Decision support for the technician routing and scheduling problem

Mette Gamst | David Pisinger

DTU Management, Technical University of Denmark, Kgs. Lyngby, Denmark

Correspondence
David Pisinger, DTU Management, Technical University of Denmark, Kgs. Lyngby, Denmark. Email: pisinger@man.dtu.dk

Funding information
Innovationsfonden, Grant/Award Number: 0224-00055B

Abstract
The technician routing and scheduling problem (TRSP) optimizes routes for technicians serving tasks subject to qualifications, time constraints, and routing costs. In the literature, the TRSP is solved either to provide actual technician work schedules or to perform what-if analyses on different TRSP scenarios. A TRSP scenario consists of a given number of tasks, technicians, skills, working hours and so forth. We present a method which builds optimal TRSP scenarios with respect to technician fleet, their skills, their working hours and digitization of task equipment. The scenarios are built such that the combined TRSP costs (OPEX) and investment costs (CAPEX) are minimized. By using a holistic approach we can generate scenarios that would not have been found by studying the investments individually. The proposed method consists of a matheuristic based on column generation. To reduce computational time, the routing costs of a technician are estimated instead of solved to optimality. The proposed method is evaluated on data from the literature and on real-life data from a telecommunication company. The evaluation shows that the proposed method successfully suggests attractive scenarios. The method especially excels in ensuring that more tasks are serviced, but also in reducing travel time with around 16% in the real-life instance. We believe that the proposed method could constitute an important strategic tool for routing companies. In the conclusion, we propose future research directions to extend the applicability.

KEYWORDS
adaptive large neighborhood search, column generation, decision support, field service, scenario generation, technician routing and scheduling

1 | INTRODUCTION

The technician routing and scheduling problem (TRSP) is the problem of assigning tasks to technicians subject to skill sets, time windows, travel times and routing costs. The problem occurs in telecommunications, where technicians perform installation or fault correction tasks. It is closely related to more general workforce scheduling problems such as general maintenance scheduling, home care scheduling and so forth [12,38]. The TRSP and workforce scheduling problems have been extensively studied in the research literature, however, mainly with an operational focus on generating actual work schedules [7].

This article considers the strategic decisions of determining the right set of technicians, technician qualifications, task outsourcing or digitization and working hours such that the operational costs can be minimized. The strategic decisions all come with an investment cost. Traditionally, strategic questions are analyzed through what-if analyses. Given a base case and
a scenario, for example, with a distinct set of technicians, the difference in the operational costs indicates if the scenario is an attractive investment [26]. We propose a method which minimizes the combined investment costs and operational costs (CAPEX + OPEX) to generate attractive scenarios. To the best of our knowledge, this has not been proposed previously for the TRSP. In a deterministic and optimal setting, this replaces the traditional what-if approach. In practice, however, the approach can be viewed as a scenario builder, where the scenarios afterwards can be evaluated in detail using what-if analysis. By considering all investment decisions in a holistic model, we are able to study scenarios that would not have been found by traditional what-if tools that analyze investment decisions one by one.

To reach computational tractability, the proposed method approximates the TRSP by solving a facility location problem (FLP) assigning tasks to technicians, instead of optimizing each technician path. We show that this is a fair approximation by comparing it to the TRSP solved by an adaptive large neighborhood search (ALNS) metaheuristic. The assignment problem is solved through column generation, where the master problem handles the investment decisions and the subproblem assigns tasks to each technician. The method is computationally evaluated on data from the literature and on real-life TRSP data.

The main contribution of this article is:

- A new method to optimize investment decisions in the TRSP using a holistic approach.
- Evaluation of investment decisions in the technician fleet, overtime, skill sets, and digitization of tasks on benchmark data and real-life TRSP data.
- An approximation of the TRSP based on assigning tasks to technicians.
- An ALNS algorithm for the TRSP.
- Publication of a real-life large TRSP dataset.

This article is organized as follows: the TRSP is formally defined in Section 2. Relevant literature is reviewed in Section 3. Then follows the proposed solution method, consisting of approximating TRSP with a task assignment model in Section 4, and of extending this approach with investment decisions in Section 5. Section 6 presents test data, Section 7 computational experiments, and Section 8 future work. Finally, Section 9 concludes the article.

## 2 PROBLEM DEFINITION

The TRSP considered in this article is formally defined. Let $T$ be a set of technicians, $V$ the set of tasks, and $S$ the set of all skills. Each technician $t \in T$ has a start and endpoint (depot) $d_t$, skills set $S(t)$ and a set of work shifts $D(t)$ consisting of time windows $[a_{ij}, b_{ij}]$. Each task $i \in V$ consists of the required skills set $S(i)$, a duration $f_i$, a penalty for not being serviced $c_i$, and a time window $[a_i, b_i]$. It is assumed that every task can be solved by a single technician.

We generate a workday technician for each workday for each technician in the set $T$. Let the set of workday technicians be denoted $T$. Each workday technician $t \in T$ has a start and endpoint (depot) $d_t$, skills set $S(t)$ and time window $[a_t, b_t]$. The set of tasks that can be served by a workday technician $t \in T$ subject to skills and time windows is denoted $V(t)$. The travel time between any two locations is denoted $c_{ij}$. Travel times are assumed to satisfy the triangle inequality. We introduce binary variables $x_{ij}$ to denote if technician $t$ travels from task $i$ to $j$, binary variables $y_i$ to denote if task $i$ is unassigned, and linear variables $d_{ij}$ to denote the arrival time at task $i$ for technician $t$.

The TRSP consists of minimizing the cost of servicing tasks subject to skills and time windows. A MIP formulation of the problem is:

$$\begin{align*}
\text{min} & \quad \sum_{t \in T} \sum_{i \in V(t)} \sum_{j \in V(t)} c_{ij} x_{ij} + \sum_{i \in V(t)} c_i y_i, \\
\text{s.t.} & \quad \sum_{t \in T} \sum_{j \in V(t)} x_{ij} + y_i \geq 1 \quad \forall i \in V, \\
& \quad \sum_{i \in V(t)} x_{ij} \leq 1 \quad \forall t \in T, \\
& \quad \sum_{j \in V(t)} x_{ij} - \sum_{j \in V(t)} x_{ji} = 0 \quad \forall t \in T, \forall i \in V(t), \\
& \quad \sum_{i \in V(t)} x_{id_i} \leq 1 \quad \forall t \in T,
\end{align*}$$

(1)
The objective function \( (1) \) minimizes the total travel time and the total penalty for unserved tasks. The penalty is given in time measurement to make the two terms in the objective function comparable. The first constraints \( (2) \) ensure that a task is either served or marked as unserved. The next three constraint sets \( (3)–(5) \) say that a workday technician can leave home at most once, must leave any entered task and return to home at most once. The final three constraint sets \( (6)–(8) \) satisfy time windows; constraints \( (6) \) keep track of start times at each visited task for each workday technician. Constraints \( (7) \) and \( (8) \) ensure that task start times are within both the time windows of the task and of the technician. Finally, bounds \( (9)–(11) \) ensure that variables take on feasible values. Note that the bound on variables \( y_i \) can be relaxed without any loss of information.

It is noted that the formulation bears strong resemblance to the multi-depot vehicle routing problem (VRP) with time windows [29] and only differs in the definition of the set \( V(t), t \in T \), and that tasks may be unserved.

3 | LITERATURE REVIEW

The TRSP was introduced by Tsang and Voudouris with application to British Telecom [48]. They solved the TRSP using a guided local search. The TRSP has since been extensively studied and mainly solved using metaheuristics [34,40]. Few contributions attempt to solve the problem to optimality [32,33] and only succeed on small instances with at most 45 customers. In comparison most of the heuristic approaches consider instances with up to 200 customers. We refer to Chen et al. [9] and Pillac et al. [39] for recent literature overviews. Most literature contributions consider real-life applications including specific extra constraints. Pillac et al. [39] consider the dynamic TRSP, where new customers may arrive during the planned period. They propose an ALNS algorithm and solve instances with up to 200 customers and 25 technicians. Mathlouthi et al. [34] consider a similar problem and apply tabu search with integrated adaptive memory. They generate a set of elite solutions, which are stored in memory, and which are used to construct new starting solutions for the tabu search. The approach has also been applied to the VRP [21,49]. In another variant of the TRSP, service times may differ depending on how experienced a technician is, and on the fact that technicians may become more and more experienced and thus faster. Chen et al. [9] take this into account and solve the problem heuristically. Cortes et al. [10] use a TRSP algorithm by [11] to simulate a complete week of operation for different fleet sizes (number of technicians). In this way various key performance indicators can be evaluated for each fleet size, and an optimal fleet size can then be selected. Apart from this article, the TRSP research literature only considers the problem of generating work schedules and/or technician teams, while the more strategic decisions, such as the number and location of technicians or their skills set, have yet to be explored.

In recent years, the TRSP and related scheduling problems, such as home care scheduling, have been considered in more overall terms as workforce scheduling problems. Recent surveys consider these closely related problems to promote learning across the problem types and to emphasize similarities in solution approaches as well as problem instance types [12,38]. Home care scheduling problems frequently contain synchronization constraints (a task needs two or more technicians at the same time) or precedence constraints (a task must be finished before another) [41].

The VRP is related to the TRSP, and is an extensively studied problem. Relevant VRP variants are; the multi-depot VRP [43]; the heterogeneous VRP [43], where the vehicles differ; the site dependent VRP, where each customer can only be served by a subset of the vehicles [8]; the vehicle fleet mix problem, where the usage of each vehicle is associated with a cost [28]; and the private fleet common carrier problem, where customers can be served by a mix of common and private vehicles and where the private vehicle cost does not depend on the routing [14]. In the TRSP, a penalty is associated with skipping a customer. This feature can also be found in the VRP related problem named the team orienteering problem (TOP). Here, a score is associated with visiting customers. The TOP has received significant attention in the research literature, with regards to both exact and heuristic methods [24]. A survey on the VRP and its variants can be found in [27]. The vehicle fleet mix problem takes on a more strategic view by combining investment costs of using a vehicle with the operational costs of serving customers [22]. Many mathematical formulations and tailored heuristics have been proposed for the problem, see, for example, [28,30,44], but they are not easily extended to other strategic investment decisions. Other interesting work on
VRP related problems includes Gaudioso and Paletta [20] who suggest an alternative heuristic for the tactical problem of minimizing fleet size, rather than the operational problem of reducing travel costs. Most other papers focus on creating actual vehicle plans.

Another interesting research contribution is the work of Mourgaya and Vanderbeck [35], who solve a tactical version of the periodic VRP in which visit schedules and customer assignments to vehicles are solved simultaneously. The sequencing of customers within vehicle paths is determined in an operational problem. The authors consider two objectives: a “workload balancing” objective that ensures an equal distribution of customers among vehicles and a “regionalization” objective that clusters customers geographically as a proxy for tour length. Focusing solely on the tactical problem facilitates the solution of larger problem instances.

