The enriched median routing problem and its usefulness in practice

Huizing, Dylan; van der Mei, Rob; Schäfer, Guido; Bhulai, Sandjai

published in
Computers and Industrial Engineering
2022

DOI (link to publisher)
10.1016/j.cie.2022.108063

document version
Publisher's PDF, also known as Version of record

document license
Article 25fa Dutch Copyright Act

Computers & Industrial Engineering 168 (2022) 108063

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

The enriched median routing problem and its usefulness in practice

Dylan Huizing[a,*], Rob van der Mei[b,c], Guido Schäfer[a,d], Sandjai Bhulai[c]

[a] Networks and Optimization Group, Centrum Wiskunde & Informatica, Amsterdam, P.O. Box 94079 1090 GB, the Netherlands
[b] Stochastics Group, Centrum Wiskunde & Informatica, Amsterdam, P.O. Box 94079 1090 GB, the Netherlands
[c] Faculty of Science, Vrije Universiteit Amsterdam, 1081 HV Amsterdam, the Netherlands
[d] Institute for Logic, Language and Computation, Universiteit van Amsterdam, 1098 XG Amsterdam, the Netherlands

ARTICLE INFO

Keywords:
Location
Routing
Emergency logistics
Combined planning
Simulation study

ABSTRACT

Emergency response fleets often have to simultaneously perform two types of tasks: (1) *urgent* tasks requiring immediate action, and (2) *non-urgent* preventive maintenance tasks that can be scheduled upfront. In Huizing et al. (2020), Huizing et al. proposed the Median Routing Problem (MRP) to optimally schedule agents to a given set of non-urgent tasks, such that the response time for urgent tasks remains minimal. They proposed both an exact MILP-solution and a fast, scalable and accurate heuristic. However, when implementing the MRP-solution in a real-life pilot with a Dutch railway provider, we found that the model needed to be extended by including additional practical objectives and constraints. Therefore, in this paper, we extend the MRP to the so-called Enriched Median Routing Problem (E-MRP), making the model much better aligned with considerations from practice. Accordingly, we extend the MRP-based solutions to the E-MRP. This allows us to compare the performance of our proposed E-MRP solutions to performance obtained in the current operational practice of our partnering railway infrastructure company. We conclude that the E-MRP solution leads to a strong reduction in emergency response times compared to current practice by smartly scheduling the same volumes of non-urgent tasks.

1. Introduction

In many emergency response organizations, there are many useful things agents could do when there is no active emergency. For instance, in the railway industry, railway emergency responders have different routine inspections they need to perform in order to prevent incidents from even happening. If planned well, the agents can do these tasks (e.g., with given frequencies, locations, and time windows) while remaining well-spread over the network so that when emergencies do occur, they can respond quickly. This way, proper coordination of non-emergency tasks (over time and space) over multiple agents can help to reduce emergency response times and at the same time increase the efficiency of non-urgent tasks. To enable this coordination, Huizing, Schäfer, van der Mei, and Bhulai (2020) proposed the Median Routing Problem (MRP) and a fast, scalable yet accurate heuristic for optimally planning non-urgent tasks, or "jobs", while staying responsive to incidents requiring immediate action. Throughout, this heuristic is called the *mediate-divide-sequence-agree* (MDSA) method.

The MRP model and proposed solution methods provide new fundamental insight into how to balance the trade-off between urgent and non-urgent tasks properly. However, the application of the results in a real-life pilot setting taught us that the model assumptions were too limiting to be of real use in practice, and that many more limitations that occur in practice should be taken into consideration. For example, in terms of objectives, low emergency response times may be the most important goal, but the distance traveled outside of emergencies is another relevant goal. Also, in terms of constraints, the details of what makes a planning feasible can be numerous: some agents may not be authorized to perform certain tasks; some tasks may have specific time windows; some agents may have to stay close to the base station at all times; some jobs may need several agents to perform; some agents may have scheduled appointments within their shifts, during which they are unable to perform jobs or respond to emergencies. These are just some of many examples of limitations encountered in practice that are not covered by the MRP-model.

Motivated by this, the contribution of this paper is fourfold. First, on the basis of extensive feedback from our pilot study, we enrich the MRP-model with the inclusion of fifteen extensions to cover the additional objectives and constraints that practical applications may require. Second, we adapt the solution methodology to provide a solution for the E-

* Corresponding author.
E-mail addresses: dylan.huizing@cwi.nl (D. Huizing), r.d.van.der.mei@cwi.nl (R. van der Mei), g.schaefer@cwi.nl (G. Schäfer), s.bhulai@vu.nl (S. Bhulai).

https://doi.org/10.1016/j.cie.2022.108063
Received 26 July 2021; Received in revised form 19 January 2022; Accepted 3 March 2022
Available online 10 March 2022

MRP, demonstrating the flexibility of the methodology. Third, based on extensive discussions with planners of our partnering railway provider, we propose a Current Practice (CP) model that accurately describes the way in which planners currently schedule tasks (today, they do so without any support from MRP). Fourth, we perform extensive simulation experiments to compare the performance of planning in the CP-model with the performance of E-MRP model in order to assess the gain that can be obtained by using E-MRP. We do so by taking real-life case study data from our partnering railway provider, modeling how they would currently plan with the given data, and comparing that with the improved planning from our heuristic. The results show that the E-MRP solution strongly improves the responsiveness of the current system, even with an increased number of non-urgent tasks. This leads to the conclusion that our the E-MRP model provides a powerful means to balance urgent and non-urgent tasks, and is applicable in practice.

The remainder of this paper is structured as follows. In Section 2, we review the related literature, including generalizations in related problems. In Section 3, we describe the E-MRP model and solution algorithm. In Section 4, we describe the CP-model that describes the way non-urgent task planning is done in current operational practice. In Section 5, we elaborate on the simulation environment in which we make comparisons. In Section 6, we discuss the numerical results of our comparison. In Section 7, we present our conclusions and recommendations. For reference, the MRP is outlined in Appendix A.

2. Related literature

Our research builds directly on the work by Huizing et al. on the MRP (Huizing et al., 2020) and the proposed MDSA-heuristic for solving the MRP. This problem has similarities to both the p-Median Problem and the Distance-Constrained Vehicle Routing Problem.

In the Multi-Period Median Routing Problem, formulated by Kraster (2020), the jobs have to be divided over multiple shifts. They proposed doing so with a constructive Median Heuristic. In this heuristic, a compatibility between any two jobs is determined by scheduling one, computing which positions give good coverage when that job is being performed, then seeing if the second job is near one of those reactive positions.

The p-Median Problem is well-studied. In it, we must choose p representatives or 'medians' from a finite set of nodes, such that the summed distance of each unpicked node to its nearest median is minimal. Early research into the p-Median Problem includes the work of ReVelle and Swain (1970). More recently, Daskin and Maass have reviewed the known algorithms and results (Daskin & Maass, 2015). While NP-hard, the p-Medians can be approximated within a constant factor by at least the following three methods. Firstly, Charikar, Guha, Tardos, and Shmoys (2002) achieve a factor of $6\frac{2}{3}$ by means of LP-rounding. Secondly, Jain and Vazirani (2001) achieve a factor $6+\varepsilon$ by applying Lagrangean relaxation on the number of medians, and applying a primal-dual 3-approximation algorithm on the resulting Uncapacitated Facility Location Problem. Thirdly, Arya et al. achieve a factor $3+2/k$ by local search with k simultaneous swaps.

The classical p-Median Problem has been generalized in a number of ways. In the Multi-Capacitated Location Problem by el El Amrani, Benadada, and Gendron (2016), we must choose to which 'degree' each facility is opened. Rather than being allowed p medians, the lower degrees take up less of a common facility budget, but they also have less capacity with which to serve clients. They propose a 'Greatest Customer Demand First' heuristic boosted with an initial Branch-and-Cut solution. In the Directional p-Median Problem by Jackson, Rouskas, and Stallmann (2007), the points lie in a k-dimensional space, and medians can only cover other nodes if they are in the positive direction of the first l dimensions. They propose a polynomial-time algorithm for the 1-dimensional case, which they use as a subroutine in their heuristic for higher dimensions. In the p-Median Problem with Distance Selection by Benati and García (2014), the nodes also lie in k-dimensional space, but we select which q dimensions we care about, as well as the p corresponding medians. This is motivated by clustering on statistical data, where the interesting features are the ones that allow for meaningful clustering. This non-linear problem is linearized in different ways, and the radius formulation performs best. In the Hamiltonian p-Median Problem, we divide the nodes in p directed cycles, rather than p stars pointing at a median. Bektaş, Gouveia, and Santos (2019) solve this problem at competitive speed by combining subtour elimination constraints from the traveling salesman problem with path elimination constraints from location-routing problems and the concept of an 'acting depot'. In the Bi-Criteria p-Median p-Dispersion Problem, we not only minimize the summed distances of nodes to their medians, but we also maximize the smallest distance between any two medians. Colmenar, Hoff, Martí, and Duarte (2018) propose a Scatter Search matheuristic to find solutions on or near the Pareto front of these two objectives.

Even more thoroughly studied than the p-Median Problem, is the Vehicle Routing Problem (VRP). Recent overviews include those of Eksioglu, Vural, and Reisman (2009), Toth and Vigo (2014) and Joubert (2007). A particularly relevant variant is the Dynamic Vehicle Routing Problem, where some clients are revealed mid-operation. This problem is often tackled with either Multiple Scenario Approaches (Pillac, Guéret, & Medaglia, 2012), a priori routes (van Ee & Sitters, 2014; Zhang, Ohlmann, & Thomas, 2014) rolling horizon approaches (Jaillet, Bard, Huang, & Dror, 2002; Palma-Behnke et al., 2013), or roll-out policies (Goodson, Ohlmann, & Thomas, 2013).

Many variants of the basic VRP have been studied. VRP-REP lists 50 (VRP-REP, 2021). In the VRP with Time Windows (Solomon, 1984), certain clients can only be visited within contiguous time windows. In Orienteering Problems (Tsiligirides, 1984), we have bounded time to collects rewards from visited clients, instead of minimizing the distance to visit all of them. In the Pollution Routing Problem (Bektaş & Laporte, 2011), we may choose to drive more slowly to save fuel, as long as we abide by the time windows. In Distance-Constrained VRPs (Laporte, 1992), routes cannot exceed a certain length. In Green VRPs (Erdougan & Miller-Hooks, 2012), alternative fuel vehicles must visit a specialized refueling station periodically. In the Carrier-Vehicle Travelling Salesman Problem (Garone, Naldi, Casavola, & Frazzoli, 2010), clients are visited by a vehicle that must stay close to a mobile, but slow, carrier. In Two-Echelon VRPs (Crainic, Perboli, Mancini, & Tadei, 2010), goods are first brought to satellite stations, and from there to nearby destinations. In Dial-a-Ride Problems (Psaraftis, 1980) and VRPs with Pickup and Delivery (Savelsbergh & Sol, 1995), goods or persons must be collected from a pickup node and brought to a delivery node. In Consistent VRPs (Groër, Golden, & Wasil, 2009), it is important that a client who is visited in multiple time periods is visited as much as possible by the same vehicle and around the same time. In Periodic VRPs (Gaudioso & Paletta, 1992), some clients must be visited multiple times, but certain combinations of visiting days are not allowed. In Location-Routing Problems (Drexl & Schneider, 2015), we must simultaneously decide where to open depots and how to route over all clients from those depots. Perhaps most importantly, in the Technician Routing and Scheduling Problem (Pillac, Gueret, & Medaglia, 2013), the clients must be serviced by a technician with the right skills, tools and spare parts within the right time window. Pillac et al. described a successful meta-heuristic for this problem, combining a Regret-based constructive heuristic, an Adaptive Large Neighbourhood Search and post-processing by a set-covering-based binary program.

