skills | | | | Basic configurations | | | | All skills | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Opt | CPU | Nodes | Tasks | Idle | Opt | CPU | Nodes | Tasks | Idle | Opt | CPU | Nodes | Tasks | Idle |
| N-40-10 | 3(2) | 0.5 | 7.5 | 10 | 0.8 | 5 | 0.4 | 0 | 9.8 | 1 | 0(5) | 0.8 | 13.8 | 10 | 0.8 |
| N-40-15 | 5 | 0.8 | - | 14.8 | 0.2 | 2(3) | 1.1 | 93 | 14 | 0.2 | 0(5) | 27.7 | 42.6 | 15 | 0 |
| N-40-20 | 4(1) | 0.6 | 3 | 17 | 0 | 3(2) | 3.4 | 120 | 17 | 0 | 0(5) | 247.5 | 325.2 | 19 | 0 |
| N-40-25 | 4(1) | 0.6 | 3 | 20 | 0 | 2(3) | 44.8 | 282 | 19.6 | 0 | 0(4) | 2379.4 | 403.5 | 24.6 | 0 |
| N-40-30 | 4(1) | 1.3 | 3 | 21 | 0 | 2(1) | 69.4 | 363 | 20.7 | 0 | 0(3) | 2916.1 | 606 | 29.3 | 0 |
| N-40-35 | 4(1) | 4.4 | 12 | 23 | 0 | 0(2) | 241.4 | 2185.5 | 23 | 0 | 0(0) | - | - | - | - |
| N-40-40 | 4(1) | 12.9 | 39 | 23 | 0 | 1(1) | 280.1 | 3279 | 23.3 | 0 | 0(0) | - | - | - | - |
| N-40-45 | 3(2) | 50.2 | 79.5 | 23.4 | 0 | 0(0) | - | - | - | 3 | 0(0) | - | - | - | - |
| N-50-10 | 3(2) | 0.3 | 12 | 9.2 | 0.6 | 4(1) | 0.7 | 3 | 8.4 | 1 | 2(3) | 0.6 | 9 | 10 | 0.8 |
| N-50-15 | 5 | 0.4 | 0 | 13.2 | 0.2 | 4(1) | 0.7 | 12 | 12 | 0.2 | 0(5) | 3.9 | 10.2 | 15 | 0 |
| N-50-20 | 5 | 0.6 | 0 | 15.4 | 0 | 5 | 0.3 | 0 | 15.6 | 0.2 | 1(4) | 35.8 | 43.5 | 19.2 | 0 |
| N-50-25 | 5 | 0.5 | 0 | 18.2 | 0 | 4(1) | 0.4 | 39 | 18.6 | 0 | 1(4) | 132.8 | 133.5 | 25 | 0 |
| N-50-30 | 4(1) | 1 | 3 | 19 | 0 | 4(1) | 4.6 | 39 | 19.7 | 0 | 0(1) | 322.1 | 363 | 26 | 0 |
| N-50-35 | 4(1) | 4.2 | 12 | 21 | 0 | 2(2) | 22.9 | 241.5 | 20.5 | 0 | 0(1) | 941.3 | 1092 | 27 | 0 |
| N-50-40 | 2(3) | 2 | 21 | 23 | 0 | 3(1) | 30.4 | 363 | 22.5 | 0 | 0(0) | - | - | - | - |