The FLP combines the cost of opening facilities with the cost of serving customers [13]. This is somewhat similar to combining investment decisions in, for example, technicians with the routing costs in the TRSP. Real-life facility location applications include opening of factories, hospitals, and placement of electric vehicle chargers [15]. Numerous solution methods have been proposed for solving the FLP, including MIP solvers, Lagrange relaxation, Benders decomposition and a variety of heuristics, see, for example, the survey on FLPs in health care [1]. The FLP differs from the investment decision problem in TRSP in the lack of routing, and in that only one type of investment decisions is possible; the opening of a facility. The FLP has frequently been used as an approximation of the VRP. For instance, Branco and Coelho [4] used the FLP to solve a kind of multi-depot VRP. A similar approach was used by Bramsel and Simchi-Levi [3].

The idea of minimizing the combined CAPEX and OPEX is also present in other research areas. In energy analyses, the capacity expansion problem (CEP) determines how to invest in energy producing units subject to meeting energy demand as cheaply as possible [31]. For example, how much renewable energy to invest in [36]. The CEP is NP-hard, and often considers at least a yearly time horizon in hourly resolution in order to determine the best energy mix [5]. Attempts to achieve computational tractability includes Dantzig-Wolfe decomposition [17], Benders decomposition [47], and heuristics consisting of aggregating the time domain [5,25]. A popular approach for aggregating the time domain consists of clustering days to collapse the time horizon and thus reduce the problem size into tractable MIPs. The survey and computational results in [5] indicate that the clustering approaches have similar performance, and one could thus consider a simple approach such as k-means clustering.

Combining CAPEX and OPEX is also present in design problems, such as wind farm layout design problems. Here, the installation costs are considered together with wind production performance, which again depends on the wind farm layout and the resulting wake effects [16].

Combining CAPEX and OPEX in optimization is thus far from new. However, it does not seem to have gained much attention in the context of routing or the TRSP. This article seeks to remedy this research gap.

4 | TASK ASSIGNMENT FOR APPROXIMATING THE TECHNICIAN ROUTING AND SCHEDULING PROBLEM

The TRSP is NP-hard and is typically solved in the literature using meta-heuristics such as local neighborhood search algorithms, where the neighborhood consists of some method to swap tasks between technicians or permuting technician paths. It is not straightforward to include investment decisions in this approach for several reasons.

- An investment is often only attractive if it results in many tasks being assigned to a technician. The neighborhood search must thus be extended to consider groups of tasks instead of individual tasks, which again may result in very large neighborhoods and poor convergence.
- An investment is often only attractive if it is utilized over a larger time period, for example, months or even years. The investment cost can be amortized into daily costs, but the investment business case is very fragile if it is built on the TRSP for a single day. The TRSP can be solved for larger time periods using the mentioned local search heuristics, but then the neighborhood search must be extended to considering the path for a technician for each day during the entire time horizon to investigate the full potential of the investment. This is not straightforward.

Generally, heuristics are greedy and shortsighted in their nature, and investment decisions are only worthwhile considering for large time horizons. Instead we define a mathematical model for the investment decision problem. To reduce problem complexity we propose an assignment of tasks to technicians, similar to the tactical assignment approach of Mourgaya and Vanderbeck [35]. This is a simplification of solving the full TRSP, because travel times are estimated instead of optimized.

The remainder of this section first studies the task assignment approach, and subsequently an ALNS algorithm for the full TRSP is presented. We use the latter to evaluate the performance of the assignment approach.
4.1 Task assignment

The task assignment problem consists of assigning tasks to technicians subject to skills and the time windows of the task and technician. The task assignment problem does not optimize each technician path and does thus not check for all overlapping services or tasks, nor is the exact travel time computed. The travel time is estimated by considering the distance between a technician’s home depot and the location of assigned tasks similar to [3, 4].

The task assignment problem is defined by using the same set and parameter notation as in Section 2. Recall that a workday technician is generated for each working day for each technician. In the following, we denote the original technician as the master technician. Furthermore, we introduce a constant $k \geq 0$ to adjust the estimated travel time. Variable $x_i^j \in \{0, 1\}$ indicates if task $i \in V$ is assigned to workday technician $t \in T$. Variable $y_i \in \{0, 1\}$ decides if task $i \in V$ is not assigned to any workday technician. Finally, variable $z_t \geq 0$ contains an estimate on how far workday technician $t$ travels provided the current assignment. The task assignment problem is:

$$
\min \sum_{t \in T} z_t + \sum_{i \in V} c_i y_i, \tag{12}
$$

s.t. \hspace{1cm} z_t \geq k \cdot c_i (x_i^j + x_j^i - 1) \hspace{1cm} \forall i, j \in V(t), t \in T, \tag{13}

$$
\sum_{i \in V(t)} f_{i,t} x_i^j \leq Q_t - \sum_{i \in V} c_{d,i} x_i^k \hspace{1cm} \forall t \in T, \tag{14}
$$

$$
\sum_{t \in T : \forall i \in V(t)} x_i^j + y_i \geq 1 \hspace{1cm} \forall i \in V, \tag{16}
$$

$$
x_i^j + x_j^i \leq 1 \hspace{1cm} \forall i, j \in V(t) : overlap(i,j), \forall t \in T, \tag{17}
$$

$$
x_i^j \in \{0, 1\} \hspace{1cm} \forall i \in V(t), \forall t \in T, \tag{18}
$$

$$
y_i \geq 0 \hspace{1cm} \forall i \in V, \tag{19}
$$

$$
z_t \geq 0 \hspace{1cm} \forall t \in T. \tag{20}
$$

The objective function (12) minimizes the total estimated travel time and the penalty for unassigned tasks. The first two sets of constraints, (13) and (14), set the estimated travel time for a workday technician to be the longest travel time between any two assigned tasks or from the depot to an assigned task. The travel time is multiplied with factor $k$ to represent intermediate driving. Constraints (15) ensure that the workday technician capacity is satisfied, where we estimate time spent on traveling to be the travel time from the depot to each assigned task. Constraints (16) say that each task must either be assigned or marked as unassigned. Because of the nonnegative costs in the objective function, a task will never both be assigned and marked as unassigned. The last set of constraints (17) ensure that a technician is not assigned two tasks, which cannot be serviced by the same technician because of overlapping time windows. Bounds (18)–(20) ensure feasible variables values.

The set overlap($i,j$) in constraints (17) contains pairs of tasks $i,j$ which cannot be assigned to the same technician. For example, consider task $a$ with duration 3 and time window $[10, 14]$ and task $b$ with duration 5 and time window $[9, 17]$. Even with zero travel time between the two tasks, a technician cannot serve both: $a$ cannot start later than at time 11 and cannot finish earlier than time 13. $b$ cannot start later than 12 and cannot finish earlier than 14. Constraints (17) obviously only cover part of the time window restrictions of the full TRSP, and one must expect solutions to the task assignment problem that are infeasible to the full TRSP.

4.2 An adaptive large neighborhood search for the TRSP

To evaluate the investment decisions made by the task assignment approach, solutions to the full TRSP before investments and after investments should be compared. The TRSP is NP-hard and difficult to solve. As seen in Section 3, the literature suggests a wide variety of meta-heuristics. We apply an ALNS algorithm to solve the TRSP as formulated in mathematical model (1)–(11). ALNS is one of the best performing meta-heuristics on VRPs [42]. The ALNS iteratively destroys and repairs the current solution. After each destroy and repair operation, the resulting solution is evaluated, and if certain requirements are met,
it becomes the new current solution. Our implementation of ALNS follows the template outlined by Ropke and Pisinger [42]. We evaluate a solution by using the record-to-record method suggested in [45]. The destroy methods are:

- **Random destroy**: random tasks are removed from workday technician paths and marked as unassigned.
- **Route destroy**: a random workday technician path is selected and all tasks unassigned.

The repair methods are:

- **Greedy repair**: the cost of inserting each unrouted task in any path is calculated, and the cheapest task is inserted first.
- **Regret repair**: the cost of the best insert and the second best insert is calculated for each unrouted task. The task with the highest difference between the best and second best insert is inserted first [42].

When a task has been inserted or removed from a path, the arrival times in the path are updated to limit waiting times.

As a final optimization, the ALNS finishes with solving a set cover like formulation on the generated paths. Let \( P \) be the set of generated paths, each path with cost \( c_p, p \in P \) and let decision variable \( x_p \in \{0, 1\} \) indicate if path \( p \in P \) is selected or not:

\[
\min \sum_{p \in P} \sum_{t \in T} c_p x_p + \sum_{i \in V} c_{yi},
\]

s.t.

\[
\sum_{p \in P} x_p \leq 1 \quad \forall t \in T,
\]

\[
\sum_{p \in P} \delta_p x_p + y_i \geq 1 \quad \forall i \in V,
\]

\[
x_p \in \{0, 1\} \quad \forall p \in P, \forall t \in T,
\]

\[
y_i \geq 0 \quad \forall i \in V.
\]

The objective (21) minimizes the total costs. Constraints (22) ensure that no more than one path is used per technician, constraints (23) say that a task is either served or marked as unserved, and bounds (24) and (25) ensure that variables take on feasible values.

The number of paths \( P \) may be very large. We only consider the 25 000 most promising paths, where the potential of a path is defined by the value of the best ALNS solution it occurred in.

### 5 | INVESTMENT DECISIONS IN THE TASK ASSIGNMENT APPROACH

To optimize investment decisions, we minimize the combined OPEX and CAPEX costs. The task assignment problem is extended with investment decisions by including the investment costs in the objective function and by adding investment decision constraints. First, we need to introduce some additional notation and define the investment decisions to consider. Then follows the mathematical model for the problem.

Recall that the set \( T \) contains workday technicians. An investment decision may, however, affect the master technician and thus several workday technicians. Let \( T \) be the set of master technicians and let parameter \( \delta_m \in \{0, 1\} \) denote if \( m \in T \) is the master technician for \( t \in T \).

Let \( A \) be the set of investment decisions and variables \( u'_a, u''_a, u'_m, u''_m \in \{0, 1\} \) indicate if investment decision \( a \in A \) is performed, either for workday technician \( t \in T \), master technician \( m \in T \) or task \( i \in V \). Each investment decision has an associated cost, \( c'_a, c''_a, c'_m, c''_m \) resp. \( c'_i, c''_i \).