Other problems that combine planned jobs with emergency response are the following. Ichoua, Gendreau, and Potvin (2000) minimize both traveled distance and lateness to jobs that appear dynamically, using Tabu Search. Bertsimas and Van Ryzin (1993) propose policies with bounded costs for a dynamic Travelling Repairman Problem, where agents may move freely over the plane to service requests that appear. van den Berg and Van Essen (2019) studied from which hospitals to temporarily expend ambulances for planned transportation, such that the detriment to emergency response times is minimal. Independently, Kergosien, Gendreau, Ruiz, and Soriano (2014) have studied this as well. Kiechle, Doerner, Gendreau, and Hartl (2009) take a more integrated approach, where the emergency ambulance fleet and the planned transport fleet are the same. They investigate whether it is better for response time if vehicles wait at the location of the job they just finished, or the location of the job they are about to start. When looking to 'cover' an area by patrolling it, rather than standing by for emergencies, the unmanned vehicle literature proposes several methods for coordination (Agarwal, Hiot, Joo, & Nghia, 2007; Doitsidis et al., 2012; Shu, Wang, Lin, Liu, & Zhou, 2013; Wang & Hussein, 2007).

In maintenance-routing literature, we typically decide when to perform maintenance where and how to route between those maintenance activities. Most literature focuses on how to execute a transportation plan with vehicles that have to frequently 'pit stop' at a maintenance location (Başdere & Bilge, 2014; Gopalan & Talluri, 1998; Maróti & Kroon, 2005; Haouari, Shao, & Sherali, 2012; Penicka, Strupchanska, & Bjørner, 2003; Sarac, Batta, & Rump, 2006; Talluri, 1998), sometimes including crew scheduling (Cohn & Barnhart, 2003). In contrast, López-Santana, Akhavan-Tabatabaei, Dieulle, Labadie, and Medaglia (2016) determine when and how often to perform maintenance, balancing known repair costs against stochastic breakdown costs, and route repairmen over these jobs. They do so by iterating between continuous non-linear optimization and mixed-integer linear programming. Fontecha et al. (2019) made this method more scalable by using a matheuristic instead of iterating. In the field of offshore wind farms, Irawan, Ouelhadj, Jones, Stålhane, and Sperstad (2017) study a maintenance-routing problem similar to a VRP with Pick-up and Delivery, and use a Dantzig-Wolfe decomposition, which is typical for the latter.

3. Enriched median routing problem

We will now describe the central problem of this paper. In Section 3.1, we describe the E-MRP and its notation. In Section 3.2, we give a formulation of the Mixed Integer Linear Program (MILP) for the E-MRP, and in Section 3.3 we propose a fast and scalable heuristic for the E-MRP.

3.1. Problem description

The E-MRP is an extended version of MRP, with additional constraints and objectives. As in MRP, we are given a network and a set of agents, non-urgent tasks (or "jobs") and discrete time steps. Per time step, agents may hop to an adjacent node or stay where they are. The goal is to decide for each job who will perform it when, and for each agent where they should be throughout the discrete-time horizon. A more complete description of MRP, including a Mixed Integer Linear Programming description, is given in Appendix A. We expand on MRP with the following extensive list of features:

1. The planned travel time is added to the objective function;
2. The makespan is added to the objective function;
3. Penalties for assigning certain jobs to certain agents are added to the objective function;
4. The start and end locations of agents are variable;
5. Agents start and end at heterogeneous times;
6. Some jobs may not be started at certain times;
7. Jobs can require more than one agent;
8. Jobs can require some or all of its agents to have certain qualifications;
9. Some agents are not available for emergency response during a part of their shift;
10. Some agents are not available for processing jobs during a part of their shift;
11. Agents have personal sub-networks they cannot leave;
12. There are mandatory appointments for certain agents to be at a place at a certain time;
13. Jobs may end at a different location than where they start;
14. Aside from preferences, some assignments of jobs to agents are given as hard constraints;
15. Emergency probabilities and response times are time-dependent.

For the complete definition of E-MRP, denote A as the set of agents, J the set of jobs, V the nodes of the network, $V(v)$ the nodes adjacent or

Table 1
Notation for the Enriched Median Routing Problem.

Set		Description
A		The set of agents
J		The set of jobs
V		The set of nodes
T		The set of time steps, $T = \{0, 1, \dots, \overline{T}\}$
H		The set of shifts in the planning horizon
$A(h)$		The agents in shift $h \in H, A(h) \subseteq A$
$T(h)$		The time steps belonging to shift $h \in H, T(h) \subseteq T$
$T(a)$		The time steps in which agent $a \in A$ is active
$V_P(t)$		At time $t \in T$, the nodes where incidents may occur ($V_P(t) \subseteq V$)
$V(u)$		The nodes within one hop distance of $u \in V$, including u
B		The set of authorizations agents can have
$X^!$		Appointments (a, v, t) that agent a must be at node v at time t
Parameter	**Domain**	**Description**
$C^!_{uvt}$	$\mathbb{Q}_{\geqslant 0}$	The emergency response time at time $t \in T$ from $u \in V$ to $v \in V_P(t)$
$C^{\rightarrow}_{uv}$	$\mathbb{Q}_{\geqslant 0}$	The non-emergency travel time from $u \in V$ to $v \in V(u)$
P_{vt}	$(0, 1]$	Probability that the next emergency is at node $v \in V_P$, time $t \in T$
$Y^{\checkmark}_{at}$	$\{0, 1\}$	Whether agent $a \in A$ is available for emergencies at time $t \in T$
$Z^{\checkmark}_{at}$	$\{0, 1\}$	Whether agent $a \in A$ is available for non-urgent jobs at time $t \in T$
$V^{\triangleright}_{at}$	$\{0, 1\}$	Whether agent $a \in A$ can start their shift at node $v \in V$
$V^{\square}_{at}$	$\{0, 1\}$	Whether agent $a \in A$ can end their shift at node $v \in V$
V^{n}_{a}	$\{0, 1\}$	The current default start and end location of agent $a \in A$
$X^{\checkmark}_{av}$	$\{0, 1\}$	Whether agent $a \in A$ is allowed to visit node $v \in V$
$B^{\checkmark}_{ab}$	$\{0, 1\}$	Whether agent $a \in A$ has authorization $b \in B$
$L^{\triangleright}_{j}$	V	The start location of job $j \in J$
$L^{\square}_{j}$	V	The end location of job $j \in J$
R_{jt}	$\{0, 1\}$	Whether job $j \in J$ may be started at time step $t \in T$
Q_j	$\mathbb{Z}_{\geqslant 0}$	The number of time steps job $j \in J$ takes
$C^{\times}_{j}$	$\mathbb{Q}_{\geqslant 0}$	The penalty for not planning job $j \in J$
M_{jb}	$\mathbb{Z}_{\geqslant 0}$	How many agents with authorization $b \in B$ are needed for job $j \in J$
N_{aj}	$\mathbb{Q}_{\geqslant 0}$	The penalty incurred when assigning job $j \in J$ to agent $a \in A$
$Z^!_{aj}$	$\{0, 1\}$	Whether it is mandatory that agent $a \in A$ is assigned to job $j \in J$
$\phi_{response}$	$\mathbb{Q}_{\geqslant 0}$	The weight of the response time objective
$\phi_{distance}$	$\mathbb{Q}_{\geqslant 0}$	The weight of the distance objective
$\phi_{preference}$	$\mathbb{Q}_{\geqslant 0}$	The weight of the assignment preference objective
$\phi_{makespan}$	$\mathbb{Q}_{\geqslant 0}$	The weight of the makespan objective
$\phi_{ignoring}$	$\mathbb{Q}_{\geqslant 0}$	The weight of the job ignoring penalty objective
Variable	**Domain**	**Description**
x_{avt}	$\{0, 1\}$	Whether agent $a \in A$ is at $v \in V$ at time $t \in T$
f_{at}	$\mathbb{Q}_{\geqslant 0}$	The distance traveled by agent $a \in A$ between times $t \in T$ and $t + 1$
y_{uvt}	$\{0, 1\}$	Whether a potential emergency at $v \in V_P$, time $t \in T$ will be responded to from $u \in V$
z_{ajt}	$\{0, 1\}$	Whether agent $a \in A$ starts job $j \in J$ at time $t \in T$
z'_{aj}	$\{0, 1\}$	Whether agent $a \in A$ performs job $j \in J$
z''_{jt}	$\{0, 1\}$	Whether job $j \in J$ is started at time $t \in T$
z'''_{j}	$\{0, 1\}$	Whether job $j \in J$ is done at all
$\overline{z}$	$[0, \overline{T}]$	The latest completion time among jobs

equal to $v \in V$, and T the set of discrete time steps. Agents may have heterogeneous working hours, denote $T(a) \subseteq T$ the working hours of agent $a \in A$. There is a set of job authorizations B an agent can have, and we denote $B^{✓}_{ab} = 1$ if agent $a \in A$ has authorization $b \in B$, and 0 otherwise ***Table 1.

We denote $V^{\triangleright}_{av} = 1$ if node $v \in V$ is an allowed start location for agent $a \in A$, and 0 otherwise. Likewise, $V^{\square}_{av}$ indicates whether a can end their shift at v. The start time and end time of $a \in A$ are the earliest and latest time steps respectively in $T(a)$. In each time step, each active agent $a \in A$ may stay where they are or move to an adjacent node. However, they have personal sub-networks in which they must stay, and $X^{✓}_{av}$ equals 1 if a is allowed to visit v and 0 otherwise.

Each job must be performed. Once a job $j \in J$ is started, the agents assigned to it must stay at the location of the job for the entire duration Q_j. They must also be available for processing jobs throughout, and we indicate whether or not an agent $a \in A$ is available for job processing at time $t \in T$ by setting $Z^{✓}_{at}$ equal to 1 or 0. Unlike in MRP, the start location $L^{\triangleright}_j \in V$ may be different from the end location $L^{\square}_j \in V$, as some jobs may consist of thoroughly inspecting an 'edge' of the network. For ease of notation, we assume that jobs do not share locations, as we can easily introduce virtual locations. Some jobs have time constraints, meaning j can only be started at time $t \in T$ if R_{jt} equals 1 rather than 0. Note that these 'time windows' are not contiguous per se: it may make sense for certain jobs to be done during the morning peak hour or the afternoon peak hour, but not somewhere in between.

