CIRRELT

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

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Mixed Integer Programming for a Multi-Attribute Technician Routing and Scheduling Problem

Ines Mathlouthi
Michel Gendreau
Jean-Yves Potvin

April 2016

CIRRELT-2016-23

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

Mixed Integer Programming 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 consider a multi-attribute technician routing and scheduling problem motivated from an application for the repair of electronic transaction equipment. This problem consists in routing technicians to serve requests for service, while taking into account many attributes like technician skills, task priorities, multiple time windows, parts inventory, breaks and overtime. A mixed integer programming model is proposed to address this problem, which is then solved with a commercial solver. The computational results explore the difficulty of the problem along various dimensions and underline its inherent complexity.

Keywords: Technician routing and scheduling problem, multiple time windows, inventory, mixed integer programming.

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, 2016

1 Introduction

In the technician routing and scheduling problem (TRSP), a number of technicians must serve different tasks while satisfying resource constraints. The TRSP belongs to the class of Vehicle Routing Problems (VRP) [4] and is related in particular to the VRP with time windows [6]. There are, however, significant differences such as skill requirements to perform different types of tasks and relatively large service times when compared to travel times. In our case, we also deal with inventory issues, in particular the availability of spare parts and special parts in the technician's vehicle.

Our TRSP, which is motivated from an application for the repair of electronic transactions equipment, can be defined as follows. There is a set of technicians, each with different skills for different types of tasks. These tasks might also be assigned different priority levels. There is a set of customers, where every customer requires the service of a technician to perform a particular task. The goal is then to assign the tasks to the technicians and to build a route for each technician, starting and ending at his home base location, so as to optimize a given objective. The solution must also satisfy various constraints related to the required skills of the technicians to perform the assigned tasks, working hours, multiple time windows, availability of spare parts and special parts. It should be noted that technician skills are sometimes accounted for in the literature as degrees of ability (instead of constraints) and are integrated into the objective.

A particular feature of our problem comes from an inventory of spare parts. More precisely, a technician leaves his home base position to start his route with an initial inventory. However, if there is not enough parts to serve all tasks, he has the opportunity to replenish once along the route by going through a particular depot, which has previously been assigned to him. There are also special parts that are not carried in the vehicle unless one or more tasks along the route require them. In such a case, the technician must also take them from his depot before performing these tasks.

Overall, the main contributions of this work come from the development of a mixed integer programming (MIP) model for a complex problem motivated from a real-world application. Furthermore, an analysis of the problem difficulty along various dimensions is provided by generating instances with different characteristics observed in practice.

The remainder of this paper is organized as follows. In section 2, we review some related work. In section 3, the problem is introduced. Then, the mathematical formulation is presented in section 4. The generation of the test instances is explained in section 5, followed by the results obtained in section 6. Finally, section 7 concludes the paper.

2 Related Work

TRSPs have received limited attention compared to VRPs, despite their numerous practical applications. The first work is reported in 1997 by Tsang and Voudouris [12] where the authors introduce the technician workforce scheduling problem faced by British Telecom. The particularity of this problem is that there are no skill constraints. They are replaced by a proficiency factor that reduces the service time depending on the technician experience. A Guided Local Search (GLS) and a so-called Fast Local Search are used to solve this problem. Later, Weigel and Coo [13] introduce the problem faced by a well-known retailer when providing on-site technical assistance. The proposed solution consists of assigning requests to technicians and then optimizing each route individually through Or-opt exchanges [7].

In [14], Xu and Chiu propose a MIP for a TRSP where the objective is to maximize the number of served requests, while taking into account request priorities, skills and overtime. Four heuristics based on local search and GRASP are reported. In [1], Blakeley et al. solve a periodic maintenance problem faced by the Schindler Elevator Corporation for their elevators and escalators. In this application, the technician routes must account for technician skills, travel times and working regulations. A little bit later, a similar application was addressed by Tang et al. [11] with a tabu search heuristic.

In 2007, the French Operations Research Society (ROADEF) initiated a challenge based on a problem encountered by France Telecom. The participants had to schedule technician tours on a multiple-day horizon. The particularity of this problem is that each task needs one or more skills with different proficiency levels, while technicians can have multiple skills. To solve this problem, teams of technicians working together must be created. However, the routing aspect of the problem is ignored. Based on this challenge, Hashimioto et al. [5], developed a GRASP for this problem while Cordeau et al. [3] proposed a mathematical model and a problem-solving methodology based on an adaptive large neighborhood search (ALNS).

A dynamic variant of the TRSP is addressed in Bostel et al. [2]. The authors introduce a problem 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 on static tasks to produce tours for every day of the week. Dynamic tasks are then integrated into the solution as they occur, still 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.

Finally, Pillac et al. [8] also address a TRSP in which a fraction of the tasks occur dynamically. A parallel architecture is proposed to speed up the calculations.

An initial solution is first created with known tasks using a regret heuristic [10]. This solution is then improved with ALNS [9]. The latter works by successively destroying (removing tasks) and repairing (reinserting tasks) to produce a new solution from the current one. When a new task is received, the part of the current solution already executed is fixed and the new task is incorporated into the solution by running the ALNS for a limited number of iterations.

To the best of our knowledge, no work considers concurrently technician skills, task priorities, multiple time windows, breaks, overtime and parts inventory. This complex problem will now be introduced more precisely in the next section.

3 Problem Statement

Our problem is a technician routing and scheduling problem encountered by a company providing repair services for electronic transactions equipment. Basically, customers call to report equipment failures that must be fixed. The task to be performed at a customer site is then characterized by the following attributes:

- Gain (based on the customer's service priority);
- Subset of technicians with required skills to perform the task;
- Types and number of required spare parts;
- Special part, if any;
- Service time;
- Multiple time windows.