The considered investment decisions are:

- **Workday technician overtime**, that is, if a workday technician should work overtime. Investment decision variable is \( u'_t, u''_t \in \{0, 1\}, t \in T \) and the actual overtime minutes are given by parameter \( w_{ot} \).
- **Digitization of tasks**: Instead of assigning a task, an investment can be made in the underlying equipment to enable remote fault handling. Investment decision variable is \( u'_{dig}, u''_{dig} \in \{0, 1\}, i \in V \) and parameter \( \delta_{dig} \in \{0, 1\} \) indicate whether or not task \( i \in V \) is a candidate for digitization.
- **Skill upgrade**, that is, if the master technician should receive skill training. Investment decision variable is \( u''_m \in \{0, 1\}, m \in T \). The investment affects the skill ability of all corresponding workday technicians, that is, all \( t \in T : \delta_m = 1 \).
- **New technicians**, that is, if a new master technician should be employed. The skills set, working days and location of the new master technician are given as input. Let \( NT \) denote the corresponding set of new workday technicians, and let the investment decision variable be denoted \( u''_{nt} \in \{0, 1\} \).
The list could be extended to any other investment decisions, which would be handled in a similar manner. The mathematical model now becomes:

\[
\begin{align*}
\text{min} & \quad \sum_{t \in T} z_t + \sum_{i \in V} c_i \sum_{a \in A} y_i + \sum_{a \in A} \left( \sum_{t \in T} c_{t}^{a} u^{a}_t + \sum_{m \in T} c_{m}^{a} u^{m}_m + \sum_{i \in V} c_{i}^{a} u^{i}_i \right), \\
\text{s.t.} & \quad z_t \geq k \cdot c_{ij}(x^i_t + x^j_t - 1) \quad \forall i, j \in V, t \in T, \quad (27) \\
& \quad z_t \geq k \cdot c_{d,ij} x^j_t \quad \forall i \in V, t \in T, \quad (28) \\
& \quad \sum_{i \in V} f_i x^i_t \leq Q_t + w_{ol} \cdot u^{ol}_t - \sum_{i \in V} c_{d,ij} x^j_t \quad \forall t \in T, \quad (29) \\
& \quad \sum_{t \in T} x^i_t + y_i + \delta^\text{dig,} u^{\text{dig}}_t \geq 1 \quad \forall i \in V, \quad (30) \\
& \quad x^i_t + x^j_t \leq 1 \quad \forall t \in T, \forall i, j \in V : \text{overlap}(i, j), \quad (31) \\
& \quad x^i_t \leq \sum_{m \in T} \delta^m_{a,} u^m_m \quad \forall i \in V, \forall t \in T, \forall s \in S(i) : s \notin S(t), \quad (32) \\
& \quad x^i_t \leq \sum_{m \in T} \delta^m_{a,} u^m_{m} \quad \forall i \in V, \forall t \in NT, \quad (33) \\
& \quad x^i_t = 0 \quad \forall i \in V, \forall t \in T : \text{tw}(i, t) = \emptyset, \quad (34) \\
& \quad x^i_t \in \{0, 1\} \quad \forall i \in V, \forall t \in T, \quad (35) \\
& \quad y_i \geq 0 \quad \forall i \in V, \quad (36) \\
& \quad z_t \geq 0 \quad \forall t \in T, \quad (37) \\
& \quad u^{a}_t \in \{0, 1\} \quad \forall a \in A, \forall t \in T, \quad (38) \\
& \quad u^{m}_m \in \{0, 1\} \quad \forall a \in A, \forall m \in T, \quad (39) \\
& \quad u^{i}_i \in \{0, 1\} \quad \forall a \in A, \forall i \in V. \quad (40)
\end{align*}
\]

The objective (26) minimizes the total estimated travel time, the penalties for unassigned tasks, and the investment decision costs. The investment decision costs must be converted to equivalent time measurements to make the terms in the objective function comparable. The first two constraint sets (27) and (28) are unchanged and determine the estimated travel time. The next constraints (29) handle time capacity for each workday technician subject to overtime investments. Constraints (30) ensure that a task is either assigned, marked as unassigned or digitized if possible. Because of the nonnegative costs in the objective function, at most one of the three variable sets in the constraints will be set for each task. Constraints (31) ensure that a technician is not assigned two tasks, which cannot be serviced by the same technician because of overlapping time windows.

The next two constraint sets are new. Constraints (32) concern skill investment and state that if a required task skill, \( s \in S(i), i \in V \) is not readily available by a workday technician \( t \in T \), then the task cannot only be assigned to the workday technician if an investment decision is made to train the corresponding master technician \( m \in T : \delta^m_m = 1 \). Constraints (33) concern investment decision in new technicians and say that a task can only be assigned to a new workday technician, if the corresponding investment decisions is made in the corresponding master technician \( m \in T : \delta^m_m = 1 \). The final constraints (34) enforce that a task cannot be assigned to a workday technician if their time windows do not overlap. Here, \( \text{tw}(i, t) \) is the set of feasible start times that workday technician \( t \in T \) can perform task \( i \in V \) subject to task duration and time windows, including overtime. Bounds (35)–(40) ensure that variables are in the correct domain.

### 5.1 Column generation

The investment decision problem is decomposed into a master and subproblem. The master problem handles investment decisions, and the subproblem assigns tasks to a workday technician. Let variable \( x^i_p \in \{0, 1\} \) represent an assignment \( p \in P \) for
workday technician \( t \in T \) with estimated travel time \( c_p^t \). The master problem then becomes:

\[
\begin{align*}
\min & \quad \sum_{p \in P} \sum_{t \in T} c_p^t x_p^t + \sum_{i \in V} c_i y_i + \sum_{a \in A} \left( \sum_{t \in T} c_i^a u_a^t + \sum_{m \in T} c_m^a u_a^m + \sum_{i \in V} c_i^a u_a^i \right), \\
\text{s.t.} & \quad \sum_{p \in P} x_p^t \leq 1 \quad \forall t \in T, \\
& \quad \sum_{p \in P} x_p^t \leq \sum_{m \in T} \delta_m^t u_m^t \quad \forall t \in NT, \\
& \quad \sum_{p \in P} \delta_{oa}^t x_p^t \leq u_o^t \quad \forall t \in T, \\
& \quad \sum_{p \in P} \delta_{ot}^t x_p^t + y_i \geq \delta_{dug}^i u_{dig}^i \geq 1 \quad \forall i \in V, \\
& \quad x_p^t \in \{0, 1\} \quad \forall p \in P, \forall t \in T, \\
& \quad y_i \geq 0 \quad \forall i \in V, \\
& \quad u_o^t \in \{0, 1\} \quad \forall a \in A, \forall t \in T, \\
& \quad u_m^m \in \{0, 1\} \quad \forall a \in A, \forall m \in T, \\
& \quad u_d^t \in \{0, 1\} \quad \forall a \in A, \forall i \in V.
\end{align*}
\]

The objective (41) minimizes the total costs. The first set of constraints (42) say that a workday technician can have at most one assignment. The next four constraint sets enforce investment decision costs. Constraints (43) concern new technicians, (44) overtime for workday technicians, (45) skill upgrade and (46) digitization of tasks together with ensuring that each task is either assigned, marked as unassigned or digitized. In these constraints, we apply parameters taking on either value 0 or 1: \( \delta_{oa}^t \) indicates if assignment \( p \) uses overtime, \( \delta_{ot}^t \) if assignment \( p \) requires skill \( s \) and \( \delta_{dug}^i \) if assignment \( p \) assigns task \( i \). Bounds (47)–(51) define the domain of the variables.

The master problem is LP-relaxed, and the reduced cost derived. The dual variables are \( \pi_a^t \leq 0, t \in T \) for constraints (42), \( \pi_o^t \leq 0, t \in NT \) for constraints (43), \( \pi_i^t \leq 0, t \in T \) for constraints (44), \( \pi_m^m \leq 0, t \in T, s \in S : s \notin S(t) \) for constraints (45) and \( \pi_i^i \geq 0, i \in V \) for constraints (46). The reduced cost for an assignment \( p \) for a (not new) workday technician \( t \) is:

\[
c_p^t - \delta_{oa}^t \pi_a^t - \sum_{s \in S : s \notin S(t)} \delta_{si}^t \pi_i^t - \sum_{i \in V} \delta_{dug}^i \pi_{dug}^i \leq \pi_d^t.
\]

The reduced cost for an assignment \( p \) for a new workday technician \( t \) includes the dual of (43) and hence becomes

\[
c_p^t - \delta_{oa}^t \pi_a^t - \sum_{s \in S : s \notin S(t)} \delta_{si}^t \pi_i^t - \sum_{i \in V} \delta_{dug}^i \pi_{dug}^i \leq \pi_d^t + \pi_o^t.
\]

The right-hand sides of the reduced costs are constants when generating new assignments. Thus the same mathematical formulation solves the assignment problem for both existing and new workday technicians. The subproblem for a workday technician \( t \in T \) for master technician \( m \in T : \delta_m^t = 1 \) is:

\[
\begin{align*}
\min & \quad \sum_{i \in T} z_i - \pi_i^t u_{ot} - \sum_{s \in S} \pi_i^t u_{ot} - \sum_{i \in V} \pi_i^t x_i^t, \\
\text{s.t.} & \quad z_i \geq k \cdot c_d(x_i^j + x_i^j - 1) \quad \forall i, j \in V, \\
& \quad z_i \geq k \cdot c_d x_i^j \quad \forall i \in V, \\
& \quad \sum_{i \in V} f_i x_i^j \leq Q_t + w_{ot} \cdot u_{ot} - \sum_{i \in V} c_d x_i^j,
\end{align*}
\]

\[ (54) \]

\[ (55) \]

\[ (56) \]

\[ (57) \]
\[ x_i^t + x_j^t \leq 1 \quad \forall t \in T, \forall i, j \in V : \text{overlap}(i,j), \quad (58) \]

\[ x_i^t \leq u_i^m \quad \forall i \in V, \forall s \in S(i) : s \not\in S(t), \quad (59) \]

\[ x_i^t = 0 \quad \forall i \in V, tw(i, t) = \emptyset, \quad (60) \]

\[ u_{ir} \in \{0, 1\} \quad \forall a \in A, \forall t \in T, \quad (61) \]

\[ u_{im} \in \{0, 1\} \quad \forall a \in A, \forall m \in T. \quad (62) \]

The objective function (54) minimizes the reduced costs. The constraints are similar to the full investment decision problem (26)–(40), but for a single workday technician only, so it is not required that each task is handled, and new workday technician costs are handled implicitly.

The subproblem is solved by a MIP solver. The column generation procedure is initialized by a simple heuristic, which greedily assigns tasks to each technician. To reduce the run time of the solution approach, a limit is set on the number of column generation iterations. Furthermore, the number of columns generated in each iteration can be limited. Once terminated, the master problem solution may be fractional. An integer solution is obtained by applying a MIP solver to the non-LP-relaxed master problem with the generated columns.

6 | DATA SETS

The solution method is validated on instances from the literature [32] and on an instance from the Danish Telecom Infrastructure Company TDC Net. All instances are available at Zenodo [19].

6.1 | Data from the literature

The benchmark instances from the literature are proposed by Mathlouthi et al. [32]. They span a single day and solve a TRSP problem, where tools are taken into account. The instances thus contain information on available tools in each vehicle and on tool depots, where technicians can restock their vehicle. Also, the instances contain both travel times and travel distances. We interpret and modify the instances to match the problem considered in this article as follows:

- Tool requirements and tool depots are ignored.
- Travel distances are ignored because we use the travel times instead.
- Technicians work from 9 to 17.
- Tasks have multiple time windows in the original data. Instead we consider a task for each time window, so if the original data had 10 tasks with 3 time windows each, we have 30 tasks with a single time window.
- The penalty for not serving a task is assumed to be in the same unit as the travel time.

Without tool requirements and depots, groups of the Mathlouthi instances end up being identical instances. For this reason, we only consider the instances in the groups: AllSkills, nbrTech, ConfigurationDeBase, RedSkills, TpsRep10-20. In the first group, all technicians can serve all tasks. In the remaining groups this is not the case, and the skill distribution differs between the groups. Furthermore, the number of technicians differs in nbrTech, and task durations are shorter in TpsRep. Finally, each group contains instances with different time window sizes (narrow and wide) and different number of tasks and technicians. An overview of the instances is provided in Table 1. The total number of investigated Mathlouthi instances is 1185.