The most impacting difference between MRP and E-MRP, is that a job may require several agents. In fact, $j \in J$ may require that M_{jb} agent with authorization $b \in B$ are assigned to it. We allow an agent with multiple authorizations to count towards the requirement M_{jb} of each of those authorizations: for instance, if a job requires two agents with basic training ($M_{j,basic} = 2$) and one agent with mechanical training ($M_{j,mechanic} = 1$), the job can be fulfilled by two agents, if they both had basic training and at least one of them had mechanical training.

The decision variables are the following. We must decide for each agent $a \in A$ and each time step $t \in T(a)$ where in the network they are, by setting $x_{avt} = 1$ if a is at node $v \in V$ at time t and 0 otherwise. We must also set z'_{aj} to 1 or 0 to indicate whether agent $a \in A$ is assigned to job $j \in J$, and z''_{jt} to 1 or 0 to indicate whether j is initiated at time $t \in T$. Given the values of these variables, the remaining variables have values that are easy to determine. We set $z_{ajt} = 1$ if $z'_{aj} = z''_{jt} = 1$ and 0 otherwise. If agent $a \in A$ decides to move between time steps $t \in T$ and $t+1$, we denote the traveled distance by $f_{at} \geqslant 0$. If this movement is from $u \in V$ to $v \in V$, we set f_{at} equal to the travel distance $C^{\rightarrow}_{uv}$. Furthermore, we keep track of the 'makespan' $\bar{z}$: if job $j \in J$ is initiated at time $t \in T$ and needs Q_j time steps to process, then the completion time is $t + Q_j$, and $\bar{z}$ is equal to the latest completion time among the jobs. Finally, at every time step $t \in T$ there are nodes $v \in V_P(t) \subseteq V$ at which an emergency can occur, and we indicate whether this emergency will be responded to from node $u \in V$ by setting y_{uvt} to 1 or 0.

One of the core objectives of our problem is minimizing emergency response time. Before we introduce the objective function of E-MRP formally, we make two remarks about how response time is defined. If an emergency occurs at node v in the time interval $(t-1, t]$ and the responding agent is at node u at time t, we denote the response time as $C^{!}_{uvt}$.

We remark first that $C^{!}_{uvt}$ can be completely unrelated to the graph distance between u and v. That is, when an emergency occurs, agents are free to ignore the graph structure and move over the fastest physical route to the emergency location, possibly even with higher speed due to being an emergency vehicle. While it may seem counter-intuitive that the graph distances do not always correspond exactly to the distances possible in the physical world, this is a result of discretizing time and space: for instance, two adjacent nodes can be only a few minutes apart in the physical world, but the graph distance would be rounded up to 'one time-step'.

As a second remark on response time, in both MRP and E-MRP, we only minimize the response time to the next emergency. When an emergency does occur, and the responding agents are dispatched, we assume the planner will create a new planning for the remainder of the time horizon, using the remaining agents, jobs and time-steps as input. While it may be somewhat myopic to prepare only one emergency ahead, it is computationally much more expensive to look multiple emergencies ahead, as this would likely require stochastic programming. Although this could improve the response time to emergencies beyond the first, we conjecture that the gain will be marginal, especially in use cases where emergencies occur infrequently. For use cases with frequent emergencies, one can hope that the response organization accordingly has many responders in its fleet, which would again limit the impact of planning only one emergency ahead.

To briefly conclude: for the emergency response time, we use the distance matrix $C^{!}_{uvt}$ that is unrelated to the graph, and we optimize the expected response time to only the next emergency because the planner can create a new planning after each emergency.

In E-MRP, we wish to minimize a sum of four objectives, weighted with normalizing factors $\phi_{response}, \phi_{distance}, \phi_{preference}$ and $\phi_{makespan}$, respectively. Firstly, we want to minimize the expected response time to emergencies in the network. At each time step $t \in T$, we know of each danger-node $v \in V_P(t)$ the probability $P_{v,t}$ of the next emergency occurring there. Once we have decided the positions x_{avt} of the agents throughout time, we know from which nodes $u \in V$ emergency coverage can be given to (v, t). We only count agents who are available for response at that time, as $Y^{✓}_{at} \in \{0, 1\}$ indicates whether agent $a \in A$ is available for emergency response at time t. It is always optimal to choose the nearest remaining u for coverage of $v \in V$. So we know the probability P_{vt} for the next emergency to be at (v, t), and we know the corresponding response time $C^{!}_{uvt}$, meaning the expected response time to the next emergency equals $\sum_{t \in T, v \in V_P(t), u \in V} P_{vt} C^{!}_{uvt} y_{uvt}$.

Secondly, we want to minimize the total time agents spend traveling, aside from emergency response. Movement requires fuel, and preventive jobs often cannot be performed while moving. This objective simply equals $\sum_{a \in A, t \in T(a)} f_{at}$.

Thirdly, planners may have their own preferences of assigning specific jobs to specific agents for reasons beyond this model. It may be, for instance, that an agent will soon have an exam for a certain type of job, and should practice that job as much as possible. We therefore define a 'penalty' $N_{ja} \geqslant 0$ of assigning agent $a \in A$ to job $j \in J$ to discourage unwanted assignments. This final cost component is equal to $\sum_{a \in A, j \in J} N_{ja} z'_{aj}$.

Finally, we want to minimize the makespan $\bar{z}$. If jobs are scheduled near the end of the shift, then any emergency will make it likely that the planned jobs can no longer be performed that day. Additionally, minimizing the makespan implicitly means that the workload is divided as 'fairly' over agents as possible.

3.2. Mixed integer linear program formulation

We present a mixed integer linear program (MILP) formulation for the E-MRP. Technically, the MILP describes a problem more general than E-MRP: planning is done over several shifts, and allow jobs $j \in J$ to be ignored against a penalty $C^{\times}_j$. We present this more general MILP, because it allows us to describe most of the subroutines in this article as a small variation of this MILP.

Let H be the set of shifts. Denote $T(h) \subseteq T$ the time steps in shift $h \in H$. Let $A(h) \subseteq A$ be the set of agents working shift h. For each $a \in A(h)$, denote $t^{\triangleright}_{ah} \in T(h)$ and $t^{\square}_{ah} \in T(h)$ the start and end time of agent a in shift h. Denote $\overrightarrow{J}(u, v) := \{j \in J : L^{\triangleright} = u, L^{\square} = v\}$ the jobs that start at u but end at v. Let the binary decision variable z'''_j indicate whether you decide to do the job $j \in J$. Then we present the following MILP. We fix

$x_{avt} = 1$ for every $(a, v, t) \in X^{!}, x_{avt} = 0$ for every $t \in T$ if $X^{\checkmark}_{av} = 0$, and $z^{'}_{aj} = 1$ if $Z^{!}_{aj} = 1$.

$$\min \quad \phi_{response} \cdot \left(\sum_{t \in T, v \in V_P(t), u \in V} P_{vt} C^{!}_{uvt} y_{uvt} \right) + \phi_{distance} \cdot \left(\sum_{a \in A, t \in T(a)} f_{at} \right)$$
$$+ \phi_{preference} \cdot \left(\sum_{a \in A, j \in J} N_{ja} z^{'}_{aj} \right) + \phi_{makespan} \cdot \overline{z} + \phi_{ignoring} \cdot \left(\sum_{j \in J} C^{\times}_j \left(1 - z^{'''}_j \right) \right)$$

subject to

$$\sum_{v \in V} V^{\triangleright}_{av} x_{avt^{\triangleright}_{ah}} = 1 \qquad \left(\forall h \in H \right) \left(\forall a \in A \left(h \right) \right) \tag{1}$$

$$\sum_{v \in V} V^{\square}_{av} x_{avt^{\square}_{ah}} = 1 \qquad \left(\forall h \in H \right) \left(\forall a \in A \left(h \right) \right) \tag{2}$$

$$\sum_{v \in V} x_{avt} = 1 \qquad \left(\forall a \in A \right) \left(\forall t \in T \left(a \right) \right) \tag{3}$$

$$x_{avt} \leqslant \sum_{u \in V_v} \left(x_{au(t-1)} + \sum_{j \in \overrightarrow{J}(u,v)} z_{aj\left(t - Q_j\right)} \right) \qquad \left(\forall a \in A \right) \left(\forall v \in V \right) \left(\forall t \in T \left(a \right) \setminus \left\{ 0 \right\} \right) \tag{4}$$

$$\sum_{a \in A} \sum_{t \in T} B^{\checkmark}_{ab} z^{'}_{aj} \geqslant M_{jb} z^{'''}_j \qquad \left(\forall j \in J \right) \left(\forall b \in B \right) \tag{5}$$

$$z_{ajt} \leqslant z^{''}_{jt} \qquad \left(\forall a \in A \right) \left(\forall j \in J \right) \left(\forall t \in T \right) \tag{6}$$

$$z_{ajt} \leqslant z^{'}_{aj} \qquad \left(\forall a \in A \right) \left(\forall j \in J \right) \left(\forall t \in T \right) \tag{7}$$

$$z_{ajt} \geqslant z^{'}_{aj} + z^{''}_{jt} - 1 \qquad \left(\forall a \in A \right) \left(\forall j \in J \right) \left(\forall t \in T \right) \tag{8}$$

$$\sum_{t \in T} z^{''}_{jt} = z^{'''}_j \qquad \left(\forall j \in J \right) \tag{9}$$

$$\sum_{\tau = t}^{t + Q_j - 1} Z^{\checkmark}_{at} x_{aL^{\triangleright}_j \tau} \geqslant Q_j z_{ajt} \qquad \left(\forall a \in A \right) \left(\forall j \in J \right) \left(\forall t \in T \right) \tag{10}$$

$$Z^{\checkmark}_{a\left(t+Q_j\right)} x_{aL^{\square}_j\left(t+Q_j\right)} \geqslant z_{ajt} \qquad \left(\forall a \in A \right) \left(\forall j \in J \right) \left(\forall t \in T \right) \tag{11}$$

$$\overline{z} \geqslant \left(t + Q_j \right) z^{''}_{jt} \qquad \left(\forall j \in J \right) \left(\forall t \in T \right) \tag{12}$$