There are one or more depots, where each depot contains a (virtually infinite) number of parts. Each technician is assigned to a particular depot for the replenishment of his spare parts or for the acquisition of special parts. Each technician can work from 9H00 AM to 5H00 PM (if no overtime) and is allowed three breaks during the day: one break of 15 minutes in the morning and afternoon, respectively, and a mid-day break of 30 minutes. The morning, mid-day and afternoon periods are defined through time window constraints. Finally, a technician cannot travel more than a given maximum distance during his workday.

With this information, the problem is to design technician routes for one day, where each route starts and ends at the technician's home base and serves a number of tasks, including one possible stop at the preassigned depot, while satisfying the required skills for each task, the multiple time windows at each location, the time window for each break, the required number of spare parts for each task and the

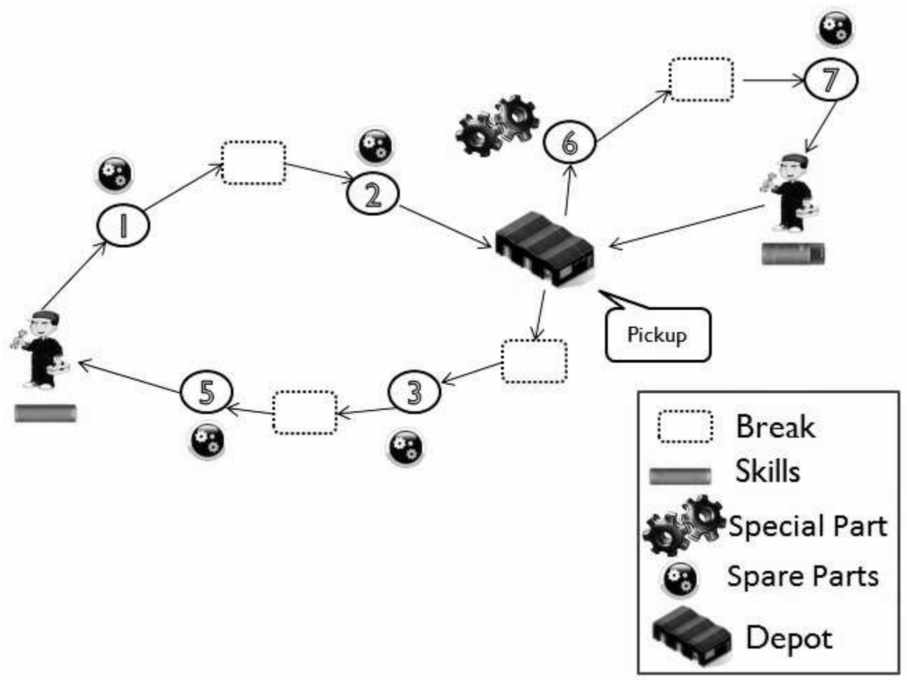

Figure 1: Example with two technician routes

requirement (or not) of a special part for each task. The goal is to optimize an objective involving overtime, total traveled distance and total gain over the performed tasks.

Figure 1 shows an example with one depot, a single customer requiring a special part, and two technician routes. The first technician has all the required skills to perform all tasks. His route serves tasks 1, 2, 3, 5 in this order. After task 2, he goes to the depot to replenish his inventory of spare parts, takes a break and then goes to task 3. The second technician does not have all the required skills to perform all tasks. His route serves tasks 6 and 7 and he must first go to the depot to pick up the special part needed by task 6. He then takes a break before going to task 7.

4 Model

In this section a mixed integer programming model for our problem is provided. The parameters are first defined, followed by the decision variables. It should be noted that we are not aware of any model that accounts for all these characteristics.

4.1 Parameters

- $D = \{1, \ldots, g\}$: Set of depots;

- $K = \{1, \ldots, m\}$: Set of technicians;
- $K^d \subseteq K$: Set of technicians assigned to depot $d \in D$;
- $I = \{1, \ldots, n\}$: Set of tasks;
- $\Theta = \{\theta^1, \ldots, \theta^m\}$: Set of starting positions of the technicians (home bases);
- $I' = I \cup \Theta$
- $I'' = I' \cup D$
- $P = \{1, \ldots, o\}$: Set of types of spare parts;
- $I_t \subseteq I$: Set of tasks that require a special part;
- $P_i \subseteq P$: Set of types of spare parts required to perform task $i \in I$;
- F_i: Set of time windows of task $i \in I$;
- α_i: Gain of task $i \in I$;
- a_i^r: Lower bound of time window $r \in F_i$ of task $i \in I$;
- b_i^r: Upper bound of time window $r \in F_i$ of task $i \in I$;
- a_l: Lower bound of time window of break $l = 1, 2, 3$;
- b_l: Upper bound of time window of break $l = 1, 2, 3$;
- a^k: Earliest workday start time of technician $k \in K$;
- b^k: Latest (regular) workday end time of technician $k \in K$;
- σ_i: Duration of service of task $i \in I$;
- σ_l: Duration of break $l = 1, 2, 3$;
- d_{ij}: Distance between i et $j \in I''$, $i \neq j$;
- t_{ij}: Travel time between i et $j \in I''$, $i \neq j$;
- $d_{\max}$: Maximum traveled distance of each technician;
- d^k: Depot of technician $k \in K$;
- z_{pi}: Number of spare parts of type $p \in P$ needed to perform task $i \in I$;
- v_p^k: Number of spare parts of type $p \in P$ in the vehicle of technician $k \in K$ after replenishment;
- $v_{p\theta^k}^k$: Number of spare parts of type $p \in P$ in the vehicle of technician $k \in K$ at his starting position (home base);

- $s_i^k = \begin{cases} 1, & \text{if technician } k \in K \text{ has the required skills to perform task } i \in I; \\ 0, & \text{otherwise.} \end{cases}$
- M: Arbitrarily large constant;
- β: Replenishment time at the depot for each technician;
- $\psi, \vartheta, \varphi, \upsilon \in [0, 1]$.