Because the instances only span a single day, the investment decisions will obviously not be realistic. The instance set is large and spans different scenarios and are thus useful to evaluate the quality of the ALNS and the task assignment approach, as well as the run times of the solution approaches.

6.2 | TDC Net data

The TDC Net instance consists of 720 workday technicians, 2677 tasks, 31 different skills, and a time horizon of 5 days. The instance is based on real-life data from the Danish Telecom Infrastructure Company TDC Net. The workday technicians work on a given day from 7:30 to 15:30. They hold a number of skills; about 10% hold less than 10 different skills, about 50% less than 15 skills, about 85% less than 20 skills, and the remaining 15% up to 27 skills. The least frequent skill required by a task
is obtained by 30 workday technicians, the most frequent skill by 645 workday technicians, and on average a required skill is obtained by 317 workday technicians.

The task time windows span from 30 min to all five days. About 25% of the tasks have a duration less than 30 min, about 33% have a duration between 30 and 60 min, about 30% have a duration between 60 and 120 min, and about 12% have a duration higher than 120 min. Each task requires exactly one skill. 18 of the skills are required by less than 25 tasks each, while the five most popular skills are required by 209 to 559 tasks each.

6.3 Investment decision data

To obtain data for the investment decision problem in Section 5, the TRSP instances from the literature and from TDC Net are extended with investment decision data. Recall that the objective function without investment decision is the sum of the total travel time and the penalty for unserved/unassigned tasks. To ensure that the objective function costs are comparable, we convert investment decision costs to travel time. The travel time is given in minutes. Assume that a van drives 10 km pr liter of fuel, and that the fuel costs €2 per liter. Also, assume that the average driving speed is 60 km/hour. This means that €1 equals 5 min of travel time.

Also assume that the monthly technician salary including overhead is €5000/month for the company. In round numbers, this gives an hourly cost of €30, which again corresponds to 150 min of travel time.

In the Mathlouthi instances, the investment decision data is:

- **Digitization**: the cost for digitizing a task is set to €500, which corresponds to 2500 min of travel time. We assume that every fifth task can be digitized.
- **Overtime**: we assume that overtime lasts for 120 min. Overtime is assumed to cost 50% extra, that is, the total cost for the company becomes €45 \times 2 = €90. This corresponds to 450 min of travel time.
- **New technician**: a technician works 8 h per day, so the workday technician cost is €240, which equals 1200 min of travel time. The new technicians consist of a duplicate of the existing technicians. For example, if an instance consists of two technicians with certain skills and locations, then it is possible to hire two new technicians with the same skills and locations as the existing ones.
- **Skill upgrade**: assume a total of 5 workdays for training, and that the training itself costs €2000. The total cost is then €30 \times 8 \times 5 + €2000 = €3200. We assume that the technician remains employed for another two years, giving a yearly cost of €1600. Assuming 219 workdays per year, the rounded daily cost is just €7. This corresponds to 35 min of travel time per day. We generate five different sets of skills based on k-means clustering, where the distance between two skills is the total number of workday technicians minus the number of workday technicians having both skills.

In the TDC Net instances, the investment decision data is:

- **Digitization**: the cost for digitizing a task is set to €100, which corresponds to 500 min of travel time. We assume that tasks requiring one of the skills “ADSL.I,” “ADSL.S,” “ISDN2.I,” “ISDN2.S,” “PSTN.I,” “PSTN.S,” “SHDSL.” can be digitized, which is a total of 450 tasks.
- **Overtime**: the same as for the Mathlouthi instances.
- **New technician**: the same as for the Mathlouthi instances.
- **Skill upgrade**: the cost is the same as for the Mathlouthi instances. The daily cost is multiplied by 5 (35 min $\cdot$ 5 = 175), because the instance spans 5 days. We generate 10 different sets of skills based on k-means clustering, where the distance between two skills is the total number of workday technicians minus the number of workday technicians having both skills.

7 | COMPUTATIONAL RESULTS

In the following, we denote the assignment approach solved by column generation as the ASSM, and we refer to columns in the ASSM as columns, assignments or paths.

The computational evaluation is twofold:

1. First, the ASSM is compared to the ALNS. The purpose is to qualify that the assignment approach is a fair approximation of the TRSP. The ALNS cannot handle investment decisions, hence the comparison is performed on instances without investment decision data.

2. Next, the ASSM investment decisions are evaluated:
   - the instances without investment decision data are solved with the ALNS, and the results are denoted *the ALNS solutions before investment decisions*.
   - investment decision data is added to the instances, and they are solved by the ASSM. From this, we both obtain information on actual investment decisions, and we obtain a solution for the approximated TRSP, that is, with the simplifications in the ASSM.
   - the instances without investment decision data are modified to include the ASSM investment decisions (e.g., by adding extra technicians or expanding their working hours to include overtime). They are then solved by the ALNS. We denote this *the ALNS solutions after investment decisions*.
   - the ALNS solutions before and after investment decisions are compared to evaluate if the OPEX savings exceed the investment decision costs.

The quality of the ALNS algorithm was assessed as part of the technical report [18] by benchmarking the ALNS against a full MIP formulation for the TRSP on small test instances. The results showed that ALNS solves the TRSP with an average gap from the optimal MIP solution far below 1%. Furthermore, the ALNS running times scale well.

The ALNS parameters were set as in [18]: the algorithm removes between 40% and 60% of tasks in the destroy methods (the number of tasks to remove is chosen at random) and tries to insert all unserved task in the repair methods. The record-to-record acceptance method is run with a start threshold of 0.0015 and the decay for calculating destroy and repair weights is set to 0.99. For the 1200 s runs, the ALNS is run with 6 retries of 200 s. For the 300 s runs, the ALNS is run with 3 retries of 100 s.

All computational evaluations are performed on a server with Intel Xeon e3-2660 CPU and 128 GB RAM. Gurobi 9.5 is used as solver. In the following tables a “-” indicates that no results are available in the particular category.

All instances and full results are available at Zenodo [19].

7.1 | Evaluation of the ASSM

The results from the ASSM are compared to the ALNS results. The comparison is performed on all instances without investment decision data. We compare the objective function values, the number of unserved tasks, path length, and number of technicians used. The costs in the objective functions are expected to differ; the ALNS optimizes the travel time, while the ASSM optimizes an estimated travel time. Still, it is relevant to observe how close the two values are.

Both the ASSM and the ALNS are given a time limit of 300 s. Also, we reduce the set of tasks, which can be assigned to a technician, by imposing an upper bound of 100 min in travel time from the technician depot to the task location. This speeds up both approaches, as fewer tasks need to be considered for each technician.

In the following, we first analyze the results on the Mathlouthi instances, then on the TDC Net instance.

7.1.1 | Results for the Mathlouthi instances

The ASSM parameters are set as follows for the Mathlouthi instances: The factor $k$ is set to 5. The number of column generation iterations is limited to 75, and in each iteration at most one path is generated per technician. The subproblem is given a 3 s time...
Table 2 shows the average gap between the ASSM and the ALNS solution values, according to the number of tasks resp. number of technicians. The gaps indicate that the ASSM finds smaller solution values than the ALNS. The gaps do not seem to increase with the size of the instances, but they are generally larger on the instances with narrow time windows. This pattern was also seen when comparing the solution values of ALNS and TRSP MIP: narrow time windows restrict the solution space, which benefits the MIP solver and challenges ALNS. This generally results in the ALNS finding larger solution values with larger gaps compared to the optimal ones. Recall that the ASSM both approximates the routing costs and relaxes the time windows constraints. The smaller solution values for the ASSM may be caused by an inaccurate routing cost approximation, but they may also stem from violation of the time windows of the technicians and/or the tasks. This seems to be the case for the instance (AllSkills, 10–25, number technicians 3, number jobs 30). From [18] we have an optimal MIP solution to the TRSP with solution value 59393.96, 11 unserved tasks and an average path length of 6.3333. The ALNS finds the same solution. The ASSM, however, finds a solution with value 56246.86, 10 unserved tasks and an average path length of 10.5. Clearly, this cannot be a feasible solution to the TRSP. Violating time windows is very likely the main reason that the ASSM generally finds better solutions for these instances than the ALNS.

Figure 1 illustrates the gap sizes. The absolute size of the gap is generally less than 10%, which we find reasonable as the ASSM is an approximation of the TRSP.

Table 3 reports the relative difference between number of unserved tasks, according to the number of tasks resp. number of technicians. The negative values mean that the ALNS on average has more unserved tasks than the ASSM. This is the main reason for the difference in solution values and is probably caused by the ASSM violating time windows. There is no clear pattern between problem instance size and the difference.

Table 4 contains the relative difference between the path lengths, according to the number of tasks resp. number of technicians. The ASSM generally finds longer paths than the ALNS. Probably because it generates less accurate schedules w.r.t. time windows and thus may produce infeasible routes.

Finally, Table 5 contains the relative difference between the number of used workday technicians, according to the number of tasks resp. number of technicians. The two methods agree on the number of used workday technicians to a large extent.

**Table 2** Average gap between solution values for the ALNS method and the ASSM, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | Narrow | Wide |
|----------------------|--------|------|
| # tasks 30           | −8.09  | −3.14|
| # tasks 45           | −12.26 | −4.00|
| # tasks 60           | −10.40 | −3.53|
| # tasks 75           | −8.82  | −3.51|
| # tasks 90           | −8.13  | −3.18|
| # tasks 105          | −9.64  | −4.28|
| # tasks 120          | −7.49  | −4.70|
| # tasks 135          | −6.96  | −5.09|
| # tasks 150          | −7.05  | −4.37|
| # tasks 210          | −2.10  | −2.01|
| # tasks 300          | −8.78  | −6.40|
| # tasks 600          | −6.67  | −4.58|

| Mathlouthi instances | Narrow | Wide |
|----------------------|--------|------|
| # techs 3 tech       | −7.92  | −3.66|
| # techs 4 tech       | −10.38 | −4.88|
| # techs 6 tech       | −8.01  | −5.19|
| # techs 12 tech      | −10.00 | −7.25|
| # techs 24 tech      | −6.48  | −3.61|

Note: Gap = (ASSM-ALNS)/ALNS. “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.

limit and the time limit for solving the master problem in each iteration is 30 s. At most 60 s can be spent on solving the integer master problem at the very end of the procedure.

The ALNS is tuned as explained in the beginning of Section 7 with 3 retries of 100 s.