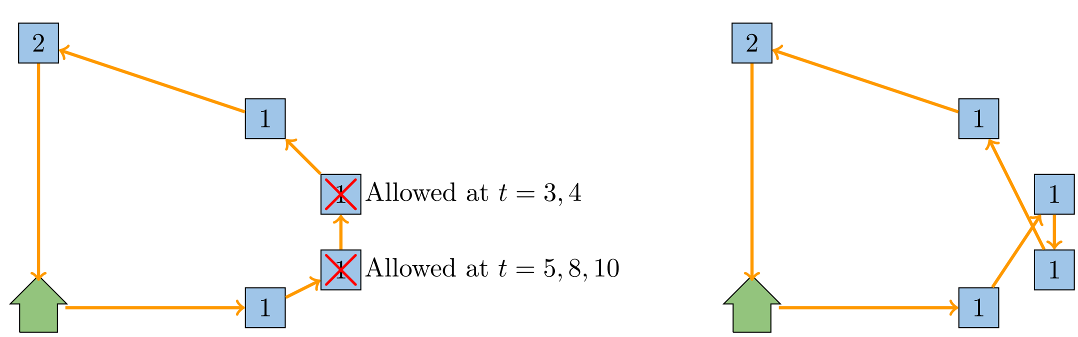
$$\sum_{u \in V} y_{uvt} = 1 \qquad \left(\forall t \in T \right) \left(\forall v \in V_P \left(t \right) \right) \tag{13}$$

$$y_{uvt} \leqslant \sum_{a \in A : t \in T(a)} Y^{\checkmark}_{at} x_{aut} \qquad \left(\forall u \in V \right) \left(\forall t \in T \right) \left(\forall v \in V_P \left(t \right) \right) | \tag{14}$$

$$C^{\rightarrow}_{uv} \left(x_{aut} + x_{av(t+1)} - 1 \right) \leqslant f_{at} \qquad \left(\forall a \in A \right) \left(\forall t \in T \left(a \right) \setminus \overline{T} \right) \left(\forall u \in V \right) \left(\forall v \in V \left(u \right) \right) \tag{15}$$

$$x_{avt}, z_{ajt}, z^{'}_{aj}, z^{''}_{jt}, z^{'''}_j \in \left\{ 0, 1 \right\}, y_{uvt}, \overline{z}, f_{at} \geqslant 0.$$

The objective function consists of the four components described in Section 3.1, with the added job ignoring penalties $\sum_{j \in J} C^{\times}_j (1 - z^{'''}_j)$. Constraints (1) and (2) indicate that agents must start and end each of their shifts at locations that are allowed for them. Constraints (3) state that an agent can be in only one place at a time. Constraints (4) state that agents can only move to adjacent nodes. An exception is made when performing jobs that end in different places than they start: if an agent initiates a job $j \in J \in \overrightarrow{J}(u, v)$ at time $t \in T$, then we allow that agent to 'teleport' from u to v at time $t + Q_j$. Constraints (5) encode that each job needs a certain number of agents for each authorization. Constraints ()()()(6)–(8) set the relation that $(z_{ajt} = 1) \Leftrightarrow (z^{'}_{aj} = z''jt = 1)$. Constraints (9) state that if you decide to do a job $j \in J$, it must get exactly one starting time. Constraints (10) ensures that agents who start a job stay at its location for the duration, except in its final step, when constraints (11) put them at the job end location. Constraints (12) make $\overline{z}$ behave as the makespan, or the latest completion time among jobs. Constraints (13) state that at every time $t \in T$, the danger-nodes $V_P(t)$ need emergency coverage. Constraints (14) state, however, that coverage can only be given from nodes with at least one agent on them who is available for emergency response. Finally, constraints (15) encode that if agent $a \in A$ starts moving at time $t \in T$ from node $u \in V$ to node $v \in V(u)$, then the distance f_{at} should equal $C^{\rightarrow}_{uv}$.



(a) The route MDSA suggests is infeasible; in this case, because some jobs cannot be initiated at all times ($R_{jt} = 0$).

(b) A feasible alternative.

Fig. 1. The first issue with MDSA: the most e?cient route may be infeasible, due to the many side constraints of E-MRP.

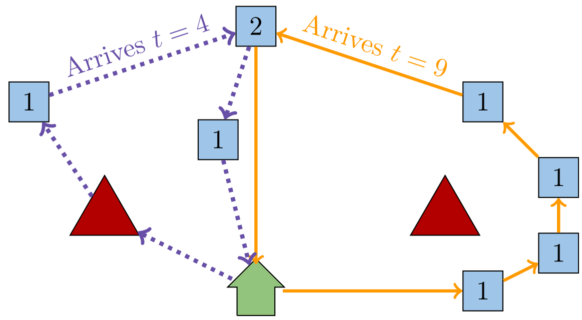


Fig. 2. The second issue with MDSA: if agents draw their own routes, they may arrive at cooperative jobs at completely different times. Due to time limits, we cannot always wait until everyone is present. If $\overline{T} = 13$, then these two routes are feasible in isolation, but infeasible when evaluated jointly.

We obtain a MILP for E-RMP by observing only one shift, and demanding $z_j''' = 1$ for all $j \in J$. We remark that constraints (1)–(5), (9), (13) and (14) were already present in some form in MRP, as well as the response time objective. See also Appendix A. The parameters, variables and indices added to those constraints, together with the new constraints and objectives, constitute the difference between MRP and E-MRP.

3.3. An MDSA-inspired heuristic

While it is possible to solve an instance of E-MRP by plugging the MILP into a MILP solver, the required computation times are much too long and unpredictable for this specific application. Whenever an emergency occurs, planners need to have a new planning within minutes, and they cannot afford to wait hours or days for the optimal one. Therefore, a fast heuristic is needed that provides good solutions within a few minutes. In Huizing et al. (2020), MDSA was found to be the most effective heuristic for MRP. Roughly, MDSA consists of four steps, which are outlined below for later reference.

Step 1: MEDIATE

In this step, the ideal positions for emergency response are determined by solving a p-Median Problem. These medians are matched to agents, such that the distance between the median and the agent's start and end location are minimal in total. Agents are temporarily solely responsible for emergency response in the region belonging to their median.

Step 2: DIVIDE

In this step, jobs are assigned as much as possible to the nearest median and its agent.

Step 3: SEQUENCE

In this step, each agent decides in what order to visit the jobs assigned to them. Discrete routes over the network are then drawn with optimal response times to the region of that agent.

Step 4: AGREE

In this step, it is evaluated if the current routes justify the initial division of emergency nodes over agents, and a small re-optimization is performed.

Adapting MDSA to E-MRP is non-trivial, due to the many new features. To be more specific, the two main hurdles to be taken are the following: (1) it is no longer given that any agent can do any task at any time, and (2) because some tasks may require multiple agents, agents are no longer free to independently sequence their jobs.

An example of the first hurdle is given in Fig. 1. If we simply compute the fastest route between all jobs assigned to an agent, we can no longer guarantee that this is feasible. For instance, it may be that some jobs j are quite restrictive in their allowed starting times R_{jt}. It no longer suffices to encode with decision variables if one job is followed by another; we must also decide the starting time t and check whether $R_{jt} = 1$. Similarly, appointments $X^{!}$ must also be fit into the schedule, giving us another reason that we must explicitly track arrival times. These are only two of the many additional features to account for in dividing and sequencing the jobs. The second hurdle, illustrated in Fig. 2, causes an even deeper issue in the methodology. In MDSA, we first divide jobs over agents, then allow them to independently decide in what order to visit these jobs. However, because we now have jobs that require multiple agents, we need to ensure that agents arrive at such jobs at the same time. This makes it very difficult to decouple the routing decisions per agent, especially in conjunction with the routing challenges in the first hurdle. In conclusion, the arrival of each agent at each key location must be synchronized much more closely, and MDSA *as is* falls down. A new algorithm is needed.

Fortunately, by understanding what made MDSA successful, we can craft a heuristic in the same spirit. To be more specific, for MDSA the quality of the solutions was concluded to be mainly due to the MEDIATE-step (outlined above). Because it is NP-complete to produce a feasible solution for MRP, MILP solvers were required as subroutines, but they terminated very quickly due to the problem being split and heavily compressed.

For E-MRP, we propose the heuristic described in Algorithm 1. We again start with determining good medians. Because finding a feasible solution is NP-complete, we need a MILP solver as a subroutine, but we can greatly speed this up by compressing the underlying network. We again split the decisions over several steps: the MILP mainly finds a feasible job schedule with reasonable distance to the medians, but we only explicitly optimize emergency response times and traveled distances in later steps.

Algorithm 1. High-level overview of the MDSA-inspired algorithm for E-MRP.

1: Obtain $|A|$ medians optimal with respect to $\sum_{t \in T} P_{vt} C^{!}_{uvt}$
2: Obtain a simplified subset $V' \subseteq V$ containing all medians and the start and end locations of agents and jobs
3: Run the MILP, but with nodes V', and $P_{vt} > 0$ only for the medians, and $|H| = 1$, and $z''' = 1$, and $\phi_{distance} = 0$, and an appropriate time-out.
4: Fixing z from the previous step, as well as $X^{!}$, determine where in V any agent $a \in A$ can be at any time $t \in T$
5: Sort A on how many start locations they can still choose from, then greedily choose start locations minimizing response time and travel time to the first goal
6: For $t \in T$, for each active agent, greedily decide their next hop based on response time and travel time

Below we elaborate on the steps in Algorithm 1.

1. Node weights and response times are now time-varying, but for simplicity, we sum these over time. This is equivalent to taking average weights and response times. This allows us to solve one p-Median problem for the entire time horizon, which is much less computationally intensive than solving one for each time step. The underlying assumption is that, though some time steps will have more emergencies and higher travel times, each node will retain more or less the same fraction of the emergency weight, and the increase in response times behaves like a constant multiplication. This is justifiable when emergency probabilities and response times increase mainly due to the morning and afternoon rush hours, and these multiplication factors are uniform over the network. Note that if P_{vt} and $C^{!}_{uvt}$ are static enough, then in practice, we can maintain a preprocessed list of medians for the typical numbers of agents in a shift.
2. It is computationally costly to input many nodes into the MILP. Therefore, we find the smallest subgraph that preserves the graph distances between the points of interest, including the medians, using the technique in Huizing, Schäfer, van der Mei, and Bhulai, (2022). If the start and end locations of the agents and jobs are static enough, this step can also be preprocessed.
3. We run the MILP from Section 3.2 for a limited time. Because we use it for an instance of E-MRP, we enforce that every job must be