4.2 Variables

- δ^k : Overtime of technician $k \in K$;
- U_{pi}^k : Number of spare parts of type $p \in P$ in the vehicle of technician $k \in K$ after performing task $i \in I$;
- τ_i: Start time of task $i \in I$;
- τ_l^k: Start time of break $l = 1, 2, 3$ of technician $k \in K$;
- $x_{ij}^k = \begin{cases} 1, & \text{if technician } k \text{ has not visited his depot before moving directly from } i \text{ to } j; \\ 0, & \text{otherwise.} \end{cases}$
- $\bar{x}_{ij}^k = \begin{cases} 1, & \text{if technician } k \text{ has visited his depot before moving directly from } i \text{ to } j; \\ 0, & \text{otherwise.} \end{cases}$
- $\tilde{x}_{ij}^k = \begin{cases} 1, & \text{if technician } k \text{ moves from } i \text{ to } j \text{ through his depot;} \\ 0, & \text{otherwise.} \end{cases}$
- $y_i^k = \begin{cases} 1, & \text{if task } i \text{ is assigned to technician } k; \\ 0, & \text{otherwise.} \end{cases}$
- $w_{il}^k = \begin{cases} 1, & \text{if technician } k \text{ takes his break } l = 1, 2, 3 \text{ after task } i; \\ 0, & \text{otherwise.} \end{cases}$
- $\bar{w}_l^k = \begin{cases} 1, & \text{if technician } k \text{ is active and does not take his break } l = 1, 2, 3; \\ 0, & \text{otherwise.} \end{cases}$
- $f_i^r = \begin{cases} 1, & \text{if task } i \text{ is performed in time window } r,\ r \in F_i; \\ 0, & \text{otherwise.} \end{cases}$
- $\bar{u}^k = \begin{cases} 1, & \text{if technician } k \text{ is inactive (not used)} \\ 0, & \text{otherwise.} \end{cases}$

4.3 Model

$$\max \varphi \sum_{k\in K}\sum_{i\in I}\alpha_i y_i^k - \vartheta \sum_{k\in K}\sum_{i,j\in I\prime}(x_{ij}^k+\bar{x}_{ij}^k)d_{ij}+\tilde{x}_{ij}^k(d_{id^k}+d_{d^kj}) - \psi\sum_{k\in K}\delta^k - \upsilon\sum_{k\in K}\sum_{l=1,2,3}\bar{w}_l^k \quad (1)$$

Subject to

Task/Technician constraints

$$\sum_{k\in K} y_i^k \leq 1 \quad i\in I \quad (2)$$

$$y_i^k \leq s_i^k \qquad i\in I,\ k\in K \quad (3)$$

Coherence between variables of type x and u

$$\bar{u}^k = 1 - \sum_{i\in I}(x_{\theta^k i}^k + \tilde{x}_{\theta^k i}^k) \qquad k\in K \quad (4)$$

Coherence between variables of type x and y

$$\sum_{j\in I}(x_{ji}^k + \tilde{x}_{ji}^k + \bar{x}_{ji}^k) + x_{\theta^k i}^k + \tilde{x}_{\theta^k i}^k = y_i^k \qquad i\in I,\ k\in K \quad (5)$$

$$\bar{x}_{i\theta^k}^k + \sum_{j\in I}\bar{x}_{ij}^k = \tilde{x}_{\theta^k i}^k + \sum_{j\in I}(\bar{x}_{ji}^k + \tilde{x}_{ji}^k) \qquad i\in I,\ k\in K \quad (6)$$

Flow constraints

$$\sum_{i\in I}(x_{\theta^k i}^k + \tilde{x}_{\theta^k i}^k) \leq 1 \qquad k\in K \quad (7)$$

$$\sum_{i\in I}(x_{\theta_k i}^k + \tilde{x}_{\theta_k i}^k) - \sum_{i\in I}(x_{i\theta^k}^k + \bar{x}_{i\theta^k}^k) = 0 \qquad k\in K \quad (8)$$

$$[x_{\theta^k i}^k + \tilde{x}_{\theta^k i}^k + \sum_{j\in I}(x_{ji}^k + \tilde{x}_{ji}^k + \bar{x}_{ji}^k)] - [x_{i\theta^k}^k + \bar{x}_{i\theta^k}^k + \sum_{j\in I}(x_{ij}^k + \tilde{x}_{ij}^k + \bar{x}_{ij}^k)] = 0 \quad (9)$$

$$i\in I,\ k\in K$$

Special parts

$$y_i^k \leq \tilde{x}_{\theta^k i}^k + \sum_{j \in I} (\tilde{x}_{ji}^k + \bar{x}_{ji}^k) \qquad \begin{array}{l} i \in I_t \\ k \in K \end{array} \tag{10}$$