| N-50-45 | 2(3) | 2.8 | 30 | 23 | 0 | 1(3) | 67.5 | 606 | 23.5 | 0 | 0(0) | - | - | - | - |
| W-40-10 | 5 | 0.5 | 0 | 10 | 0.8 | 4(1) | 0.3 | 12 | 10 | 1 | 1(4) | 1.5 | 3 | 10 | 0.8 |
| W-40-15 | 3(2) | 0.5 | 12 | 14.4 | 0 | 2(3) | 1 | 12 | 14 | 0.4 | 0(5) | 13.1 | 17.4 | 15 | 0 |
| W-40-20 | 5 | 0.5 | 0 | 18.2 | 0 | 4(1) | 13.8 | 39 | 17 | 0 | 0(3) | 2170.3 | 525 | 20 | 0 |
| W-40-25 | 2(3) | 0.6 | 18 | 21 | 0 | 1(4) | 91.8 | 909.8 | 21 | 0 | 0(1) | 6485.7 | 1092 | 25 | 0 |
| W-40-30 | 4(1) | 2.1 | 39 | 22.2 | 0 | 1(1) | 25.2 | 120 | 22 | 0 | 0(1) | 5700.3 | 1092 | 27 | 0 |
| W-40-35 | 4(1) | 5.1 | 39 | 23.4 | 0 | 1(1) | 177.7 | 1092 | 23 | 0 | 0(0) | - | - | - | - |
| W-40-40 | 5 | 26.7 | 0 | 23.6 | 0 | 0(1) | 239.9 | 3279 | 25 | 0 | 0(0) | - | - | - | - |
| W-40-45 | 0(5) | 70.5 | 109.2 | 23.8 | 0 | 0(0) | - | - | - | 0 | 0(0) | - | - | - | - |
| W-50-10 | 4(1) | 0.3 | 3 | 9.5 | 0.8 | 4(1) | 0.6 | 3 | 8.4 | 1 | 2(3) | 0.5 | 6 | 10 | 0.8 |
| W-50-15 | 4(1) | 0.4 | 3 | 14 | 0 | 3(2) | 0.5 | 12 | 12.2 | 0.2 | 0(5) | 23.8 | 21 | 15 | 0 |
| W-50-20 | 5 | 0.5 | 0 | 15.2 | 0 | 4(1) | 1.7 | 12 | 15.6 | 0.2 | 0(5) | 120.8 | 33.6 | 20 | 0 |
| W-50-25 | 5 | 0.6 | 0 | 17.8 | 0 | 5 | 5.8 | 0 | 19.3 | 0 | 0(3) | 298.2 | 66 | 24.6 | 0 |
| W-50-30 | 5 | 1.4 | 0 | 19.2 | 0 | 5 | 15.1 | 0 | 19.8 | 0 | 0(0) | - | - | - | - |
| W-50-35 | 4(1) | 3.5 | 3 | 20.6 | 0 | 3(0) | 28.3 | 0 | 22 | 0 | 0(0) | - | - | - | - |
| W-50-40 | 4(1) | 4.3 | 12 | 21 | 0 | 1(3) | 178.7 | 2550 | 25 | 0 | 0(0) | - | - | - | - |
| W-50-45 | 5 | 17.6 | 0 | 23.2 | 0 | 2(1) | 233.8 | 3279 | 23.5 | 0 | 0(0) | - | - | - | - |