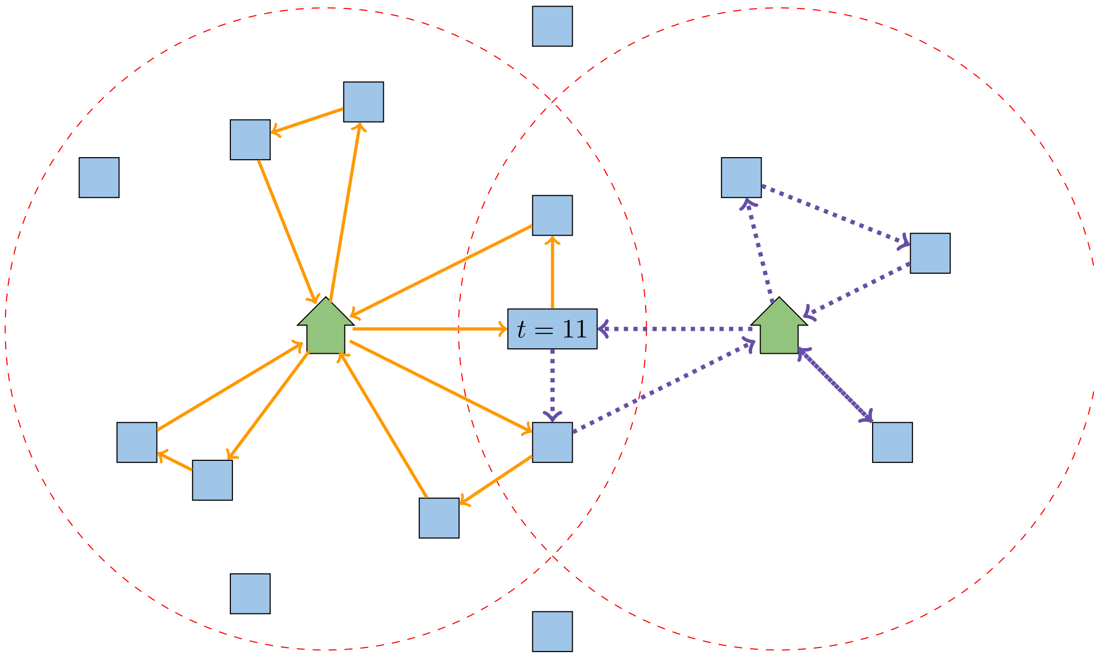


Fig. 3. A schematic representation of the current practice. Two agents, living at the green house-shaped nodes, determine which jobs are within a given radius of their home base. They disregard jobs outside of that radius. One job in their intersection has to be done by at least two people: a planner decides these two should do it, at time $t = 11$. Apart from that, agents are free to divide jobs over their shifts. The left agent divides eight jobs over his four shifts (solid orange lines), the right agent divides five jobs over his three shifts (purple dotted lines). They are unaware that the same job (just below the cooperative job) is visited by both of them at some point, which is superfluous. The left agent could have instead visited a job that now no one does.

scheduled, and we only observe one shift. We knowingly delay optimizing the traveled distance to a later step, trusting that a good makespan will imply an agent's jobs being close together. This also allows us to skip constraint (15), which is by far the most costly to build.

4. In later steps, we will greedily choose cost-improving hops. However, we need to make sure that we do not make choices that turn out to be infeasible. Therefore, we perform this step to compute which nodes can be visited at which times when abiding by the job schedule z from the previous step and $X^!$. For each agent, we determine from z and $X^!$ their fixed locations at those times. Then, because this set of fixed locations is small and the network adjacency is constant over time, we can make the following computation. For each location of interest for an agent $a \in A$, we use breadth-first search to determine how many hops away it is from each node $v \in V$, remembering that $a \in A$ can only visit nodes $u \in V$ with $X^{\checkmark}_{au} = 1$. Then if a has a fixed location (u_1, t_1) and next has to be at (u_2, t_2), we know for each $v \in V$ that $a \in A$ can be there at times $t_1 + hops(u_1, v) \leqslant \tau \leqslant t_2 - hops(v, u_2)$, which may be an empty set of τ. The result of this step is, for each agent $a \in A$ and each time step $t \in T$, a list of nodes $V_z(a, t) \subseteq V$ which that agent can visit at that time.
5. While agents can have multiple start locations to choose from, this selection may be further limited by fixing the job schedule z. We sort A in increasing order of how many start locations they can still choose according to V_z. We initiate the set U of chosen start locations empty. Then, for each agent $a \in A$ in the sorted A, and for each allowed start location u, we determine the summed response cost for $U \cup \{u\}$ as $(C^!)'_u = \sum_{t \in T, v \in V_P(t)} P_{vt} \min_{u' \in U \cup \{u\}} C^!_{u'vt}$. If a visits any jobs or appointments, and l is the starting location of the first such visit, we set $(C^{\rightarrow})'_u = C^{\rightarrow}_{ul}$; otherwise, $(C^{\rightarrow})'_u = 0$. We then pick the start location u with minimal $\phi_{response}(C^!)'_u + \phi_{distance}(C^{\rightarrow})'_u$, set the location of a at their start time to u, add u to U, and move on to the next agent.
6. For each time step $t \in T \backslash \{\overline{T}\}$, and each agent active at that time step and the next, we obtain their current location v and look at all nodes in $V_z(a, t+1)$. Let U be the (possibly empty) set of locations for which it is decided that some response available agent will be there at $t+1$. Then for each $u \in V_z(a, t+1)$, we again determine $(C^!)'_u = \sum_{t \in T, v \in V_P(t)} P_{vt} \min_{u' \in U \cup \{u\}} C^!_{u'vt}$, and we set $(C^{\rightarrow})'_u = C^{\rightarrow}_{vu}$. We let the location of a at time $t+1$ be $u := \mathrm{argmin}_{u \in V_z(a, t+1)} (\phi_{response}(C^!)'_u + \phi_{distance}(C^{\rightarrow})'_u)$.

In summary, Algorithm 1 yields a schedule for the jobs, and a movement instruction for the agents. By construction, this algorithm will find a feasible solution if it exists, unless the time-out on step 3 is too narrow.

4. Current practice model

To assess the performance improvement of our E-MRP solution with the performance obtained with the current way of planning, it is important to understand how the planning is done in current practice. By making an accurate quantitative model of the current practice, we can computationally investigate the influence that certain decisions have on different metrics. This allows us to investigate a large number of potential scenarios. Furthermore, a model allows us to compare the performance in these scenarios, without having to track how well our proposed solution is abided by in practice. To this end, we have set up extensive interviews with task planners from our partnering railway provider. This has led to a common understanding of the informal steps taken in the current way of planning, which is mainly done manually (without any support from MRP). Throughout, this model will be referred to as the Current Practice (CP) model, and is described below. In the next section, this CP-model will be used as a benchmark to compare the performance of the E-MRP model proposed in this paper for a realistic use case scenario obtained from our partnering railway provider.

We have observed ten shifts in a single week. A shift consists of several agents, and a shift leader responsible for their coordination. Some agents, including the shift leader, have a role in a shift that requires them to stay at the base of operations. Almost every shift has a designated truck driver, who cannot stray too far from the garage of the heavy emergency-response truck, which is also at the base station.

In current way of working, agents in our case study make their planning in a rather decentralized manner. Some of the preventive jobs require some paperwork and planning, or collaboration, and these are typically planned and assigned by the shift leader. After that, however, the shift leader asks the agents to propose for themselves which jobs they will do, based on their own expertise and preferences. Agents have often accumulated local knowledge in the area around their home, and prefer to stay in that area. They will typically divide the tasks close to their home over their shifts that week. Once they have performed a sufficient amount of work, they are allowed to await emergencies from their home base. See also Fig. 3.

As long as this means someone is active in the 'northern part of the region', as well as in the 'middle part' and 'southern part', the expected emergency response time is low enough for the shift leader to be content. Working hours have been designed around that logic, and typically, each shift contains an agent living in the north, the middle and the south. For the ten shifts in our case study, the active agents indeed have home bases in these three areas, and the decentral plans of the agents were not interfered with.

In this way of working, agents may know roughly in what corners of the networks their coworkers are, but they do not know where exactly or what they are doing. Some agents in the late shift may choose to perform a periodic patrol on a piece of the railway, not knowing that a coworker has already patrolled there the same morning. Moreover, agents in the *same* shift may encounter each other patrolling the same piece of the railway, and continue their patrol together. Two agents staying together is clearly suboptimal with respect to emergency response times. Meanwhile, if some other part of the railway has been neglected for weeks on end, this is untracked and unknown.

The current practice has some practical benefits: it requires very little communication, which makes it robust in the face of emergencies. Furthermore, experience with a given piece of the railway is undoubtedly beneficial to inspecting it effectively. However, now that the communication infrastructure and the size of the organization are growing, we believe our model and heuristic can help coordinate schedules and decrease response times. We will support this claim in the remainder of this article.

We obtain all sets and parameters from the case study data, with one exception. For each agent $a \in A$, from among the variable starting and ending locations, only V_a^n is allowed, the one nearest to their home. If an agent works from the base station or as a truck driver, the base station is taken as the start location. Because the agents must select a subset of jobs to perform, we provide a job 'importance' $C_j^{\times}$ that was also approved by the employee of the organization. We simulate the current practice as follows, with H the ten shifts in the observed week.

Let $\mathtt{MILP}(A', J', V', H', F)$ denote that we call the MILP from Section 3.2 with agent subset A', job subset J', node subset V', shift subset H', and a collection F of fixed values for certain variables and parameters. Then Algorithm 2 describes the way plans are made in the CP-model.

Algorithm 2. Mathematical interpretation of the current practice.

1: Collect a simplified node set V', either precomputed or from Huizing et al. (2022).
2: Encode in collection F_2, for all shifts $h \in H$ and agents $a \in A(h)$ and time-steps $t \in T(h)$, that $x_{a,V_a^n,t_{ah}^{\triangleright}} = 1, x_{a,V_a^n,t_{ah}^{\square}} = 1$ and that $x_{avt} = 0$ if node v is not accessible for agent a because of their role in the shift h. Also encode for each agent $a \in A$ and each job $j \in J$ that $z'_{aj} = 0$ if j is too far away, that is, $C^{\rightarrow}_{V_a^n, L_j^{\triangleright}} > 60$ or $C^{\rightarrow}_{V_a^n, L_j^{\square}} > 60$.
3: Compile centralized jobs J': jobs that are only done once, and jobs j that must be assigned at least two agents, meaning $\neg \exists a \in A : (B^{\checkmark}_{ab} \geqslant M_{jb} \ \forall b \in B)$.
4: The planner plans J'. That is, encode in a new collection F that $z''' = 1$, and $\phi_{response} = 0$ and $\phi_{ignoring} = 0$. Then call $\mathtt{MILP}(\{a\}, J', V', H, F \cup F_2)$. Encode in collection F_4 the following: $z'_{aj} = 1$ if $j \in J'$ is assigned to $a \in A$ and 0 otherwise, and $z''_{jt} = 1$ if multi-agent job $j \in J'$ is scheduled at time-step $t \in T$.

(*continued on next column*)

(*continued*)

5: Each agent $a \in A$ divides jobs over their shifts. That is, encode in a new collection F that $\phi_{response} = \phi_{preference} = 0$. Then call $\mathtt{MILP}(\{a\}, J(a), V', H, F \cup F_2 \cup F_4)$. For each shift $h \in H$ and each agent $a \in A(h)$, denote $J(a, h)$ the jobs agent $a \in A$ schedules in shift $h \in H$.
6: Agents minimise their makespan in each shift. That is, for each $h \in H$ and $a \in A(h)$, encode in a new collection F that $z'''_j = 1$ for all $j \in J(a,h)$, and $\phi_{response} = \phi_{preference} = \phi_{distance} = \phi_{ignoring} = 0$. Then call $\mathtt{MILP}(\{a\}, J(a,h), V', \{h\}, F \cup F_2 \cup F_4)$. Encode in collection F_6 that $z''_{jt} = 1$ if start-time t was chosen for job j.
7: Agents minimise their distance, abiding by this makespan. That is, for each $h \in H$ and $a \in A(h)$, compute the $C^{\rightarrow}$-shortest tour from V_a^n across $J(a, h)$, while not violating $F_2 \cup F_4 \cup F_6$. If this tour is shorter than $T(h)$, let the agents spend the remainder of their shift at V_a^n.