Inventory constraints

$$U_{pj}^k + (1 - (x_{ji}^k + \tilde{x}_{ji}^k + \bar{x}_{ji}^k))M + v_p^k \tilde{x}_{ji}^k \geq z_{pi} \qquad \begin{array}{l} (i \neq j) \in I \\ p \in P_i \\ k \in K \end{array} \tag{11}$$

$$v_{p\theta^k}^k + (1 - (x_{\theta^k i}^k + \tilde{x}_{\theta^k i}^k))M + v_p^k \tilde{x}_{\theta^k i}^k \geq z_{pi} \qquad \begin{array}{l} i \in I \\ p \in P_i \\ k \in K \end{array} \tag{12}$$

$$U_{pi}^k \leq U_{pj}^k - z_{pi} + (1 - (x_{ji}^k + \bar{x}_{ji}^k + \tilde{x}_{ji}^k))M + v_p^k \tilde{x}_{ji}^k \qquad \begin{array}{l} (i \neq j) \in I \\ p \in P_i \\ k \in K \end{array} \tag{13}$$

$$U_{pi}^k \leq U_p^k - z_{pi} + (1 - (x_{\theta^k i}^k + \tilde{x}_{\theta^k i}^k))M + v_p^k \tilde{x}_{\theta^k i}^k \qquad \begin{array}{l} i \in I \\ p \in P_i \\ k \in K \end{array} \tag{14}$$

Maximal distance constraint

$$\sum_{i,j \in I} ((x_{ij}^k + \bar{x}_{ij}^k) d_{ij} + \tilde{x}_{ij}^k (d_{id^k} + d_{d^k j})) + \sum_{i \in I} d_{\theta^k i} x_{\theta^k i}^k + \sum_{i \in I} (d_{\theta^k d^k} + d_{d^k i}) \tilde{x}_{\theta^k i}^k \tag{15}$$

$$+ \sum_{i \in I} d_{i\theta^k} (x_{i\theta^k}^k + \bar{x}_{i\theta^k}^k) \leq d_{\max}$$

$$k \in K$$

Time constraints

$$\delta^k \geq (\tau_i + \sigma_i + t_{i\theta^k} - b^k) - (1 - (x_{i\theta^k}^k + \bar{x}_{i\theta^k}^k))M \qquad \begin{array}{l} i \in I \\ k \in K \end{array} \tag{16}$$

$$\tau_j + \sigma_j + (x_{ji}^k + \bar{x}_{ji}^k) t_{ji} + \tilde{x}_{ji}^k (t_{jd^k} + t_{d^k i} + \beta) + \sum_{l=1,2,3} \sigma_l w_{jl}^k \tag{17}$$

$$\leq \tau_i + (1 - (x_{ji}^k + \bar{x}_{ji}^k + \tilde{x}_{ji}^k))M$$

$$\begin{array}{l} (i \neq j) \in I \\ k \in K \end{array}$$

$$a^k + x^k_{\theta^k i} t_{\theta^k i} + \tilde{x}^k_{\theta^k i}(t_{\theta^k d^k} + t_{d^k i} + \beta) \leq \tau_i + (1 - (x^k_{\theta^k i} + \tilde{x}^k_{\theta^k i}))M \qquad \begin{matrix} i \in I \\ k \in K \end{matrix} \tag{18}$$

$$\sum_{r \in F_i} f^r_i a^r_i \leq \tau_i = \sum_{r \in F_i} f^r_i b^r_i \qquad i \in I \tag{19}$$

$$\sum_{r \in F_i} f^r_i = \sum_{k \in K} y^k_i \qquad i \in I \tag{20}$$

Break constraints

$$\sum_{i \in I} w^k_{il} + \bar{w}^k_l + \bar{u}^k = 1 \qquad \begin{matrix} k \in K \\ l = 1, 2, 3 \end{matrix} \tag{21}$$

$$w^k_{il} \leq y^k_i \qquad \begin{matrix} i \in I \\ k \in K \\ l = 1, 2, 3 \end{matrix} \tag{22}$$

$$\tau_i + \sigma_i \leq \tau^k_l + (1 - w^k_{il})M \qquad \begin{matrix} i \in I \\ k \in K \\ l = 1, 2, 3 \end{matrix} \tag{23}$$

$$a_l(1 - \bar{w}^k_l - \bar{u}_k) \leq \tau^k_l \leq b_l(1 - \bar{w}^k_l - \bar{u}_k) \qquad \begin{matrix} k \in K \\ l = 1, 2, 3 \end{matrix} \tag{24}$$

Domain restrictions

$$\tau_i, \tau^k_l, \delta^k \geq 0 \tag{25}$$

$$x^k_{ij}, \bar{x}^k_{ij}, \tilde{x}^k_{ij}, y^k_i, f^r_j, w^l_i, \bar{w}^k_l, \bar{u}^k \in \{0, 1\} \tag{26}$$

This model contains three different types of x variables to account for the three possible cases along a technician route: technician k travels from i to j and has not previously visited his depot (x^k_{ij}); technician k travels from i to j and has previously visited his depot ($\bar{x}^k_{ij}$); technician k visits his depot while traveling from i to j ($\tilde{x}^k_{ij}$). Clearly, coherence among these three types of variables must be maintained along each route.

The main components of our model can now be described.

- The objective function (1) maximizes the total gain minus the total distance and total overtime. The fourth component is used to force the technicians to take their breaks. Each component is weighted by a different parameter.