Table 2 shows the average gap between the ASSM and the ALNS solution values, according to the number of tasks resp. number of technicians. The gaps indicate that the ASSM finds smaller solution values than the ALNS. The gaps do not seem to increase with the size of the instances, but they are generally larger on the instances with narrow time windows. This pattern was also seen when comparing the solution values of ALNS and TRSP MIP: narrow time windows restrict the solution space, which benefits the MIP solver and challenges ALNS. This generally results in the ALNS finding larger solution values with larger gaps compared to the optimal ones. Recall that the ASSM both approximates the routing costs and relaxes the time windows constraints. The smaller solution values for the ASSM may be caused by an inaccurate routing cost approximation, but they may also stem from violation of the time windows of the technicians and/or the tasks. This seems to be the case for the instance (AllSkills, 10–25, number technicians 3, number jobs 30). From [18] we have an optimal MIP solution to the TRSP with solution value 59393.96, 11 unserved tasks and an average path length of 6.3333. The ALNS finds the same solution. The ASSM, however, finds a solution with value 56246.86, 10 unserved tasks and an average path length of 10.5. Clearly, this cannot be a feasible solution to the TRSP. Violating time windows is very likely the main reason that the ASSM generally finds better solutions for these instances than the ALNS.

Figure 1 illustrates the gap sizes. The absolute size of the gap is generally less than 10%, which we find reasonable as the ASSM is an approximation of the TRSP.

Table 3 reports the relative difference between number of unserved tasks, according to the number of tasks resp. number of technicians. The negative values mean that the ALNS on average has more unserved tasks than the ASSM. This is the main reason for the difference in solution values and is probably caused by the ASSM violating time windows. There is no clear pattern between problem instance size and the difference.

Table 4 contains the relative difference between the path lengths, according to the number of tasks resp. number of technicians. The ASSM generally finds longer paths than the ALNS. Probably because it generates less accurate schedules w.r.t. time windows and thus may produce infeasible routes.

Finally, Table 5 contains the relative difference between the number of used workday technicians, according to the number of tasks resp. number of technicians. The two methods agree on the number of used workday technicians to a large extent.
FIGURE 1  Box and whisper plot of the solution value gap between the ASSM and the ALNS, applied to the Mathlouthi instances. The gap is computed as (ASSM-ALNS)/ALNS.

TABLE 3  Average relative difference in % in unserved tasks for the ALNS method and the ASSM, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | Narrow | Wide |
|----------------------|--------|------|
| # tasks              |        |      |
| 30                   | −10.80 | −4.17|
| 45                   | −13.81 | −5.81|
| 60                   | −9.25  | −4.30|
| 75                   | −9.13  | −5.76|
| 90                   | −10.21 | −4.41|
| 105                  | −11.65 | −6.56|
| 120                  | −9.96  | −6.89|
| 135                  | −9.36  | −7.60|
| 150                  | −7.85  | −5.77|
| 210                  | −2.43  | −2.21|
| 300                  | −5.48  | −3.80|
| 600                  | −6.50  | −3.55|

| Mathlouthi instances | Narrow | Wide |
|----------------------|--------|------|
| # techs              |        |      |
| 3                    | −10.09 | −5.96|
| 4                    | −9.23  | −4.14|
| 6                    | −5.05  | −3.42|
| 12                   | −6.13  | −4.18|
| 24                   | −9.70  | −5.72|

Note: Relative difference = (ASSM-ALNS)/ALNS. “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.
TABLE 4 Average relative difference in % in path lengths for the ALNS method and the ASSM, grouped by the number of tasks resp. technicians.

| Mathlouthi instances |            |            |
|----------------------|------------|------------|
| # tasks              | Narrow     | Wide       |
| 30                   | 22.11      | 21.62      |
| 45                   | 21.43      | 8.38       |
| 60                   | 17.51      | 8.71       |
| 75                   | 19.45      | 12.27      |
| 90                   | 24.63      | 10.80      |
| 105                  | 32.10      | 16.54      |
| 120                  | 31.32      | 20.40      |
| 135                  | 31.66      | 24.33      |
| 150                  | 28.58      | 19.65      |
| 210                  | 18.40      | 16.88      |
| 300                  | 16.18      | 10.73      |
| 600                  | 18.48      | 11.89      |

Mathlouthi instances

| # techs              | Narrow     | Wide       |
|----------------------|------------|------------|
| 3                    | 27.41      | 17.73      |
| 4                    | 18.72      | 10.23      |
| 6                    | 15.60      | 9.90       |
| 12                   | 17.36      | 11.57      |
| 24                   | 22.34      | 15.36      |

Note: Relative difference = (ASSM-ALNS)/ALNS. “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.

TABLE 5 Average relative difference in % in the number of used workday technicians for the ALNS method and the ASSM, grouped by the number of tasks resp. technicians.

| Mathlouthi instances |            |            |
|----------------------|------------|------------|
| # tasks              | Narrow     | Wide       |
| 30                   | −2.33      | −6.83      |
| 45                   | −0.67      | 4.00       |
| 60                   | 0.00       | 0.00       |
| 75                   | 0.00       | 0.00       |
| 90                   | 0.00       | 0.00       |
| 105                  | 0.00       | 0.00       |
| 120                  | 0.00       | 0.00       |
| 135                  | 0.00       | 0.00       |
| 150                  | 0.00       | 0.00       |
| 210                  | 0.00       | 0.00       |
| 300                  | 0.00       | 0.00       |
| 600                  | 0.00       | 0.00       |

Mathlouthi instances

| # techs              | Narrow     | Wide       |
|----------------------|------------|------------|
| 3                    | −0.38      | −0.17      |
| 4                    | 0.00       | −0.83      |
| 6                    | 0.00       | 0.00       |
| 12                   | 0.00       | 0.00       |
| 24                   | 0.00       | 0.00       |

Note: Relative difference = (ASSM-ALNS)/ALNS. “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.
All in all, the results indicate that the ASSM finds solutions which compare reasonably well to the ALNS, but which probably makes use of routes violating time window constraints of the full TRSP. This is seen in the lower solution values, and in that the ASSM finds longer paths for each workday technician and serves more tasks.

7.1.2 Results for the TDC Net instance

The TDC Net instance is significantly larger than the Mathlouthi instances, so the time limit is increased to 3 h. For both the ASSM and the ALNS, we keep a maximum travel time of 100 min from the depot to any task. In the ASSM, we also keep $k = 5$. In each column generation iteration, at most 500 columns can be generated. We set this limit to promote more column generation iterations before timing out, in hopes of achieving better dual variable values and thus better columns in the latter column generation iterations. As the number of technicians exceed the number of generated paths per iteration, we sort the technicians according to the decreased order of the right-hand side of the reduced cost in constraint (52) (or constraint (53) for new technicians). Preliminary tests revealed that if we try to generate a column for each technician in each column generation iteration, the instance times out early, and the number of generated columns explode.

We increase the time limit for solving the relaxed master problem in each column generation iteration to 2 min. We preserve a 3 s time limit for solving each subproblem, because the travel limit of 100 min from the depot limits the number of tasks to consider for each technician and thus also the subproblem size. Once the column generation procedure finishes, the integer problem is given a time limit of 20 min.

The ALNS is also given more time; a total of 2 restarts each with a time limit of 5400 s. Preliminary testing suggests that the ALNS needs more time to search the neighborhood, hence we have reduced the number of restarts to provide more time in each iteration. Furthermore, recall that the ALNS finishes by solving a set cover like model with up to 25 000 of the most promising generated paths. This will further increase the total run time.

All in all, this makes both the ASSM and the ALNS rather time consuming. In practice, this should not be a problem, as it is used for making strategic decisions and not operational plans. It is fair to assume that more time is available for deciding on strategic changes to the technician scheduling and routing setup.

Results for both approaches are summarized in Table 6. They show that the ASSM times out both in the column generation and when solving the final integer master problem. Still, the solution value of the ASSM is better than that of the ALNS, because it has fewer unserved tasks. This corresponds to what we saw for the Mathlouthi instances and is probably caused by the ASSM violating time windows. Also, the last ALNS improvements happen rather late, which suggests that the ALNS also times out before convergence.

If we subtract the penalty of unserved tasks from the objective, the travel time is significantly smaller for the ALNS. This indicates that the travel time estimate in the ASSM is inaccurate, and that time windows may be violated by the ASSM. However, looking at the total objective the two solution approaches only differ by 13%. Since both approaches time out, it is difficult to assess whether the difference stems from the different solution approaches and objectives or from lack of time.

A final note is that in real-life, TDC Net does not leave tasks unserved. Instead, overtime would be appointed to technicians, or technicians from a neighboring planning area would help with the final tasks. As a final resort, a task could be pushed to the next day and given very high priority.

| TDC Net instance | ASSM | ALNS |
|------------------|------|------|
| Objective        | 383 756 | 441 858 |
| Total time sec   | 11201.78 | 13478.58 |
| # unserved tasks | 16   | 42   |
| Total penalty    | 142 836 | 406 357 |
| Avg. path length | 5.86 | 4.70 |
| # paths in solution | 545 | 561 |
| Total number generated paths | 19 484 | 292 026 |
| Number cg. iters. | 38  |  -  |
| Optimal cg.      | False |  -  |
| Optimal int. master problem | False |  -  |
| Avg. number improvements | -  | 4.5  |
| Avg. number iters. |  -  | 1500 |
| Avg. last impr. iter. |  -  | 1336 |

Note: The last six lines concern results for the ASSM resp. the ALNS only.
7.2 Evaluation of investment decisions

The Mathlouthi and TDC Net instances are solved by the ASSM with investment decision data. The resulting investment decisions are included in the instances, and they are then resolved by the ALNS (to obtain ALNS solutions after investment decisions). This can then be compared to the ALNS solutions before investment decisions to analyze if the ASSM approach actually produces positive business cases and sane investment decisions. In the following, we first analyze results for the Mathlouthi instances, then for the TDC Net instance.

7.2.1 Mathlouthi instances

The algorithms in this section are run with same parameter settings as explained in Section 7.1.1. The Mathlouthi instances are first solved with the ASSM including investment decision data. Table 7 shows that many of the smaller instances and few of the larger instances are solved to MIP optimality by the column generation approach. The number of column generation iterations in Table 8 decreases as the instances grow larger, due to timing out. Finally, we consider the number of generated paths in Table 9. More paths are generated for the larger instances, but not significantly more, because the number of column generation iterations also decreases. The nonoptimal MIP solutions could mean that the investment decisions could be improved further.