In words, Algorithm 2 does the following.

- We compile the jobs $J' \subseteq J$ that the shift leader decides on. These are the jobs that require multiple agents, and the jobs that cannot be done by agents without the shift leader providing some paperwork. We simulate these jobs being planned over the week, by solving a variation of the MILP from Section 3.2. That is, we fix z'_{aj} to 0 if job $j \in J$ is not within one hour of the home of agent $a \in A$, measured in $C^{\rightarrow}$. We set the start and end locations of the agents as discussed by fixing $x_{avt} = 1$ if $v \in V$ is the default start and end location of a for that shift, and $t \in T$ is a shift start or end time for a that shift. If a is a driver or has to stay at the base station, we forbid the other locations by setting $x_{avt} = 0$. We demand all jobs J' are assigned, so we set $z'''_j = 1$ for all $j \in J'$, as J' is quite small in this case study. We set $\phi_{response} = \phi_{ignoring} = 0$. As we care mainly about setting start times for those few jobs in J', it suffices to use a simplified node set V'. J' is small enough that we believe a high-quality schedule can be found even when setting a time-out on the MILP. The result is a decision of whom the jobs in J' are assigned to, and in the case of multi-agent jobs, at what time they will be performed.
- Each agent $a \in A$ assembles the jobs $J(a) \subseteq J$ within one hour of their start location, and divides these over the week using the MILP. Some values of x_{avt} and z'_{aj} are fixed by the same rules as in the previous step. In addition, the previous step assigned the jobs J' to specific people, and we further fix z'_{aj} accordingly. If those jobs were in J' because they required several people, we also enforce $z''_{jt} = 1$ for the chosen start time t of job $j \in J'$, to ensure that everyone shows up at the same time.

  However, after having those variables fixed, we allow a to decide which jobs to do when. We use the agent set $\{a\}$, the simplified nodes $V' \subseteq V$, and the nearby jobs $J(a) \subseteq J$. We do not constrain z'''_j, and we set $\phi_{response} = \phi_{preference} = 0$. We run the MILP with a time-out. The result is a decision, for each agent, what jobs they will do in each of their shifts.
- Each agent $a \in A$ then determines, for each of their shifts $h \in H$, how to do the jobs $J(a, h)$ they chose for that shift as quickly as possible. We first minimize the makespan. This is done with the MILP, with agents $\{a\}$, jobs $J(a, h)$, simplified nodes V', and shifts $\{h\}$. We fix $z'''_j = 1$, and set $\phi_{response}$ and $\phi_{distance}$ and $\phi_{preference}$ and $\phi_{ignoring}$ to 0. After running this MILP to optimality, we use an adaptation of Dijkstra's algorithm to find the quickest route with respect to $C^{\rightarrow}$. This route starts at the start location, visits the jobs at the times scheduled by the MILP, and goes back to the start location as quickly as possible. The result is a complete solution: for each shift, we know where the agents are at each time step and what jobs they start performing, albeit that some jobs may be performed multiple times.

We emphasize that the CP-model described above describes the informal steps taken by planners in our partnering railways company in today's practice. Simulation experiments show that the response times predicted by the CP-model closely match those observed actually observed in current practice. As we will see in Section 6, the CP-model predicts an expected response time of 41.1 min. In a recent sample from the partnering company, an average response time of 43.9 min was measured over 45 incidents, with a standard deviation of 17.31 min. If our hypothesis is that these incidents were sampled from a distribution with mean 41.1, then the measured 43.9 corresponds to a p-value of 0.285, which implies that the hypothesis can not be rejected.

From this, we conclude that the CP-model gives a good description of the current way of working. In the remainder of this paper, we will use this CP-model to answer what-if questions about the performance under different, both realistic and hypothetical, planning situations.

5. Performance metrics, planning scenarios and use case description

In this section, we benchmark the performance of the proposed solution to the E-MRP model with the performance of the CP-model. For this, we have performed extensive simulations for a range of realistic planning scenarios. In Section 5.1, we will discuss the performance metrics we wish to improve on. In Section 5.2, we will describe the planning scenarios we have simulated to investigate the potential improvements. In Section 5.3, we describe the real-life use case that use for our experiments. The remaining details of our simulation experiment are given in Section 5.5.

5.1. Performance metrics

In our experiments, we consider the following seven relevant performance metrics:

- **Expected response time**: given the positions x_{avt} of a solution, we know the positions $U(t) := \{u \in V : \sum_{a\in A} Y^{\checkmark}_{at} x_{aut}\}$ from which emergency response can be given. The expected response time then equals $\sum_{t\in T, v\in V_P(t)} \min_{u\in U(t)} P_{vt} C^{!}_{uvt}$.
- **Percentage of incidents with response time less than 15 min**: Again given $U(t)$, we define this metric as

$$\sum_{t\in T, v\in V_P(t)} P_{vt}\cdot 1\left(\exists u \in U\left(t\right) : C^{!}_{uvt} \leqslant 15\right)$$

- **Percentage of incidents with response time less than 30 min**: Same, but with 30 min.
- **Percentage of incidents with response time less than 45 min**: Same, but with 45 min.
- **Percentage of incidents with response time less than 60 min**: Same, but with 60 min.
- **Weighted unique jobs**: While we have defined the value $C^{\times}_j$ of doing job $j \in J$, this does not account for the fact that a set of decentralized agents may unknowingly do a job multiple times in the same week, which is considered 'useless'. We thus define this metric as $\sum_{j\in J} C^{\times}_j \cdot 1(j \textit{ is scheduled at least once})$.
- **Planned traveling**: Given the positions x_{avt}, this is simply the sum of $C^{\rightarrow}_{uv}$ for every movement from $u \in V$ at time $t \in T \backslash \{\overline{T}\}$ to node $v \in V$ at time $t+1$.

We do not take computation time into account as a metric, because computation time is only meaningful in a small subset of the scenarios we compare. In our experiments, however, our Algorithm 1-based solver needs only 19.8 s per shift, even on modest hardware and without a time-out on Step 3. Computation times, thus, are not a restricting issue in our application.

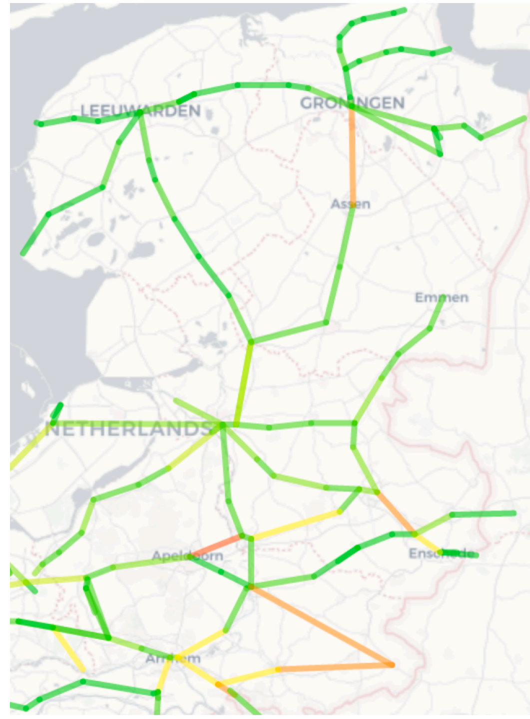


Fig. 4. The studied section of railway network in the North-Eastern part of the Netherlands. Green edges have low incident rate, while orange and red ones have high incident rates (based on historical incident data).

5.2. Planning scenarios

In practice, there are many choices to be made with respect to the restrictions in the planning. To gain insight into the implications of these options, we define a number of planning scenarios, for which the performance for our use case study is evaluated.

1. **Current practice ("own jobs, own route"):** agents pick their own jobs, schedules and routes, as described in Section 4.
2. **Jobless current practice ("no jobs, own route"):** in the later parts of shifts, agents are often already at home in the current practice. To see what kind of emergency response this incurs, we ran the same analysis as in Section 4, but with an empty set of jobs. The resulting solutions amount to agents starting at their start location, and staying there until the end of their shift.
3. **New practice ("planned jobs, solver route"):** as part of the case study, jobs were picked by an experienced planner from our industry partner, who had access to the solver running the heuristic from this article. The planner picked jobs for all shifts in the case study, and did so without interference from the researchers. These jobs were picked and divided over the shifts so that jobs are not done multiple times, but agents still have jobs somewhat near one of their start locations. In this planning scenario, the job schedules and routes are determined by running the heuristic in Section 3.3.
4. **Current jobs with heuristic ("own jobs, solver route"):** when comparing the current practice with the new practice, the quality of the new solutions depends in part on how well the planner chose the jobs. To filter out this effect, in this scenario, we ran the solver with the same set of jobs as in the current practice. The aim of this scenario is to show that, even without changing which jobs are done, the solver can still find job schedules and positions which perform better. We compute this scenario by deciding jobs for shifts as in Section 4, but then running the solver for each shift with the jobs thus chosen.
5. **Lower bound response time ("best response"):** to give more context to the results, we computed a lower bound on the expected

Table 2
Computational results of the seven scenarios. By 'Best $C^{!}$', we mean the 'Best response' scenario. By 'Best $C^{\times}$', we mean the 'best jobs' scenario. Most results of that scenario have been omitted, because only the weighted unique jobs matter there, and the other results are prone to lead to misinterpretation.

	Own jobs Own route	No jobs Own route	Planned jobs Solver route	Own jobs Solver route	Best $C^{!}$	Best $C^{\times}$	Manual jobs Solver route
Expected response (m)	41.1	41.8	35.9	32.7	30.4	–	35.3
Under 15 min (%)	6.7	3.4	9.6	10.6	12.8	–	9.3
Under 30 min (%)	30.5	29.0	42.1	46.6	53.1	–	41.4
Under 45 min (%)	60.3	61.8	72.0	78.2	85.9	–	73.7
Under 60 min (%)	83.6	82.6	90.4	94.9	97.2	–	93.1
Weighted unique jobs	71	0	45	71	0	102	93
Planned travelling (m)	3580.8	0.0	3284.7	5457.7	2459.6	–	3697.0

response time that can be achieved in this case study setting. We did so by running the MILP from Section 3.2 for each shift, but with $J = \emptyset$, and $\phi_{distance} = 0$.

6. **Upper bound weighted unique jobs ("best jobs"):** Likewise, to upper bound the weighted unique jobs metric, we ran the MILP with $\phi_{response} = \phi_{distance} = \phi_{preference} = \phi_{makespan} = 0$.
7. **Adjusted jobs with solver ("manual jobs, solver route"):** as it turned out, the planner selected significantly fewer jobs for the case study than the simulated agents could handle. This was because at the time, this new technology and way of working were unfamiliar, and not all available agents were taken into account. To correct for this, we manually picked a larger set of jobs to input into the heuristic.