- Constraints (2) and (3) ensure that each task is visited by at most one technician with the required skills.
- Constraint (4) establishes a relationship between variables x, $\tilde{x}$ and $\bar{u}^k$. It states that a technician is inactive if he does not depart from his home base.
- Constraints (5) and (6) ensure the coherence between the values of variables x, $\tilde{x}$, $\bar{x}$ and y. Constraint (5) states that task i must be assigned to technician k if he reaches this task either from his home base or from another task. Constraint (6) establishes a relationship among the three types of x variables. That is, if a technician travels from task i to return to his home base or to visit another task and if he has already visited the depot, then this visit to the depot must have taken place just before reaching i or anywhere else along the route before reaching i.
- Constraints (7) to (9) are the flow conservation constraints. Constraints (7) and (8) state that a technician can depart from the depot at most once and if he departs then he must return to the depot. Constraint (9) corresponds to the flow conservation constraints for each task.
- Constraint (10) states that a technician must visit his depot before serving a task that requires a special part.
- Constraints (11) and (12) ensure that a technician has the required spare parts to perform a task. Constraint (11) states the following: assuming that task j is visited just before task i, the amount of available spare after the service of j, plus any additional parts obtained by visiting the depot while traveling from j to i should cover the requirements of task i. Constraint (12) covers the case when task i is visited directly from the depot.
- Constraints (13) and (14) are aimed at updating the inventory after performing a task. Again, constraint (14) covers the case when task i is visited directly from the depot.
- Constraint (15) forces the maximum travel distance of each technician to be satisfied. This constraint takes into account the three different types of x variables introduced above while considering, at the same time, if i is the first, the last or an intermediary task along the route.
- Constraint (16) defines the overtime of each technician. If i is the last task in a technician route, then the return time at the depot is derived from the service start time of task i (i.e., service start time plus service time plus travel time to the home base). Then, the return time at the depot minus the end time of a regular workday provides the overtime (if any).
- Constraints (17) and (18) ensure the time continuity of each route. Basically, if task i is visited after j in the route of technician k, then the service start

time at j plus the service time plus any break time of technician k plus the time to travel to i (either directly or through the depot) should not exceed the service start time of task i.

- Constraints (19) and (20) are related to the time windows. Constraint (19) forces the service start time of each task to take place within one of the multiple time windows. Constraint (20) establishes a relationship between variables of type f and y. Basically, a task can be visited within one of its time windows if and only if it is assigned to a technician.
- Constraints (21) to (24) force each technician to take his breaks at an appropriate time. Constraint (21) states that a technician is either active and takes a break, active and does not take a break or inactive (for each one of the morning, mid-day and afternoon breaks). Constraint (22) establishes a relationship between variables of type w and y. Basically, a technician can only take a break after some task i if he has been assigned to this task. Constraint (23) ensures the time continuity of a technician route. If a break is taken by a technician after task i, then the service start time at i plus the service time should not exceed the start time of the break. Finally, constraint (24) forces the start time of the break to take place between the appropriate time bounds.
- Constraints (25) and (26) define the domain of the variables.

5 Test instances

An instance generator was developed for testing purposes. In the following, we explain how the various characteristics of each instance were generated.

1. *Service area.* The service area corresponds to a 40 km × 40 km or 50 km × 50 km squared area.
2. *Depot Location.* Each depot is randomly located within the service area.
3. *Task location.* Each task is randomly located within the service area.
4. *Task duration.* The duration or service time of each task is randomly chosen between 30 and 45 minutes.
5. *Task gain.* The gain associated with a task in randomly chosen between 1 and 10.
6. *Technician home base.* The first two technicians are located at the opposite ends of the service area (along the diagonal). The other technicians are located randomly within the service area.

7. *Technician skills.* With each technician is associated the percentage of tasks that he can perform, which is chosen from 100%, 50% and 25%. This percentage translates into a probability for each technician when a task is generated.

8. *Parts.* The number of spare parts needed to perform a task is randomly chosen between 0 and 3. A special part is required with a given probability.

9. *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 set between 2 and 3 hours after the upper bound of the previous time window.

Figure 2 shows an example with 20 tasks, 2 depots and 3 technicians within a 40 km × 40 km service area. The crow fly distance is assumed between each pair of locations and the speed of the vehicles is set at 50 km/h.

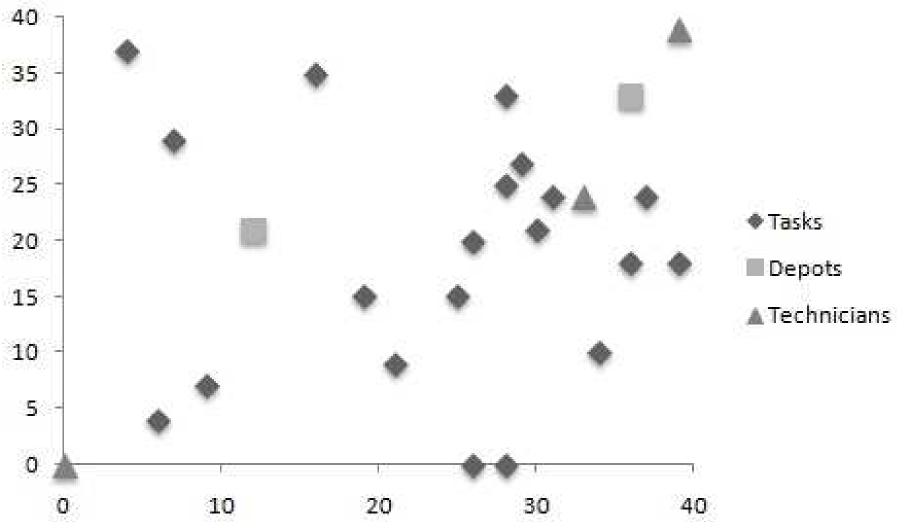

Figure 2: An example with 20 tasks, 3 technicians and 2 depots

A total of $2 \times 2 \times 4 = 16$ subsets of instances, with 5 instances in each subset, are generated by considering every possible combination of the parameter values shown in Table 1. After reporting the results obtained with CPLEX on these instances, we evaluate the individual impact of parameters Skills, Service time, Special parts and # Technicians, by considering different values for each one of them, while keeping the other parameter values fixed to those in Table 1. Table 2 presents the new value(s) considered for each parameter.