Table 6: Skills

| Name | Basic configurations (15-45 min) | | | | | Reduced service times (15-30 min) | | | | | Reduced service times (10-20 min) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Opt | CPU | Nodes | Tasks | Idle | Opt | CPU | Nodes | Tasks | Idle | Opt | CPU | Nodes | Tasks | Idle |
| N-40-10 | 5 | 0.4 | 0 | 9.8 | 1 | 4(1) | 0.4 | 3 | 9.8 | 1 | 4(1) | 0.5 | 3 | 9.8 | 1 |
| N-40-15 | 2(3) | 1.1 | 93 | 14 | 0.2 | 1(4) | 2.1 | 9.8 | 13.8 | 0.2 | 1(4) | 2.8 | 9.8 | 13.8 | 0.2 |
| N-40-20 | 3(2) | 3.4 | 120 | 17 | 0 | 3(2) | 17.7 | 12 | 17 | 0 | 3(2) | 22.4 | 25.5 | 17 | 0 |
| N-40-25 | 2(3) | 44.8 | 282 | 19.6 | 0 | 2(3) | 298.5 | 174 | 20.2 | 0 | 2(2) | 354.8 | 241.5 | 20.3 | 0 |
| N-40-30 | 2(1) | 69.4 | 363 | 20.7 | 0 | 3(1) | 343.8 | 363 | 21.8 | 0 | 2(2) | 393.8 | 363 | 21.5 | 0 |
| N-40-35 | 0(2) | 241.4 | 2185.5 | 23 | 0 | 1(2) | 2356.6 | 3279 | 22.3 | 0 | 0(2) | 2430.5 | 2185.5 | 23 | 0 |
| N-40-40 | 1(1) | 280.1 | 3279 | 23.3 | 0 | 2(0) | 3309.2 | 0 | 25.5 | 0 | 0(1) | 3707.3 | 3279 | 24 | 0 |
| N-40-45 | 0(0) | - | - | - | 3 | 0(1) | 2751.2 | 3279 | 25 | 0 | 0(1) | 4285.9 | 3279 | 24 | 0 |
| N-50-10 | 4(1) | 0.7 | 3 | 8.4 | 1 | 5 | 0.5 | 0 | 8.4 | 1 | 5 | 0.6 | 0 | 8.4 | 1 |
| N-50-15 | 4(1) | 0.7 | 12 | 12 | 0.2 | 3(2) | 0.9 | 3 | 11.8 | 0.4 | 3(2) | 0.4 | 7.5 | 11.8 | 0.4 |
| N-50-20 | 5 | 0.3 | 0 | 15.6 | 0.2 | 4(1) | 1.6 | 12 | 14.8 | 0.2 | 4(1) | 1.1 | 12 | 14.8 | 0.2 |
| N-50-25 | 4(1) | 0.4 | 39 | 18.6 | 0 | 3(2) | 5.2 | 7.5 | 17.8 | 0 | 3(2) | 22.8 | 25.5 | 17.8 | 0 |
| N-50-30 | 4(1) | 4.6 | 39 | 19.7 | 0 | 4(1) | 5.6 | 12 | 19.8 | 0 | 3(2) | 34.4 | 39 | 20.2 | 0 |
| N-50-35 | 2(2) | 22.9 | 241.5 | 20.5 | 0 | 3(2) | 162.5 | 79.5 | 20.8 | 0 | 3(2) | 250.5 | 241.5 | 23.2 | 0 |