Next, we consider the investment decisions made by the ASSM for the Mathlouthi instances, which are summarized in Tables 10–14. Generally, the number of investment decisions increases when the number of tasks and technicians increases. Investments in new technicians, see Table 10, are larger when time windows are narrow than when time windows are wide. Skill investments (Tables 11 and 12) are generally attractive in all instances. We have here included the subgroups of instances, as we see a clear pattern in skill investments. In the AllSkills instances, each technician already possesses all skills thus no skill investments are made. The highest number of skill investments are observed in the instance set RedSkills, where the skills per technician is even further reduced. Task digitization is also a popular investment, see Table 13, especially as the number of tasks increase. Investment in overtime, see Table 14, reveals that many technicians are assigned overtime. Still, investments are made in new technicians, which indicate that the workload is high relative to the number of technicians. This is also supported by the number of tasks being digitized.

| Mathlouthi instances | # tasks | Narrow | Wide |
|----------------------|---------|--------|------|
| 30                   | 96.00   | 96.00  |
| 45                   | 82.00   | 60.00  |
| 60                   | 76.00   | 32.00  |
| 75                   | 70.00   | 22.00  |
| 90                   | 70.00   | 4.00   |
| 105                  | 54.00   | 4.00   |
| 120                  | 34.00   | 4.00   |
| 135                  | 34.00   | 0.00   |
| 150                  | 18.89   | 1.11   |
| 210                  | 0.00    | 0.00   |
| 300                  | 0.00    | 0.00   |
| 600                  | 0.00    | 0.00   |

| Mathlouthi instances | # techs | Narrow | Wide |
|----------------------|---------|--------|------|
| 3                    | 61.28   | 24.87  |
| 4                    | 37.78   | 15.56  |
| 6                    | 4.00    | 2.00   |
| 12                   | 0.00    | 0.00   |
| 24                   | 0.00    | 0.00   |

Note: “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.
TABLE 8  Average number of column generation iterations before optimality or timeout, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | # tasks | Narrow | Wide |
|----------------------|---------|--------|------|
|                      | 30      | 32.00  | 44.24|
|                      | 45      | 38.44  | 58.12|
|                      | 60      | 42.08  | 67.34|
|                      | 75      | 48.04  | 71.40|
|                      | 90      | 54.68  | 74.76|
|                      | 105     | 62.24  | 74.80|
|                      | 120     | 66.30  | 74.48|
|                      | 135     | 69.38  | 75.00|
|                      | 150     | 72.49  | 74.87|
|                      | 210     | 75.00  | 75.00|
|                      | 300     | 24.04  | 31.80|
|                      | 600     | 2.87   | 3.95 |

| Mathlouthi instances | # techs | Narrow | Wide |
|----------------------|---------|--------|------|
|                      | 3       | 53.29  | 68.18|
|                      | 4       | 62.87  | 71.29|
|                      | 6       | 47.72  | 53.36|
|                      | 12      | 9.63   | 12.73|
|                      | 24      | 2.00   | 2.95 |

Note: “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.

TABLE 9  Average number of generated paths in column generation, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | # tasks | Narrow | Wide |
|----------------------|---------|--------|------|
|                      | 30      | 125.36 | 121.88|
|                      | 45      | 188.06 | 230.82|
|                      | 60      | 201.44 | 321.08|
|                      | 75      | 218.72 | 363.80|
|                      | 90      | 235.98 | 410.30|
|                      | 105     | 284.50 | 402.72|
|                      | 120     | 328.70 | 395.94|
|                      | 135     | 345.66 | 405.18|
|                      | 150     | 419.99 | 480.10|
|                      | 210     | 355.20 | 400.10|
|                      | 300     | 366.68 | 483.98|
|                      | 600     | 132.07 | 155.73|

| Mathlouthi instances | # techs | Narrow | Wide |
|----------------------|---------|--------|------|
|                      | 3       | 240.46 | 321.90|
|                      | 4       | 333.46 | 432.97|
|                      | 6       | 490.74 | 587.46|
|                      | 12      | 253.85 | 323.95|
|                      | 24      | 142.68 | 169.30|

Note: “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.
### TABLE 10
Average number of investments in new technicians, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | # tasks | Narrow | Wide |
|----------------------|---------|--------|------|
|                      | 30      | 1.10   | 0.28 |
|                      | 45      | 1.74   | 0.62 |
|                      | 60      | 2.76   | 1.40 |
|                      | 75      | 3.10   | 2.14 |
|                      | 90      | 3.20   | 2.78 |
|                      | 105     | 3.20   | 2.96 |
|                      | 120     | 3.20   | 3.08 |
|                      | 135     | 3.20   | 3.18 |
|                      | 150     | 3.76   | 3.49 |
|                      | 210     | 3.00   | 3.00 |
|                      | 300     | 8.40   | 7.88 |
|                      | 600     | 18.24  | 15.53|

Note: “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.

### TABLE 11
Average number of investments in skill upgrades, grouped by the number of tasks.

| Mathlouthi instances | Avg. number skill investments | 30 tasks | 45 tasks | 60 tasks | 75 tasks | 90 tasks | 105 tasks | 120 tasks | 135 tasks | 150 tasks | 210 tasks | 300 tasks | 600 tasks |
|----------------------|--------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Narrow               | 5.08                           | 7.22    | 8.84    | 11.10   | 12.44   | 13.22   | 15.16   | 15.60   | 17.06   | 15.30   | 39.22   | 81.76   |
| Wide                 | 4.06                           | 5.04    | 7.00    | 8.76    | 12.00   | 12.60   | 13.56   | 15.16   | 16.24   | 16.00   | 38.08   | 77.78   |
| AllSkills            | 0.00                           | 0.00    | 0.00    | 0.00    | 0.00    | 0.00    | 0.00    | 0.00    | 0.00    | 0.00    | -       | -       |
| ConfigurationDeBase  | 5.30                           | 7.55    | 9.45    | 12.20   | 14.25   | 14.65   | 15.75   | 16.50   | 17.88   | 15.65   | 38.65   | 79.88   |
| nbrTech              | 5.65                           | 8.05    | 10.80   | 12.85   | 16.10   | 17.70   | 20.65   | 22.75   | 22.30   | -       | -       | -       |
| RedSkills            | 7.30                           | 9.40    | 12.70   | 15.70   | 19.15   | 20.55   | 22.00   | 23.70   | 23.45   | -       | -       | -       |
| TpsRep10-20          | 4.60                           | 5.65    | 6.65    | 8.90    | 11.60   | 11.65   | 13.40   | 13.95   | 14.70   | -       | -       | -       |

Note: New technicians may also have their skills set upgraded, and that each technician may have multiple upgrades.

### TABLE 12
Average number of investments in skill upgrades, grouped by the number of technicians.

| Mathlouthi instances | Avg. number skill investments | 3 tech | 4 tech | 6 tech | 12 tech | 24 tech |
|----------------------|--------------------------------|--------|--------|--------|---------|---------|
| Narrow               | 10.83                          | 16.43  | 26.32  | 58.90  | 95.52   |         |
| Wide                 | 9.96                           | 13.98  | 25.44  | 57.23  | 92.15   |         |
| AllSkills            | 0.00                           | -      | -      | -      | -       |         |
| ConfigurationDeBase  | 13.35                          | -      | 25.88  | 58.06  | 94.02   |         |
| nbrTech              | -                              | 15.21  | -      | -      | -       |         |
| RedSkills            | 17.11                          | -      | -      | -      | -       |         |
| TpsRep10-20          | 10.12                          | -      | -      | -      | -       |         |

Note: New technicians may also have their skills set upgraded, and that each technician may have multiple upgrades.
### TABLE 13  Average number of investments in task digitization, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | Narrow | Wide |
|----------------------|--------|------|
| # tasks              |        |      |
| 30                   | 1.52   | 4.70 |
| 45                   | 2.24   | 6.32 |
| 60                   | 3.98   | 8.72 |
| 75                   | 5.64   | 11.06|
| 90                   | 8.52   | 14.04|
| 105                  | 11.04  | 17.12|
| 120                  | 14.26  | 20.48|
| 135                  | 19.70  | 25.52|
| 150                  | 24.40  | 28.56|
| 210                  | 63.70  | 61.50|
| 300                  | 47.32  | 57.74|
| 600                  | 84.69  | 109.33|

### TABLE 14  Average number of investments in overtime, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | Narrow | Wide |
|----------------------|--------|------|
| # techs              |        |      |
| 3                    | 13.22  | 17.64|
| 4                    | 8.28   | 14.48|
| 6                    | 42.16  | 49.16|
| 12                   | 73.35  | 86.03|
| 24                   | 56.48  | 88.10|

Note: “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.

### TABLE 15  Average number of investments in overtime, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | Narrow | Wide |
|----------------------|--------|------|
| # tasks              |        |      |
| 30                   | 2.70   | 1.78 |
| 45                   | 2.84   | 2.58 |
| 60                   | 3.72   | 3.30 |
| 75                   | 4.52   | 4.50 |
| 90                   | 4.78   | 5.04 |
| 105                  | 5.36   | 5.38 |
| 120                  | 5.48   | 5.42 |
| 135                  | 5.86   | 5.58 |
| 150                  | 6.57   | 6.50 |
| 210                  | 6.00   | 5.90 |
| 300                  | 15.02  | 15.38|
| 600                  | 30.20  | 30.63|

| Mathlouthi instances | Narrow | Wide |
|----------------------|--------|------|
| # techs              |        |      |
| 3                    | 4.45   | 4.27 |
| 4                    | 5.46   | 5.27 |
| 6                    | 10.92  | 10.70|
| 12                   | 21.83  | 22.35|
| 24                   | 35.16  | 37.45|

Note: New technicians may also be assigned overtime. “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.
Finally, we note that investment decisions are made in all four categories. A duration curve of the number of investment decisions is illustrated in Figure 2, and a duration curve of how many of the four investment decision types are utilized is illustrated in Figure 3. A duration curve depicts the values in descending order of magnitude. Together, this highlights the method’s capability to exploit synergies between the investment decisions instead of the more traditional approach, where typically only one type of investment decision is analyzed at a time.

The Mathlouthi instances are extended with the investment decisions made by the ASSM (e.g., by adding extra technicians or modifying time windows to include overtime) and then solved by the ALNS in order to obtain the ALNS solutions after investment decisions. The purpose is to analyze the consequence of the investment decisions in the full TRSP. The first comparison concerns the actual business case; does the OPEX saving exceed the CAPEX cost? The OPEX saving is calculated as the ALNS OPEX before investment decisions minus the ALNS OPEX after investment decisions. The ALNS OPEX
FIGURE 4  Duration curve of the business case value for the investments for the Mathlouthi instances. A duration curve depicts the values in descending order of magnitude. The business case is calculated as the savings in the ALNS before and after investment decisions, subtracted the investment costs: $\text{ALNS} (\text{before inv}) - \text{ALNS} (\text{after inv}) - \text{CAPEX}$. Note the logarithmic y axis. A positive business case value means that the savings in OPEX cost exceed the investment CAPEX cost.

TABLE 15  The average business case value, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | # tasks | Narrow       | Wide       |
|----------------------|---------|--------------|------------|
|                      | 30      | 39572.56     | 43036.02   |
|                      | 45      | 71932.06     | 65584.00   |
|                      | 60      | 121807.27    | 106535.19  |
|                      | 75      | 168483.44    | 156765.33  |
|                      | 90      | 217591.93    | 213806.93  |
|                      | 105     | 263747.35    | 263026.00  |
|                      | 120     | 300898.46    | 306920.51  |
|                      | 135     | 348778.73    | 355101.67  |
|                      | 150     | 371975.63    | 370678.97  |
|                      | 210     | 506711.24    | 501449.06  |
|                      | 300     | 667242.94    | 665625.99  |
|                      | 600     | 1314148.90   | 1259082.05 |

| Mathlouthi instances | # techs | Narrow       | Wide       |
|----------------------|---------|--------------|------------|
|                      | 3       | 227146.87    | 226447.56  |
|                      | 4       | 215208.96    | 212845.82  |
|                      | 6       | 548953.51    | 548819.16  |
|                      | 12      | 100010.01    | 973789.48  |
|                      | 24      | 1291207.76   | 1184787.57 |

Note: The business case is calculated as the savings in the ALNS before and after investment decisions, subtracted the investment costs: $\text{ALNS}(\text{before inv}) - \text{ALNS}(\text{after inv}) - \text{CAPEX}$. “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.

is the objective function in the TRSP (1). The CAPEX is the total cost of the investment decisions made by the ASSM. Figure 4 shows a duration curve for the business case values and the average business case values are found in Table 15. The business case for each instance is positive and ranges from approx. 15 000 to 1 500 000. Recall that the objective function value is given in travel time, see Section 6.3. Reverting to actual cost, the business case range is from approx. €3000 to approx. €300 000.
A duration curve of the OPEX for the Mathlouthi instances after investment decisions are shown in Figure 5. They span from approx. 6500 to approx. 950 000 which corresponds to approx. €1300 to €190 000. The average saving in OPEX is displayed in Table 16 spanning from 40 000 to approx. 1 575 000 which is €8000 to €3 150 000. The saving in OPEX is significant compared to the resulting OPEX.