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 automat-

Table 1: Basic parameter values

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

Table 2: New parameter values

Skills 1	50% (2 technicians), 25% (1 technician)
Skills 2	100% (3 technicians)
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

ically takes the tasks that cannot be performed by the first technician. In this way, every task is covered. Also, when the number of technicians is 4, the fourth technician can perform 25% of all tasks. That is, the basic skill configuration shown in Table 1 becomes 100% (1 technician), 50% (1 technician) and 25% (2 technicians).

Finally, the maximum distance traveled by each technician during his workday was set to 125 km, while the weight parameters in the objective function were set to $\psi = 5$ (overtime), $\vartheta = 5$ (total distance), $\varphi = 550$ (total gain) and $\upsilon = 50$ (breaks). Thus, more emphasis is given to the total gain over all performed tasks. The value of parameter υ is also relatively high to force the technicians to take their breaks.

6 Computational results

This section reports the results obtained with CPLEX 12.6 on the subsets of instances introduced in the previous section. The solver was given a maximum of 24 hours of computation time on a 3.07GHz Intel Xeon X5675 processor. We recall that each subset is made of 5 different instances. The tables of results provide the following information (in this order):

- *Name.* The name of each subset of instances. It has the form X-Y-Z, where X corresponds to the time windows (either N for narrow or W for wide), Y to the size of the service area (either 40 for 40kms × 40kms or 50 for 50kms × 50 kms) and Z to the number of tasks (either 10, 15, 20 or 25).

- *# Opt*: number of instances solved to optimality (the number of instances for which an integer solution was obtained, without any proof of optimality, is put between parentheses);
- *CPU*: average computation time in hours:minutes:seconds (only for instances solved to optimality);
- *Gap*: average gap in percentage (only for instances not solved to optimality)
- *# Tasks*: average number of tasks performed by the technicians;
- *# Idle*: average number of idle (unassigned) technicians.

Table 3 reports the results obtained with the basic parameter values shown in Table 1. We observe that CPLEX can solve all instances of size 10 within minutes. However, CPLEX begins to strive when the number of tasks increase to 15. Only 9 instances out of 20 are solved to optimality and the computation times now vary between 7 and 16 hours. Only two instances of size 20 are solved to optimality and none of size 25. It should be noted that two technicians are often enough to cover all tasks on the smallest instances of size 10 (as indicated by the average number of idle technicians). Also, increasing the size of the service area leads to a fewer number of served tasks, which is quite understandable due to (1) the increased travel times between locations and (2) the maximum travel distance constraint of each technician which is now binding in some cases.

Considering now Table 4, we observe that the instances are generally easier to solve when the percentage of special parts increases. In particular, the number of instances that are solved to optimality increases from 26 (no special parts) to 31 (25% of special parts). In fact, the need for special parts constrains the solution space, thus reducing the combinatorial complexity.

Table 5 reports the results when the skills are modified. When each technician has all the required skills (which, in fact, eliminates this distinctive characteristic from the problem), the instances become more difficult to solve due to the increased flexibility in the assignment of technicians to tasks. Conversely, when the skills of the technicians are more restricted, a larger number of optimal solutions emerges, even for instances of size 25 (at the expense of a substantial increase in computation times).

The impact of service times is reported in Table 6. We observe a slight increase in the number of idle technicians and number of tasks performed by the technicians when the service time is reduced (except for W-50-25). This result makes sense because each technician can now serve more tasks during his workday. Finally, and as expected, Table 7 shows that the number of idle technicians and number of tasks performed by the technicians increase when an additional technician is available.

Overall, these results show that the technicians' skills have a significant impact on the difficulty of the problem. The percentage of special parts and the service times also have an impact, although to a lesser extent. Given that CPLEX can only return optimal solutions on instances of small size, the problem considered in this work is clearly a difficult one and deserves further studies.

7 Conclusion

We have introduced a technician routing and scheduling problem motivated from an application in the domain of electronic transactions equipment. A mixed integer programming formulation is proposed and solved with CPLEX. The results show that the problem is very difficult, given that only instances of small size can be solved to optimality. Our next step will be to consider a column generation approach that should allow larger instances to be solved. Metaheuristics will also be experimented with, thus sacrificing optimality in the hope of solving even larger instances.

Table 3: Basic configurations

Name	# Opt	CPU	Gap	# Tasks	# Idle
N-40-10	5	00:13:59	0%	9.6	0.8
N-40-15	2(3)	16:37:55	5%	14	0.2
N-40-20	0(5)	-	10%	17	0
N-40-25	0(5)	-	12%	20.6	0
N-50-10	5	00:09:02	0%	8.4	0.8
N-50-15	3(2)	08:10:29	10%	11.8	0.2
N-50-20	1(4)	15:17:09	17%	15.6	0.2
N-50-25	0(5)	-	18%	18.6	0
W-40-10	5	00:12:25	0%	10	0.6
W-40-15	1(4)	12:49:02	3%	14	0.4
W-40-20	0(5)	-	9%	17	0.2
W-40-25	0(5)	-	9%	21	0
W-50-10	5	00:10:33	0%	8.4	0.8
W-50-15	3(2)	07:05:31	10%	12.2	0.2
W-50-20	1(4)	07:02:50	17%	15.6	0.2
W-50-25	0(3)	-	12%	19.33	0