5.3. Use case description

In this section, we describe the realistic use case, obtained from our industry partners, which we have used for our simulation experiments. In this case study, we observe a portion of the railway network. We have summarized this as a graph with 294 nodes, which we can simplify to a graph of 126 nodes using the node aggregation technique described in Huizing et al. (2022). The nodes represent either railway stations, the 'midway points' of railway sections between the stations, cargo stations or a depot office. Fig. 4 gives a visualization of the studied network section in the North-Eastern part of the Netherlands. A fleet of agents live in or near this network, and for privacy reasons, their default start and end locations are rounded to the nearest railway node. When using variable start locations, agents have on average 4.11 different locations to choose from. We have a collection of 45 regular tasks on this network: most of them consist of patrolling from one major station to another, but some are facility inspections that should happen exactly once per week. The jobs have different lengths, priorities and locations, and need either one or two agents. The facility inspections require a special training, and may only be performed during daylight hours.

We examine a working week of ten shifts: a morning shift and a late shift, from Monday to Friday. After discretization, the number of time-steps in a shift is either 14 or 12. The number of agents active in a shift lies somewhere between 4 and 10, averaging exactly at 7. Among these, there is always a 'manager' who stays at the depot and is only available for emergency response, and there is often a 'truck driver' who must stay within half an hour of the depot but can perform nearby activities. By design, each shift has at least one agent with a default location in the 'northern' part of the region, and an agent in the 'southern' part.

5.4. Concerning emergency simulation

In the case study organization, the number of emergencies is reasonably low: the organization reported 407 impacting emergencies in 2021 (ProRail BV, 2022), amounting to roughly one impacting emergency per day in the entire country. The studied portion of the railway network amounts to roughly one fourth of the country, meaning it is rare in our use case that more than one impacting emergency occurs in any given shift. However, when emergencies do occur, they are typically very major infrastructural undertakings and it is hard to predict how many agents and how much time are needed to resolve them. We trust the planner to decide which agents are still kept available and which jobs are still considered relevant and to use this to create a new plan for

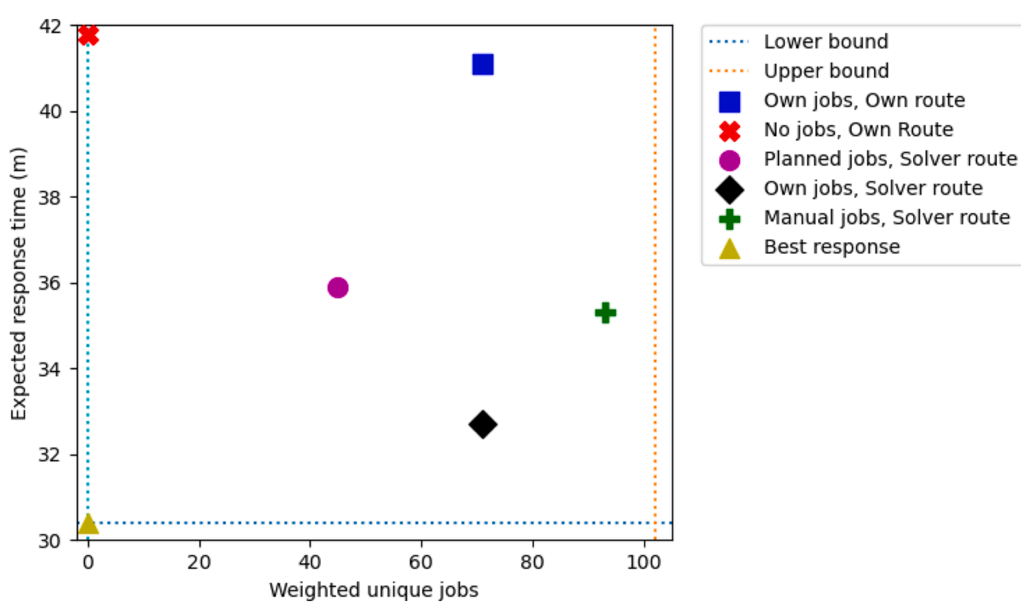


Fig. 5. Weighted unique jobs versus expected response time in the computed scenarios. On the horizontal axis, high is good; on the vertical axis, low is good.

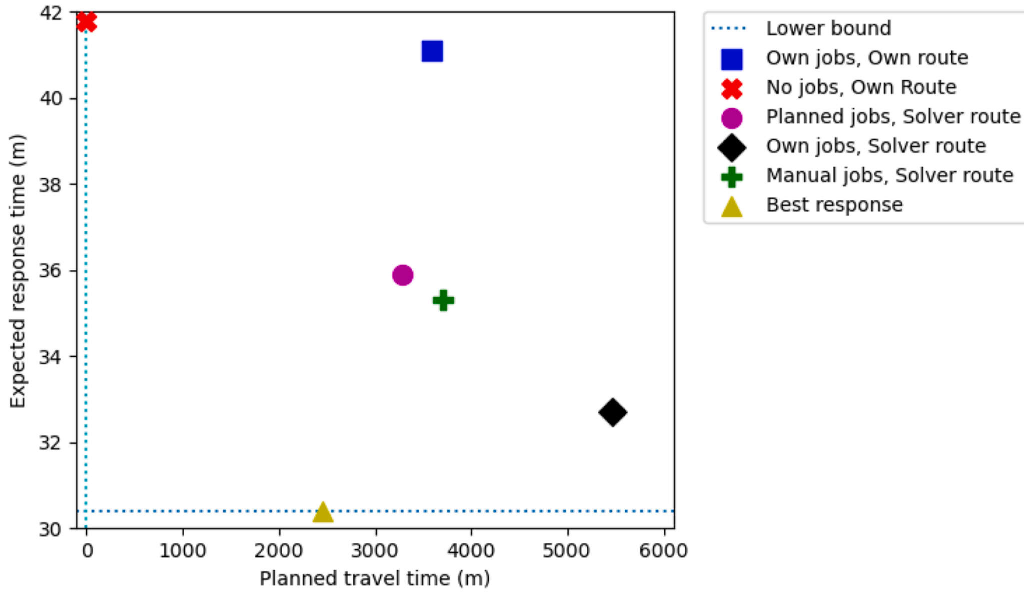


Fig. 6. Planned travel time versus expected response time in the computed scenarios. On both axes, low is good.

Table 3
Improvements with respect to the current ‘Own jobs, Own route’ scenario. Only the three scenarios that include jobs, and in which we care about response time, are taken into account.

	Planned jobs Solver route	Own jobs Solver route	Manual jobs Solver route
Improvement expected response time (%)	12.7	20.5	14.2
Reduction incidents later than 60 (%)	41.1	68.7	58.1
Improvement weighted unique jobs (%)	−36.6	0	31.0
Improvement planned travelling (%)	8.3	−52.4	−3.3

the remainder of the shift.

Therefore, to save on further complexity, we have left the actual simulation of emergencies out of scope for this study. Under the assumption that only one emergency occurs in any given shift, any emergencies that we would simulate would have response time that should converge to the *expected* response time that we do measure. We thus focus on how this expected response time can be improved, in combination with the other E-MRP objective components.

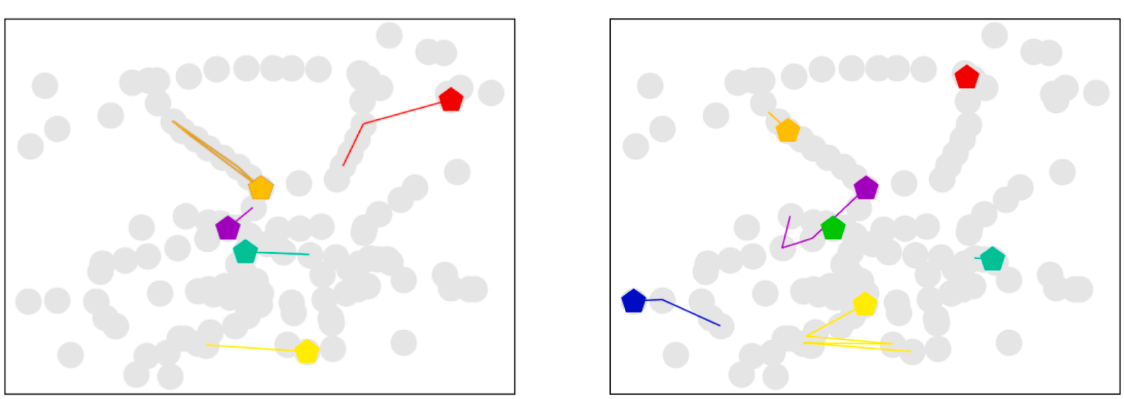
5.5. Implementation details

The case study organization has access to a solver running this heuristic on a dedicated server. However, for this article, all experiments were conducted on a single ThinkPad L470, with an Intel Core i5-7200U CPU processor, 2.50 GHz, 8 GB RAM.

6. Results

In this section, we present the results of our simulation experiments for our case study, and for the scenarios and KPIs listed above. Table 2 and Figs. 5 and 6 give an overview of the results. Table 3 gives the relative improvements of the KPIs compared to the CP-model.

Overall, the results show significant improvements compared to current practice. To elaborate, if we compare the CP (“Own jobs, Own route”) with the E-MRP solution (“Planned jobs, Solver route”), our



(a) Solution in current practice scenario. Five out of seven agents start near the center of the network, and four of them stay central.

(b) Solution in same shift in new practice scenario. The seven agents start with a better spread and largely maintain it.

Fig. 7. For one of the ten shifts, a visual comparison of the solutions in the current practice and the new practice. The grey circles are the 126 nodes of the simplified network. For each agent, a colored pentagon shows where they start, and the lines in the same color illustrate their movement over the network. In the new practice, agents are more equally spread over the network.

simulations indicate an improvement of 12.7% in average emergency response times. This number is significant, since taking averages over many response times usually tends to flatten out performance gains. Interestingly, looking at a 60-min response-time target, we observe that the fraction of late arrivals is reduced by as much as 41.1%. We even see a small reduction in the planned traveling time; however, this is likely due to the planned jobs being significantly fewer than the jobs agent picked for themselves.

If we do not change the jobs of a shift, but only use a solver to redivide the jobs and to determine positions of the agents, we see an even stronger reduction in response time. However, this comes at the cost of 52.4% increased planned traveling outside of jobs and emergencies. Movement is costly and irksome, and it is debatable whether a 20.5% improvement in response time is worth a 52.4% increase in travel time.

In the two scenarios where agents choose their own routes, we see expected response times of 41.1 and 41.8 min, respectively. In the three scenarios where the solver determines routes across non-empty sets of jobs, we see expected response times of 35.9, 32.7, and 35.3 min, respectively. This suggests that the improved response time is somewhat insensitive to the choice of jobs, but is mainly due to the routing being done by a solver. We expect this improvement in response time is mainly due to how ‘idle time’ is filled. If agents choose their own routes, the latter part of their shifts are spent near their homes, which are distributed somewhat arbitrarily. However, the solver often chooses waiting locations and start locations that are distributed more intelligently with the expected response times in mind. See also Fig. 7.