The next investigation is the driving force behind the investments. We compare the results for the ALNS before and after investment decisions. The average difference in the number of unserved tasks is provided in Table 17, and a duration curve of the difference in travel time is shown in Figure 6. The results clearly show that reducing the number of unserved tasks drives
TABLE 17 The average number of unserved tasks by the ALNS before investments after investment decisions, grouped by the number of tasks resp. technicians.

| Mathlouthi instances | # tasks | Narrow | Wide |
|----------------------|---------|--------|------|
|                      | 30      | 8.32   | 9.42 |
|                      | 45      | 14.84  | 14.66|
|                      | 60      | 26.48  | 25.08|
|                      | 75      | 35.24  | 36.12|
|                      | 90      | 43.26  | 48.18|
|                      | 105     | 49.64  | 55.24|
|                      | 120     | 55.90  | 61.62|
|                      | 135     | 63.98  | 69.48|
|                      | 150     | 71.06  | 75.02|
|                      | 210     | 101.10 | 99.90|
|                      | 300     | 143.62 | 149.54|
|                      | 600     | 292.87 | 291.45|

| Mathlouthi instances | # techs | Narrow | Wide |
|----------------------|---------|--------|------|
|                      | 3       | 43.70  | 46.60|
|                      | 4       | 41.37  | 45.23|
|                      | 6       | 112.16 | 118.50|
|                      | 12      | 213.25 | 219.10|
|                      | 24      | 309.76 | 297.15|

Note: Gap = ALNS(before inv) − ALNS(after inv). “Narrow” and “Wide” refer to the size of the time windows in the Mathlouthi instances.

FIGURE 6 Duration curve of change in the travel time for the ALNS before and after investment decisions, calculated as ALNS (before inv) − ALNS (after inv). A duration curve depicts the values in descending order of magnitude.
investments. In fact, the travel time tends to increase after investment decisions are made. The reason is twofold; the penalty of unserved tasks dominates the objective function and is given higher priority than travelling. Also, serving more tasks very likely increases the travel time, because more traveling is necessary.

The OPEX savings suggest that the business cases are generally very strong. Of course, the instances only span a single day, which is not enough to make actual investment decisions. Still, the results strongly suggest that the OPEX for the Mathlouthi instances could be optimized significantly through strategic decisions.

7.2.2 TDC Net instance

Preliminary test runs reveal that adding investment decision data to the TDC Net instance makes the problem significantly harder to solve. We thus warmstart the ASSM with the solution without investment decision data from Section 7.1.2. Other than that, the ASSM and ALNS are run with the same parameter settings as in Section 7.1.2.

Results for the ASSM with investment decision data and the ALNS after investment decisions are summarized in Table 18. They show that the ASSM times out after 16 column generation iterations, and that solving the final integer master problem also times out. It generates 8545 paths in total. Recall that without investment decision data, the ASSM managed 38 iterations and generated 19 484 before timeout. This indicates that including investment decision data makes the problem harder to solve. The reason is twofold; the subproblem becomes more time consuming because of the investment decision variables; and the relaxed master problem also becomes more time consuming to solve mainly because of the investment decision variables. The ALNS also manages fewer iterations and generated paths, probably because the number of technicians has increased.

We observe a difference in the number of unserved paths and also a difference in the objective function value, just like we did for the TDC Net instance without investment decision data (ASSM)/before investment decisions (ALNS). If we subtract the penalty of unserved tasks from the objective, we again see that the travel times are quite different, with the ASSM having a large estimated travel time.

The actual investment decisions made by the ASSM are summarized in Table 19. The number of investments seems fair compared to the instance size. With 6 new technicians, the total number of workday technicians in the instance is 745 (5 days of 6 new technicians). 95 skill upgrades take place and 35 technicians must work overtime. Also, of the 2677 tasks, 7 are digitized.

The total investment costs sum up to 43 075, which is a small value compared to the objective function value. Using the ALNS results, we can quantify the business case for the TRSP as 441 858 − 183 977 − 43 075 = 214 806. The business case is clearly positive, which indicates attractive investment decisions. The TDC Net instance only spans 5 days, so more analyses would be needed to ensure the robustness of the investment decisions.

A closer look at the ALNS solutions before and after investment decisions shows that the largest decrease in the objective stems from fewer unserved tasks, going from 42 to 16 unserved tasks. Does this decrease justify employment of 6 new technicians? TDC Net is very focused on customer satisfaction and thus prioritizes task service very highly. Still, if the number of new technicians is too high, the penalty of unserved tasks should be lowered. The higher the penalty, the more investment decisions are made. It is, however, interesting to see that the amount of overtime does not explode, but instead the investments indicate that new technicians and skill upgrades are an attractive alternative to increase customer satisfaction.

| Comparison                   | ASSM       | ALNS       |
|------------------------------|------------|------------|
| Objective                    | 284 678    | 183 977    |
| Total time sec               | 11341.65   | 14039.03   |
| # unserved tasks             | 6          | 16         |
| Total penalty                | 47 668     | 154 164    |
| Avg. path length             | 6.14       | 4.67       |
| # paths in solution          | 560        | 568        |
| Total number generated paths | 8545       | 271 522    |
| Number cg. iters.            | 16         | -          |
| Optimal cg.                  | False      | -          |
| Optimal int. master problem  | False      | -          |
| Avg. number improvements     | -          | 7          |
| Avg. number iters.           | -          | 1200       |
| Avg. last impr. iter.        | -          | 932        |

Note: TDC Net instances. The last two sections concern results for the ASSM resp. the ALNS only.
TABLE 19  The actual investment decisions found by the ASSM.

| Investments          |         |
|----------------------|---------|
| # skill upgrades     | 95      |
| Skill upgr. cost     | 16 625  |
| # new techs          | 6       |
| # new daily techs    | 30      |
| New tech cost        | 7200    |
| # dig. tasks         | 7       |
| Dig. task cost       | 3500    |
| # overtime           | 35      |
| Overtime cost        | 15 750  |

Note: TDC Net instances.

Unlike for the Mathlouthi instances, we do see a significant decrease in the travel time in the ALNS after investment decisions. Before investment decisions the travel time was 35 501 (objective value minus penalty: 441 858 – 406 357) and after investment decisions 29 813 (183 977 – 154 164). This is a reduction of 5688 travel minutes or 16%. The saving in travel time corresponds to almost 95 h of driving in just 5 days. Relative to the total number of tasks, the reduction in the number of unserved tasks is very small, and therefore does not have a negative impact on the travel time, as was the case for the Mathlouthi instances.

Still, the decrease in travel time is smaller than the investment decision costs. The investment decisions are driven by unserved tasks, as the penalty dominates the objective function values. Had the focus been on investment decisions to reduce travel time, more emphasis should be put on improving the travel time estimate in the ASSM. Also, if the ASSM scaled better, we would probably be able to identify even better investment decisions. In the following section, we discuss future research directions concerning these aspects.

8 | FUTURE WORK

The investment decision master problem is very generic and could be applied to many other problems than the TRSP. Most obvious is to solve other routing problems, but generally any set cover like problem could be considered. In this section, however, we remain focused on the TRSP and possible extensions of the proposed solution approach.

8.1 | Investment decisions in the TRSP

If investment decisions are motivated by reducing the travel time and not the number of unserved tasks, then it would be beneficial to consider investment decisions in the original TRSP. That is, instead of approximating the TRSP with the task assignment approach, the full TRSP should be considered. This would eliminate the inaccuracies from estimating the travel time and the time window constraints in the assignment approach. The column generation approach could be preserved with the same master problem, and where the subproblem becomes the TRSP for a single technician. To limit the size of of the subproblem, an upper bound can be set on how far technicians can travel from their home depot. Still, the subproblem remains difficult to solve, and it is fair to assume that the approach would require significantly more time to find a solution of acceptable quality.

8.2 | Reducing the problem instance size

To obtain robust investment decisions, it is desirable to solve the investment decision problem for a larger time horizon, for example, a year. This, however, would lead to very large problem instances with many workday technicians (and subproblems), and it is fair to assume the column generation approach would require many iterations before the investment decisions converge.

The time horizon can be reduced by clustering similar time periods, as done for the CEP, see Section 3, and the work of Buchholz et al. [5]. Specifically, k-means clustering can be applied to cluster weeks according to task types, number of technicians and the total task duration. We suggest to cluster weeks instead of days, as many tasks have time windows spanning more than a day. If this is not the case in certain instances, then clustering days may be a better approach as it is probably possible to reduce the time horizon even more.

Another method to reduce the problem instance size is to reduce the geographical scope of the problem. Obviously, this is a heuristic approach, especially if the instance contains tasks or technicians close to the border between geographical areas.
Decomposing problems according to geography is a known method from the VRP research area. Santini et al. [46] propose and survey decomposition methods for heuristics solving the VRP. They argue that decomposing the VRP into independent subproblems is beneficial, as this allows for parallelism. In the literature decompositions are based on geography [23]; on vehicle capacity [46]; randomly [46]; using unsupervised machine learning such as k-means clustering [2] or historical related k-medoids clustering [37]; or by collapsing tasks into hyper nodes in a path based decomposition [46]. Santini et al. [46] evaluate the decomposition methods on instances with 600 customers, and they conclude that the unsupervised machine learning methods find best results. The TRSP is a rich VRP, so it is fair to assume that results from the VRP would also give attractive results for the TRSP.

8.3 | Investment decisions in new technicians

The location of new technicians can be optimized by setting the same travel time from their depot to every task. The estimated travel time of a resulting path would then be set by the travel time between two assigned tasks. This approach would, however, make the subproblem more difficult to solve, because all tasks are equal candidates to be assigned to the technician. This could be remedied by clustering tasks according to their geographical location and generate a new technician for each cluster with equal travel time to the tasks in that cluster.

8.4 | Limiting the investment decisions

For practical reasons it may not be possible to make too many changes to the technician staff. For instance, we cannot upgrade skills of all technicians at the same time due to capacity limits. Such constraints can easily be added to the model as budget constraints for each investment decision, or an overall investment decision pool budget.