| N-50-40 | 3(1) | 30.4 | 363 | 22.5 | 0 | 3(2) | 430.6 | 120 | 23 | 0 | 0(4) | 487 | 363 | 27.5 | 0 |
| N-50-45 | 1(3) | 67.5 | 606 | 23.5 | 0 | 2(2) | 453 | 241.5 | 23.8 | 0 | 0(4) | 516.2 | 1213.5 | 25 | 0 |
| W-40-10 | 4(1) | 0.3 | 12 | 10 | 1 | 5 | 0.4 | 9.8 | 9.8 | 1 | 5 | 0.5 | 0 | 9.8 | 1 |
| W-40-15 | 2(3) | 1 | 12 | 14 | 0.4 | 1(4) | 1.8 | 12 | 13.8 | 0.4 | 1(4) | 1.7 | 9.8 | 13.8 | 0.4 |
| W-40-20 | 4(1) | 13.8 | 39 | 17 | 0 | 3(2) | 35.2 | 39 | 17.4 | 0 | 3(2) | 39.6 | 39 | 17.4 | 0 |
| W-40-25 | 1(4) | 91.8 | 909.8 | 21 | 0 | 2(1) | 138.6 | 120 | 21 | 0 | 3(1) | 153.7 | 120 | 21 | 0 |
| W-40-30 | 1(1) | 25.2 | 120 | 22 | 0 | 2(2) | 381.8 | 241.5 | 22 | 0 | 3(2) | 845.5 | 363 | 21.8 | 0 |
| W-40-35 | 1(1) | 177.7 | 1092 | 23 | 0 | 1(1) | 501.5 | 363 | 24.5 | 0 | 0(2) | 902.3 | 727.5 | 23 | 0 |
| W-40-40 | 0(1) | 239.9 | 3279 | 25 | 0 | 0(1) | 482.6 | 363 | 25 | 0 | 1(0) | 925.9 | - | 25 | 0 |
| W-40-45 | 0(0) | - | - | - | 0 | 0(0) | - | - | | 3 | 0(0) | - | - | | 3 |
| W-50-10 | 4(1) | 0.6 | 3 | 8.4 | 1 | 4(1) | 0.4 | 3 | 8.4 | 1 | 4(1) | 0.5 | 3 | 8.4 | 1 |
| W-50-15 | 3(2) | 0.5 | 12 | 12.2 | 0.2 | 3(2) | 0.7 | 7.5 | 12.2 | 0.4 | 3(2) | 0.6 | 12 | 12.2 | 0.4 |
| W-50-20 | 4(1) | 1.7 | 12 | 15.6 | 0.2 | 5 | 2 | 0 | 14.6 | 0.2 | 5 | 1.7 | 0 | 14.6 | 0.2 |
| W-50-25 | 5 | 5.8 | 0 | 19.3 | 0 | 5 | 15.6 | 0 | 18 | 0 | 5 | 17.8 | 0 | 18 | 0 |
| W-50-30 | 5 | 15.1 | 0 | 19.8 | 0 | 5 | 17.6 | 0 | 20.2 | 0 | 4(1) | 22.9 | 39 | 20.2 | 0 |
| W-50-35 | 3(0) | 28.3 | 0 | 22 | 0 | 3(1) | 28.6 | 39 | 22.8 | 0 | 3(1) | 36 | 39 | 23 | 0 |
| W-50-40 | 1(3) | 178.7 | 2550 | 25 | 0 | 0(4) | 188.6 | 99.8 | 23.8 | 0 | 1(3) | 415.7 | 282 | 24 | 0 |
| W-50-45 | 2(1) | 233.8 | 3279 | 23.5 | 0 | 3(1) | 246.5 | 120 | 25 | 0 | 0(4) | 477.3 | 302.3 | 25.8 | 0 |