Name	**No special parts (0%)**					**Basic configurations (12.5%)**					**More special parts (25%)**				
	# Opt	CPU	Gap	# Tasks	# Idle	# Opt	CPU	Gap	# Tasks	# Idle	# Opt	CPU	Gap	# Tasks	# Idle
N-40-10	5	00:26:23	0%	9.8	0.8	5	00:13:59	0%	9.6	0.8	5	00:08:36	0%	9.6	1
N-40-15	1(4)	10:25:01	4%	13.8	0.2	2(3)	16:37:55	5%	14.0	0.2	1(4)	08:28:50	4%	14.0	0.2
N-40-20	0(5)	-	10%	16.4	0.2	0(5)	-	10%	17.0	0	0(5)	-	9%	17.0	0
N-40-25	0(5)	-	12%	20.6	0	0(5)	-	12%	20.6	0	0(5)	-	12%	19.8	0
N-50-10	5	00:16:15	0%	8.4	0.8	5	00:09:02	0%	8.4	0.8	5	00:05:45	0%	8.4	0.8
N-50-15	3(2)	07:28:21	11%	12.4	0.2	3(2)	08:10:29	10%	11.8	0.2	3(2)	03:58:37	8%	11.8	0.2
N-50-20	0(5)	-	16%	15.8	0.2	1(4)	15:17:09	17%	15.6	0.2	1(3)	13:08:14	15%	15.75	0
N-50-25	0(5)	-	20%	18.2	0	0(5)	-	18%	18.6	0	0(4)	-	14%	19.0	0
W-40-10	5	00:21:04	0%	10.0	0.8	5	00:12:25	0%	10.0	0.6	5	00:11:04	0%	10.0	0.6
W-40-15	0(5)	-	3%	14.2	0.6	1(4)	12:49:02	3%	14.0	0.4	2(3)	06:27:19	4%	14.0	0.4
W-40-20	0(5)	-	7%	17.6	0.2	0(5)	-	9%	17.0	0.2	0(5)	-	7%	17.6	0
W-40-25	0(5)	-	8%	21.6	0	0(5)	-	9%	21.0	0	0(5)	-	10%	21.0	0
W-50-10	5	00:21:27	0%	8.4	0.8	5	00:10:33	0%	8.4	0.8	5	00:08:07	0%	8.4	0.6
W-50-15	2(3)	10:18:42	9%	12.2	0.2	3(2)	07:05:31	10%	12.2	0.2	3(2)	07:22:17	9%	12.0	0.2
W-50-20	0(5)	-	14%	16.0	0.2	1(4)	07:02:50	17%	15.6	0.2	1(4)	06:57:52	17%	15.6	0.2
W-50-25	0(5)	-	21%	18.8	0	0(3)	-	12%	19.33	0	0(5)	-	19%	18.8	0

Table 4: Special parts

Name	Reduced skills					Basic configurations					All skills				
	# Opt	CPU	Gap	# Tasks	# Idle	# Opt	CPU	Gap	# Tasks	# Idle	# Opt	CPU	Gap	# Tasks	# Idle
N-40-10	5	00:00:13	0%	9.8	0.6	5	00:13:59	0%	9.6	0.8	5	04:32:46	0%	10	0.8
N-40-15	5	00:20:42	0%	14	0.2	2(3)	16:37:55	5%	14	0.2	0(5)	-	1%	15	0
N-40-20	3(2)	02:32:16	4%	16.6	0	0(5)	-	10%	17	0	0(5)	-	2%	19	0
N-40-25	1(4)	06:51:25	9%	20.6	0	0(5)	-	12%	20.6	0	0(5)	-	4%	24	0
N-50-10	5	00:00:15	0%	8.4	0.2	5	00:09:02	0%	8.4	0.8	5	02:04:45	0%	10	0.8
N-50-15	5	00:14:10	0%	11.6	0	3(2)	08:10:29	10%	11.8	0.2	0(5)	-	1%	15	0
N-50-20	3(2)	01:18:27	7%	15	0	1(4)	15:17:09	17%	15.6	0.2	0(5)	-	2%	19.6	0
N-50-25	3(2)	16:50:36	7%	17.8	0	0(5)	-	18%	18.6	0	0(5)	-	5%	24	0.2
W-40-10	5	00:00:04	0%	10	0.2	5	00:12:25	0%	10	0.6	5	02:44:09	0%	10	0.8
W-40-15	5	00:20:30	0%	13.6	0.2	1(4)	12:49:02	3%	14	0.4	0(5)	-	1%	15	0
W-40-20	3(2)	04:00:09	4%	17.2	0	0(5)	-	9%	17	0.2	0(5)	-	1%	20	0
W-40-25	1(4)	21:50:40	10%	21	0	0(5)	-	9%	21	0	0(5)	-	2%	24.6	0
W-50-10	5	00:00:07	0%	8.4	0.2	5	00:10:33	0%	8.4	0.8	5	01:49:19	0%	10	0.8
W-50-15	5	00:17:07	0%	12	0	3(2)	07:05:31	10%	12.2	0.2	0(5)	-	1%	15	0
W-50-20	3(2)	02:13:02	13%	15.2	0	1(4)	07:02:50	17%	15.6	0.2	0(5)	-	2%	20	0
W-50-25	3(2)	06:35:34	7%	17.6	0	0(3)	-	12%	19.33	0	0(5)	-	6%	22.6	0