8.5 | Portfolio of investment decisions

Future direction of research could also be to use the framework proposed in Buchholz et al. [6] to find alternative near-optimal investment choices. Given an initial solution to the investment decision problem, the investment decision problem is re-optimized, where the objective function is replaced by finding a solution with the most different investment decisions (given some predefined measurement), and where a constraint is added to ensure CAPEX + OPEX is no more than Δ higher than the initial investment decision solution value. In this way, the framework seeks to find a portfolio of near-optimal investment choices that are as different as possible. Having a plurality of near-optimal solutions makes it possible to include soft preferences, and it also uncovers sensitivity of the different choices.

9 | CONCLUSION

This article proposed a method for building scenarios for the TRSP. Given a set of tasks, technicians, skills, and working hours, the method is capable of investing in extra technicians, upgrading technician skills, extending working hours, and eliminating tasks by digitizing the corresponding task equipment.

In real-life, scenarios and strategic decisions are often based on what-if analyses, where single investment decisions are investigated one at a time. The proposed method excels in its holistic view and capability of investigating synergies between several investment decisions. To the best of our knowledge, such an approach has not been proposed previously to the TRSP, and similar approaches for the related VRP have a much narrower scope to, for example, optimizing the fleet. We thus believe that the proposed method helps close a research gap on strategic decisions in a routing context.

The proposed method consists of a matheuristic based on column generation. The master problem minimizes the total travel time, penalties of unserved tasks and investment decision costs. The subproblem assigns tasks to each technician subject to possible investment decisions. Instead of calculating the full routing problem for the technician, it rather approximates the travel time and the time window constraints. This reduces the solution time of the subproblem.

The article furthermore proposed an ALNS meta-heuristic for the TRSP to evaluate if the investment decisions found by the matheuristic are also beneficial to the full TRSP.

The proposed matheuristic was evaluated on benchmark instances from the literature and on a real-life test instance from the telecommunication company TDC Net. Every run of the matheuristic resulted in scenarios that also produced a positive business case in the ALNS. That is, the reduced TRSP costs exceeded the investment decision costs. A closer look at the results revealed that the investment decisions were driven by the goal of reducing the number of unserved tasks. The penalty of an
unserved task is typically very high and thus dominates the objective function. In the real-life test instance, travel time was also decreased with around 16%.

The proposed method of estimating travel time is reasonable in this context, but if the driving force of investment decisions is to reduce travel time, it would be more appropriate to adapt the matheuristic to solve the full TRSP. This would, however, be more time consuming.

ACKNOWLEDGMENTS

We thank TDC Net for providing data and useful information on the real-life technician routing and scheduling problem. We also thank Stefan Røpke for sharing his ALNS code. Finally, we thank the reviewers for their insightful comments which helped improve this paper. This work was funded by the Innovation Fund Denmark as part of the Grand Solution research project, GREENFORCE, project number 0224-00055B.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Zenodo Research at https://zenodo.org, reference number https://doi.org/10.5281/zenodo.7904015.

REFERENCES

[1] A. Ahmadi-Javid, P. Seyedi, and S. S. Syam, A survey of healthcare facility location, Comput. Oper. Res. 79 (2017), 223–263.
[2] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” Proc. 18th Annu. ACM-SIAM Symp. Discret. Algorithm, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2007, pp. 1027–1035.
[3] J. Bramel and D. Simchi-Levi, A location based heuristic for general routing problems, Oper. Res. 43 (1995), 649–660.
[4] C. Groër, B. Golden, and E. Wasil, A simulation-based approach for fleet design in a technician dispatch problem with stochastic demand, J. Oper. Res. Soc. 62 (2011), no. 8, 1510–1523.
[5] C. Groër, B. Golden, and E. Wasil, “A new algorithm for the site-dependent vehicle routing problem,” Advances in computational and stochastic optimization, logic programming, and heuristic search, Operations Research/Computer Science Interfaces Series, Vol 9, D. L. Woodruff (ed.), Springer, Boston, MA, 1998, pp. 301–312.
[6] X. Chen, B. W. Thomas, and M. Hewitt, The technician routing problem with experience-based service times, Omega 61 (2016), 49–61.
[7] C. E. Cortés, M. Gendreau, D. Leng, and A. Weintraub, A simulation-based approach for fleet design in a technician dispatch problem with stochastic demand, J. Oper. Res. Soc. 62 (2011), no. 8, 1510–1523.
[8] C. E. Cortés, S. Souyris, A. Weintraub, and M. Gendreau, Solving a technician dispatch problem using branch and price and constraint programming, Technical report working paper No. 14, Millennium Institute Complex Engineering Systems, 2010.
[9] D. Euchi, The vehicle routing problem with private fleet and multiple common carriers: Solution with hybrid metaheuristic algorithm, veh. Commun. 9 (2017), 97–108.
[10] R. Z. Farahani and M. Hekmatfar, Eds., Facility location: Concepts, models, algorithms and case studies, Springer, Berlin, 2009.
[11] M. Fischetti and D. Pisinger, Mathematical optimization and algorithms for offshore wind farm design: An overview, Bus. Inf. Syst. Eng. 61 (2019), 469–485.
[12] A. Flores-Quiroz, R. Palma-Behnke, G. Zakeri, and R. Moreno, A column generation approach for solving generation expansion planning problems with high renewable energy penetration, Electr. Power Syst. Res. 136 (2016), no. 20, 232–241.
[13] M. Gamst and D. Pisinger. Decision support the technician routing and scheduling problem. arXiv preprint arXiv: 2211.16968, 2022.
[14] M. Gamst and D. Pisinger. Data and results for the paper “Decision support for the technician routing and scheduling problem,” Technical report, Zenodo, 2023. https://doi.org/10.5281/zenodo.7904015.
[15] M. Gaudioso and G. Paletta, A heuristic for the periodic vehicle routing problem, Transp. Sci. 26 (1992), no. 2, 86–92.
[16] M. Gendreau, F. Guertin, J.-Y. Potvin, and E. Taillard, Parallel tabu search for real-time vehicle routing and dispatching, Transp. Sci. 33 (1999), no. 4, 381–390.
[17] B. Golden, A. Assad, L. Levy, and F. Ghysens, The fleet size and mix vehicle routing problem, Comput. Oper. Res. 1 (1984), no. 11, 49–66.
[18] C. Groër, B. Golden, and E. Wasil, A parallel algorithm for the vehicle routing problem, INFORMS J. Comput. 23 (2011), no. 2, 315–330.
[19] A. Gunawan, H. C. Lau, and P. Vansteenwegen, Orienteering problem: A survey of recent variants, solution approaches and applications, Eur. J. Oper. Res. 255 (2016), 315–332.
[20] N. Kolsaksili and M. Georgiadis, A multi-period, multi-regional generation expansion planning model incorporating unit commitment constraints, Appl. Energy 158 (2015), 310–331.
[21] J. E. Kottemann and W. E. Remus, The effect of planning horizon on the effectiveness of what-if analysis, Omega 20 (1992), no. 3, 295–301.
[22] S. Kumar and R. Panneerselvam, A survey on the vehicle routing problem and its variants, Intell. Inf. Manag. 4 (2012), no. 3, 66–74.
[23] R. Lahyani, L. C. Coelho, and J. Renaud, Alternative formulations and improved bounds for the multi-depot fleet size and mix vehicle routing problem, OR Spectr. 40 (2018), 125–157.
[24] G. Laporte, Y. Nobert, and D. Arpin, Optimal solutions to capacitated multidepot vehicle routing problems, Congr. Numer. 44 (1984), 283–292.
[30] F. Li, B. Golden, and E. Wasil, A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem, Comput. Oper. Res. 9 (2007), no. 34, 2734–2742.

[31] P. Masse and R. Gibrat, Application of linear programming to investments in the electric power industry, Manag. Sci. 3 (1957), 149–166.

[32] I. Mathlouthi, M. Gendreau, and J.-Y. Potvin, Mixed integer linear programming for a multi-attribute technician routing and scheduling problem, Inf. Syst. Oper. Res. 56 (2018), no. 1, 33–49.

[33] I. Mathlouthi, M. Gendreau, and J.-Y. Potvin, Branch-and-price for multi-attribute technician routing and scheduling problem, SN Oper. Res. Forum 2 (2021), no. 1, 1–35.

[34] I. Mathlouthi, M. Gendreau, and J.-Y. Potvin, A metaheuristic based on tabu search for solving a technician routing and scheduling problem, Comput. Oper. Res. 125 (2021), 105079.

[35] M. Mourgaya and F. Vanderbeck, Column generation based heuristic for tactical planning in multi-period vehicle routing, Eur. J. Oper. Res. 183 (2007), 1028–1041.

[36] V. Oree, S. Z. Sayed Hassen, and P. Fleming, Generation expansion planning optimisation with renewable energy integration: A review, Renew. Sust. Energ. Rev. 69 (2017), 790–803.

[37] H.-S. Park and C.-H. Jun, A simple and fast algorithm for k-medoids clustering, Expert Syst. Appl. 36 (2009), no. 2, 3336–3341.

[38] D. L. Pereira, J. C. Alves, and M. C. de Oliveira Moreira, A multiperiod workforce scheduling and routing problem with dependent tasks, Comput. Oper. Res. 118 (2020), 104930.

[39] V. Pillac, C. Guéret, and A. Medaglia, On the dynamic technician routing and scheduling problem (Proc. 9th Metaheuristics Conf. (MIC 2011)), 2011.

[40] V. Pillac, C. Guéret, and A. Medaglia, A parallel matheuristic for the technician routing and scheduling problem, Optim. Lett. 7 (2013), no. 7, 1525–1535.

[41] M. S. Rasmussen, T. Justesen, A. Dohn, and J. Larsen, The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies, Eur. J. Oper. Res. 219 (2012), no. 3, 598–610.

[42] S. Ropke and D. Pisinger, An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, Transp. Sci. 40 (2006), no. 4, 455–472.

[43] S. Salhi, A. Imran, and N. A. Wassan, The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation, Comput. Oper. Res. 52 (2014), 315–325.

[44] S. Salhi and M. Sari, A multi-level composite heuristic for the multi-depot vehicle fleet mix problem, Eur. J. Oper. Res. 103 (1997), 95–112.

[45] A. Santini, S. Ropke, and L. M. Hvattum, A comparison of acceptance criteria for the adaptive large neighbourhood search metaheuristic, J. Heurist. 24 (2018), no. 5, 783–815.

[46] A. Santini, M. Schneider, T. Vidal, and D. Vigo, Decomposition strategies for vehicle routing heuristics, INFORMS J. Comput. 35 (2023), 519–709.

[47] A. Schwele, J. Kazempour, and P. Pinson, Do unit commitment constraints affect generation expansion planning? A scalable stochastic model, Energy Syst. 11 (2020), 247–282.

[48] E. Tsang and C. Voudouris, Fast local search and guided local search and their application to British telecom’s workforce scheduling problem, Oper. Res. Lett. 20 (1997), 119–127.

[49] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, A unified solution framework for multi-attribute vehicle routing problems, Eur. J. Oper. Res. 234 (2014), no. 3, 658–673.

How to cite this article: M. Gamst and D. Pisinger, Decision support for the technician routing and scheduling problem, Networks. 83 (2024), 169–196. https://doi.org/10.1002/net.22188