Table 7: Service times

| Name | Basic configurations (3 technicians) | | | | | More technicians (4 technicians) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Opt | CPU | Nodes | Tasks | Idle | Opt | CPU | Nodes | Tasks | Idle |
| N-40-10 | 5 | 0.4 | 0 | 9.8 | 1 | 3(2) | 0.7 | 3 | 10 | 1.8 |
| N-40-15 | 2(3) | 1.1 | 93 | 14 | 0.2 | 0(5) | 2.3 | 10.2 | 15 | 0.4 |
| N-40-20 | 3(2) | 3.4 | 120 | 17 | 0 | 1(4) | 53.2 | 16.5 | 20 | 0 |
| N-40-25 | 2(3) | 44.8 | 282 | 19.6 | 0 | 2(2) | 36.5 | 39 | 24.3 | 0 |
| N-40-30 | 2(1) | 69.4 | 363 | 20.7 | 0 | 1(3) | 257.5 | 93 | 27.3 | 0 |
| N-40-35 | 0(2) | 241.4 | 2185.5 | 23 | 0 | 2(1) | 1213.1 | 1092 | 31.3 | 0 |
| N-40-40 | 1(1) | 280.1 | 3279 | 23.3 | 0 | 0(1) | 1158.4 | 1092 | 35 | 0 |
| N-40-45 | 0(0) | - | - | - | - | 0(0) | - | - | - | - |
| N-50-10 | 4(1) | 0.7 | 3 | 8.4 | 1 | 2(3) | 0.7 | 18 | 10 | 1.6 |
| N-50-15 | 4(1) | 0.7 | 12 | 12 | 0.2 | 4(1) | 2.4 | 12 | 14.2 | 0.4 |
| N-50-20 | 5 | 0.3 | 0 | 15.6 | 0.2 | 2(3) | 7 | 21 | 18 | 0 |
| N-50-25 | 4(1) | 0.4 | 39 | 18.6 | 0 | 3(2) | 44.8 | 39 | 22.8 | 0 |
| N-50-30 | 4(1) | 4.6 | 39 | 19.7 | 0 | 1(4) | 62.6 | 79.5 | 25.2 | 0 |
| N-50-35 | 2(2) | 22.9 | 241.5 | 20.5 | 0 | 1(3) | 989.1 | 606 | 27 | 0 |
| N-50-40 | 3(1) | 30.4 | 363 | 22.5 | 0 | 1(2) | 802.1 | 727.5 | 27.3 | 0 |
| N-50-45 | 1(3) | 67.5 | 606 | 23.5 | 0 | 0(1) | 891.5 | 1092 | 29 | 0 |
| W-40-10 | 4(1) | 0.3 | 12 | 10 | 1 | 4(1) | 0.4 | 3 | 10 | 2 |
| W-40-15 | 2(3) | 1 | 12 | 14 | 0.4 | 1(4) | 3.8 | 16.5 | 15 | 0.6 |
| W-40-20 | 4(1) | 13.8 | 39 | 17 | 0 | 0(4) | 15.3 | 39 | 20 | 0 |
| W-40-25 | 1(4) | 91.8 | 909.8 | 21 | 0 | 1(1) | 98.8 | 120 | 24 | 0 |
| W-40-30 | 1(1) | 25.2 | 120 | 22 | 0 | 2(1) | 3722.5 | 1092 | 27.3 | 0 |
| W-40-35 | 1(1) | 177.7 | 1092 | 23 | 0 | 1(1) | 5202.1 | 3279 | 29 | 0 |
| W-40-40 | 0(1) | 239.9 | 3279 | 25 | 0 | 0(0) | - | - | - | - |
| W-40-45 | 0(0) | - | - | - | - | 0(0) | - | - | - | - |
| W-50-10 | 4(1) | 0.6 | 3 | 8.4 | 1 | 0(5) | 0.6 | 4.8 | 9.6 | 1.2 |
| W-50-15 | 3(2) | 0.5 | 12 | 12.2 | 0.2 | 3(2) | 1.8 | 12 | 14.4 | 0.6 |
| W-50-20 | 4(1) | 1.7 | 12 | 15.6 | 0.2 | 1(3) | 27.8 | 30 | 18.3 | 0 |
| W-50-25 | 5 | 5.8 | 0 | 19.3 | 0 | 2(2) | 317.1 | 241.5 | 21.8 | 0 |
| W-50-30 | 5 | 15.1 | 0 | 19.8 | 0 | 1(2) | 508.5 | 363 | 24.3 | 0 |
| W-50-35 | 3(0) | 28.3 | 0 | 22 | 0 | 1(1) | 4663.9 | 3279 | 25.5 | 0 |
| W-50-40 | 1(3) | 178.7 | 2550 | 25 | 0 | 0(1) | 1353.3 | 3279 | 27 | 0 |
| W-50-45 | 2(1) | 233.8 | 3279 | 23.5 | 0 | 0(0) | - | - | | - |

Table 8: Number of technicians

### 5.3 Comparison with CPLEX

Branch-and-price has allowed us to solve larger instances at the optimum than CPLEX, when applied to the arc-based MIP formulation in [8], as shown in Table 9. Since the MIP model penalizes skipped breaks, but does not force technicians to take them, an alternative version of our branch-and-price algorithm was developed where breaks were not enforced when solving the pricing problem (which led to a more difficult pricing problem). The results show a huge difference between the branch-and-price algorithm and CPLEX with regard to the number of instances solved to optimality and computation times. In particular, CPLEX only solves instances with up to 15 tasks. We recall that our branch-and-price algorithm was able to solve instances with up to 45 tasks.

| **Name** | **CPLEX** | | **Branch-and-Price** | |
|---|---|---|---|---|
| | Opt | CPU | Opt | CPU |
| N-40-10 | 5 | 1073 | 5 | 1.2 |
| N-40-15 | 0 | - | 5 | 24.6 |
| N-50-10 | 5 | 732 | 5 | 0.6 |
| N-50-15 | 0 | - | 5 | 0.7 |
| W-40-10 | 5 | 1593 | 5 | 2.4 |
| W-40-15 | 0 | - | 5 | 23.2 |
| W-50-10 | 5 | 655 | 5 | 0.6 |
| W-50-15 | 2 | 24752 | 5 | 2.0 |