Table 5: Skills

Name	Basic configurations (15-45 min)					Reduced service times (15-30 min)					Reduced service times (10-20 min)				
	# Opt	CPU	Gap	# Tasks	# Idle	# Opt	CPU	Gap	# Tasks	# Idle	# Opt	CPU	Gap	# Tasks	# Idle
N-40-10	5	00:13:59	0%	9.6	0.8	5	00:06:53	0%	10	1	5	00:08:48	0%	10	1
N-40-15	2(3)	16:37:55	5%	14	0.2	2(3)	12:22:50	3%	14	0.6	2(3)	10:52:24	3%	14	0.6
N-40-20	0(5)	-	10%	17	0	1(4)	11:54:04	6%	17.4	0.2	1(4)	10:51:57	8%	17.4	0
N-40-25	0(5)	-	12%	20.6	0	0(5)	-	10%	21	0	0(4)	-	7%	21.75	0
N-50-10	5	00:09:02	0%	8.4	0.8	5	00:11:09	0%	8.4	0.8	5	00:10:47	0%	8.4	0.8
N-50-15	3(2)	08:10:29	10%	11.8	0.2	3(2)	05:58:33	9%	11.8	0.6	3(2)	05:50:03	9%	11.8	0.6
N-50-20	1(4)	15:17:09	17%	15.6	0.2	1(4)	09:24:39	16%	15.8	0.6	1(4)	09:38:45	16%	15.8	0.6
N-50-25	0(5)	-	18%	18.6	0	0(5)	-	15%	19.4	0	0(5)	-	16%	19.4	0
W-40-10	5	00:12:25	0%	10	0.6	5	00:10:07	0%	10	1	5	00:11:34	0%	10	1
W-40-15	1(4)	12:49:02	3%	14	0.4	3(2)	10:33:01	4%	14.4	0.8	3(2)	11:35:29	4%	14.4	0.8
W-40-20	0(5)	-	9%	17	0.2	0(5)	-	5%	17.8	0.4	0(5)	-	4%	18.2	0.4
W-40-25	0(5)	-	9%	21	0	0(5)	-	7%	21.6	0	0(5)	-	6%	22	0
W-50-10	5	00:10:33	0%	8.4	0.8	5	00:10:53	0%	8.4	1	5	00:12:42	0%	8.4	1
W-50-15	3(2)	07:05:31	10%	12.2	0.2	3(2)	05:55:33	9%	12.2	0.4	3(2)	06:24:20	10%	12.2	0.4
W-50-20	1(4)	07:02:50	17%	15.6	0.2	1(4)	07:12:34	18%	15.8	0.6	1(4)	07:20:44	18%	15.8	0.6
W-50-25	0(3)	-	12%	19.33	0	0(5)	-	18%	18.4	0	0(5)	-	16%	19.2	0

Table 6: Repair times

Name	**Basic configurations (3 technicians)**					**More technicians (4 technicians)**				
	# Opt	CPU	Gap	# Tasks	# Idle	# Opt	CPU	Gap	# Tasks	# Idle
N-40-10	5	00:13:59	0%	9.6	0.8	5	00:28:40	0%	10	1.6
N-40-15	2(3)	16:37:55	5%	14	0.2	1(4)	23:13:39	2%	14.8	0.6
N-40-20	0(5)	-	10%	17	0	0(5)	-	2%	20	0
N-40-25	0(5)	-	12%	20.6	0	0(5)	-	3%	24	0
N-50-10	5	00:09:02	0%	8.4	0.8	5	00:09:08	0%	10	1
N-50-15	3(2)	08:10:29	10%	11.8	0.2	4(1)	10:45:13	4%	14.6	0
N-50-20	1(4)	15:17:09	17%	15.6	0.2	0(5)	-	5%	18	0
N-50-25	0(5)	-	18%	18.6	0	0(5)	-	8%	22	0
W-40-10	5	00:12:25	0%	10	0.6	5	00:25:42	0%	10	1.6
W-40-15	1(4)	12:49:02	3%	14	0.4	1(4)	16:52:17	1%	15	0.6
W-40-20	0(5)	-	9%	17	0.2	0(5)	-	2%	20	0.2
W-40-25	0(5)	-	9%	21	0	0(5)	-	4%	24.2	0
W-50-10	5	00:10:33	0%	8.4	0.8	5	00:08:10	0%	10	1
W-50-15	3(2)	07:05:31	10%	12.2	0.2	4(1)	08:14:21	5%	14.6	0
W-50-20	1(4)	07:02:50	17%	15.6	0.2	0(5)	-	4%	18.4	0
W-50-25	0(3)	-	12%	19.33	0	0(5)	-	11%	22.2	0

Table 7: Number of technicians

References

[1] F. Blakeley, B. Arguello, B. Cao, W. Hall, and J. Knolmajer. Optimizing periodic maintenance operations for schindler elevator corporation. *Interfaces*, 33(1):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 Bruce Golden, S. Raghavan, and Edward Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 503–525. Springer US, 2008.

[3] J-F. Cordeau, G. Laporte, F. Pasin, and S. Ropke. Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13(4):393–409, 2010.

[4] B.L. Golden, A.A. Assad, and E.A. Wasil. Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 245–286. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[5] 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(1):143–161, 2011.

[6] B. Kallehauge, J. Larsen, O.B. Madsen, and M.M. Solomon. Vehicle routing problem with time windows. In G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors, *Column Generation*, pages 67–98. Springer, 2005.

[7] 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, Illinois, 1976.

[8] V. Pillac, C. Gueret, 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.

[9] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

[10] 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(3):331–340, 1993.

[11] H. Tang, E. Miller-Hooks, and R. Tomastik. Scheduling technicians for planned maintenance of geographically distributed equipment. *Transportation Research Part E: Logistics and Transportation Review*, 43(5):591 – 609, 2007.

[12] 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(3):119 – 127, 1997.

[13] D. Weigel and B. Cao. Applying GIS and OR techniques to solve Sears technician-dispatching and home delivery problems. *Interfaces*, 29(1):112–130, January 1999.

[14] J. Xu and S.Y. Chiu. Effective heuristic procedures for a field technician scheduling problem. *Journal of Heuristics*, 7(5):495–509, 2001.