Table 9: CPLEX vs Branch-and-Price

## 6 Conclusion

In this paper, a technician routing and scheduling problem was tackled with a branch-and-price algorithm, using two different branching strategies. Based on the test instances in [8], we first evaluated the benefits provided by DSSR over the standard pricing problem, using both branching strategies. The DSSR implementation with ternary branching proved to be the best. Then, we considered the impact of different problem characteristics on the performance of our algorithm. Finally, a comparison with an arc-based MIP formulation solved with CPLEX showed that branch-and-price is largely superior and can solve larger instances. Given that real-world instances are typically larger than those reported in this paper, our research will now focus on developing metaheuristics and matheuristics for the problem. Furthermore, we want to consider a variant where new tasks occur dynamically and must be integrated in real-time into the current solution.

# References

[1] F. Blakeley, B. Arguello, B. Cao, W. Hall, and J. Knolmajer. Optimizing periodic maintenance operations for schindler elevator corporation. *Interfaces*, 33:67–79, 2003.

[2] N. Bostel, P. Dejax, P. Guez, and F. Tricoire. Multiperiod planning and routing on a rolling horizon for field force optimization logistics. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 503–525. Springer, 2008.

[3] J.-F. Cordeau, G. Laporte, F. Pasin, and S. Ropke. Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13:393–409, 2010.

[4] R.B. Damm, M.G.C. Resende, and D. P. Ronconi. A biased random key genetic algorithm for the field technician scheduling problem. *Computers & Operations Research*, 75:49 – 63, 2016.

[5] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44:216–229, 2004.

[6] S. Gélinas, M. Desrochers, J. Desrosiers, and M.M. Solomon. A new branching strategy for time constrained routing problems with application to backhauling. *Annals of Operations Research*, 61:91–109, 1995.

[7] H. Hashimoto, S. Boussier, M. Vasquez, and C. Wilbaut. A GRASP-based approach for technicians and interventions scheduling for telecommunications. *Annals of Operations Research*, 183:143–161, 2011.

[8] I. Mathlouthi, M. Gendreau, and J.-Y. Potvin. Mixed integer programming for a multi-attribute technician routing and scheduling problem. Technical Report CIRRELT-2016-23, CIRRELT, 2016, forthcoming in Information Systems and Operational Research.

[9] I. Or. Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking. Technical report, PhD dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1976.

[10] V. Pillac, C. Guéret, and A. Medaglia. On the dynamic technician routing and scheduling problem. In *Proceedings of the 5th International Workshop on Freight Transportation and Logistics (ODYSSEUS 2012)*, Mikonos, Greece, May 2012.

[11] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34:2403–2435, 2007.

[12] J.-Y. Potvin and J.-M. Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66:331–340, 1993.

[13] G. Righini and M. Salani. Dynamic programming algorithms for the elementary shortest path problem with resource constraints. *Electronic Notes in Discrete Mathematics*, 17:247 – 249, 2004.

[14] H. Tang, E. Miller-Hooks, and R. Tomastik. Scheduling technicians for planned maintenance of geographically distributed equipment. *Transportation Research Part E*, 43:591 – 609, 2007.

[15] E. Tsang and C. Voudouris. Fast local search and guided local search and their application to British Telecom's workforce scheduling problem. *Operations Research Letters*, 20:119 – 127, 1997.

[16] D. Weigel and B. Cao. Applying GIS and OR techniques to solve Sears technician-dispatching and home delivery problems. *Interfaces*, 29:112–130, January 1999.

[17] J. Xu and S.Y. Chiu. Effective heuristic procedures for a field technician scheduling problem. *Journal of Heuristics*, 7:495–509, 2001.

[18] E. Zamorano and R. Stolletz. Branch-and-price approaches for the multiperiod technician routing and scheduling problem. *European Journal of Operational Research*, 257:55 – 68, 2017.