The trade-off between improved the expected response time and the movement needed to attain it is perhaps high-lit most by comparing the “No jobs, Own route” and “Best $C^!$” scenarios. In the former, we hit the lower bound of 0 min on planned movement. In the latter, we hit the lower bound of 30.4 min on the expected response time. In total, the latter requires 2459.6 min of planned traveling. Over the ten shifts, there are 62 shifts worked by agents, so a rough estimate amounts to an agent moving 39.7 min per shift to improve the expected response time to emergencies by 9.1 min, and to be ‘late’ only 2.8% of the times instead of 17.4% of the times.

Alternatively, we can investigate the trade-off by comparing the current situation with the scenario where the same jobs are chosen, but they are scheduled and routed by the solver. While doing the same jobs, agents have to move 30.3 min more on average over their shift to improve the expected response time by 8.4 min, and to be late only 5.1% of the times instead of 16.4% of the times. Our research is conducted under the notion that response time is much more important than planned traveling time. However, these numerical results could be valuable input for a management discussion concerning this trade-off.

As a final remark, our simulations demonstrate that for our realistic use case, we can simultaneously improve the expected response times by 14.2% *and* the number of weighted unique jobs by 31.0%. We observe this in our final scenario, where we manually correct for the small number of jobs chosen by the case study planner. While these improvements do come at the cost of more planned movements, this increase is only 3.3%. In this scenario, the number of weighted unique jobs is 93, which is only 9% away from the upper bound of 102. The expected response time is 35.3 min, which is only 16% away from the lower bound of 30.4. The jobs in this planning scenario were chosen manually, and if this job selection were made methodical, these improvements would likely become even larger.

7. Conclusion

In this paper, we have generalized the MRP-methodology to incorporate additional objectives and constraints that are required in practice. To quantify the performance improvement that can be realized, we performed an extensive comparison study by simulations, comparing current practice to the results from our optimization heuristics. We observed that, between the current practice and the new way of working allowed by our heuristic, we could strongly reduce response times and late arrivals in a realistic setting provided by our partnering railway provider. Most importantly, if the jobs are chosen ambitiously enough, we see that we can significantly improve both the response time performance and the (weighted) number of non-urgent tasks at the same time. We believe the reduction in response time is primarily due to agents being positioned in strategic locations during their idle time, rather than the endpoints of their job tours. This improvement can come at the expense of more planned movements outside of emergency response. However, in the most favorable scenario examined, this increase in planned movement is only marginal.

Finally, we address a number of topics for further investigation. Most importantly, we suspect that smartly assigning non-urgent jobs to agents intelligently is crucial to improve performance. In this research, tasks were mostly chosen by hand, which was seen as input to the model. But with an automated selection methodology, a system could smartly plan several shifts ahead independently. This requires further elaboration. Another promising direction is to include the incorporation of planned travel times in such a way that it does not require as many unfavorable constraints. Lastly, graph adjacency is currently independent of time, but this may not be realistic in cases with heavy traffic congestion.

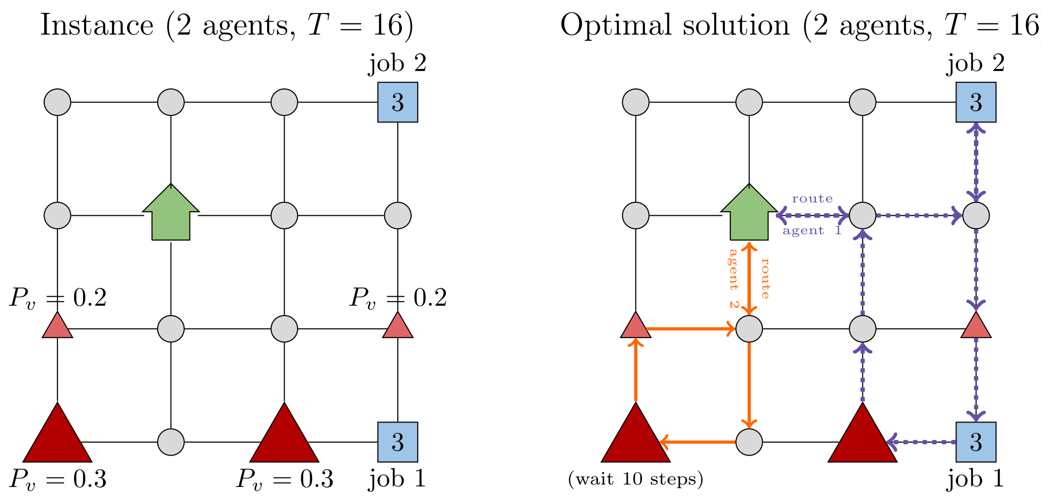
Acknowledgments

We thank our railway industry partner for co-funding this research and for supplying realistic case study data.

Appendix A. Overview of the Median Routing Problem

For reference, here we give an outline of the Median Routing Problem (MRP) introduced in Huizing et al. (2020). The goal of MRP is to have emergency responders perform non-urgent jobs in a network, but to spread them such that their expected emergency response time remains minimal. We provide a visual example in Fig. A.1. In MRP, we are given an unweighted, undirected graph with nodes V. We denote V_v the nodes within one ‘hop’ from $v \in V$, this being the neighbors of v and v itself. Each node has a non-negative weight P_v, and we denote V_P the nodes with strictly positive weight. There is a distance matrix $C^!_{uv}$ from nodes $u \in V$ to nodes $v \in V_P$, and this distance can be unrelated to the graph. We also have a set of agents A, who will provide ‘coverage’ over the network. They can move throughout a discrete-time horizon $T = \{0, \ldots, \overline{T}\}$, and if an agent is at some $v \in V$ at time $t \in T \backslash \{\overline{T}\}$, then they can be at any node in V_v at $t+1$. Note that it is allowed, thus, that agents wait where they are for some time-steps. Each agent $a \in A$ must start at a given start node $S(a)$ at time $t = 0$, and end at a given end node $E(a)$ at time $t = \overline{T}$. At time $t \in T$, we can examine the current locations $U(t)$ of the agents, and evaluate the ‘expected emergency response time’ at t as $\sum_{v \in V_P} P_v \min_{u \in U(t)} C^!_{uv}$. We seek feasible movement that minimises this response metric, summed over the entire discrete time horizon. However, aside from emergency coverage, there are jobs J that must be scheduled. Each job $j \in J$ has a location $L_j \in V$, and a number of time-steps Q_j required for processing the job. Once an agent starts a job j, they must stay at L_j for Q_j time-steps in a row: jobs cannot be interrupted.

We can write MRP as a MILP by introducing the following binary variables. We denote $x_{avt} = 1$ if agent $a \in A$ is at node $v \in V$ at time $t \in T$, and



(a) An instance. (b) An optimal solution.

Fig. A.1. An example of MRP. Grey circles are 'normal nodes'. Red triangles are nodes with positive weight P_v, the bigger ones having more weight. Blue squares are jobs containing a node with duration Q_j , and for this example, both agents start and end at the green pentagonal 'depot' node. The illustrated solution is optimal: on weighted average, each red triangle has an agent at a Manhattan distance of only 1.247.

0 otherwise. We denote $y_{uvt} = 1$ if emergencies at node $v \in V_P$ at time $t \in T$ are responded to from node $u \in V$, and 0 otherwise. Note that we can only set $y_{uvt} = 1$ if at least one agent is present at node u at time t. Finally, $z_{ajt} = 1$ if agent $a \in A$ starts job $j \in J$ at time $t \in T$ and 0 otherwise. We can relax y_{uvt} to be continuous, because MILP solvers will still choose integral y_{uvt} given x_{avt}.

$$\begin{aligned} \min \quad & \sum_{t\in T}\sum_{v\in V_P} P_v \sum_{u\in V} C^1_{uv} y_{uvt} \\ s.t. \quad & x_{aS_a0} = 1 \qquad (\forall a \in A) \end{aligned} \tag{A.1}$$

$$x_{aE_aT} = 1 \qquad (\forall a \in A) \tag{A.2}$$

$$\sum_{v\in V} x_{avt} = 1 \qquad \left(\forall a \in A\right)\left(\forall t \in T\right) \tag{A.3}$$

$$x_{av(t+1)} \leqslant \sum_{u\in V_v} x_{aut} \qquad \left(\forall a \in A\right)\left(\forall v \in V\right)\left(t = 0, \ldots, T-1\right) \tag{A.4}$$

$$\sum_{u\in V} y_{uvt} = 1 \qquad \left(\forall v \in V_P\right)\left(\forall t \in T\right) \tag{A.5}$$

$$y_{uvt} \leqslant \sum_{a\in A} x_{aut} \qquad \left(\forall u \in V\right)\left(\forall v \in V_P\right)\left(\forall t \in T\right) \tag{A.6}$$

$$\sum_{a\in A}\sum_{t\in T} z_{ajt} = 1 \qquad \left(\forall j \in J\right) \tag{A.7}$$

$$\sum_{\tau=t}^{t+Q_j} x_{aL_j\tau} \geqslant \left(Q_j + 1\right) z_{ajt} \qquad \left(\forall a \in A\right)\left(\forall j \in J\right)\left(\forall t \in T\right) \tag{A.8}$$

$$x_{avt}, z_{ajt} \in \{0, 1\}, \ y_{uvt} \in [0, 1].$$

Constraints (A.1) and (A.2) encode that the agents have fixed start and end locations. Constraints (A.3) state that an agent can only be in one place at a time. Constraints (A.4) enforce that agents move over the graph. Constraints (A.5) state that each emergency node needs coverage at each time, but constraints (A.6) state that coverage can only be given from a node that currently has agents on it. Constraints (A.7) are the hard constraints that all jobs are scheduled, and constraints (A.8) ensure that an agent who starts a job, stays for its duration.

MRP generalises the p-Medians problem, thus is NP-hard. We can solve it by the given MILP, and among the heuristics studied in Huizing et al. (2020), MDSA was concluded to be the most effective in practice.

References

Agarwal, A., Hiot, L. M., Joo, E. M., & Nghia, N. T. (2007). Rectilinear workspace partitioning for parallel coverage using multiple unmanned aerial vehicles. *Advanced Robotics, 21*, 105–120.

Başdere, M., & Bilge, Ü. (2014). Operational aircraft maintenance routing problem with remaining time consideration. *European Journal of Operational Research, 235*, 315–328.

Bektaş, T., Gouveia, L., & Santos, D. (2019). Revisiting the hamiltonian p-median problem: A new formulation on directed graphs and a branch-and-cut algorithm. *European Journal of Operational Research, 276*, 40–64.

Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological, 45*, 1232–1250.

Benati, S., & García, S. (2014). A mixed integer linear model for clustering with variable selection. *Computers & Operations Research, 43*, 280–285.

van den Berg, P., & Van Essen, T. (2019). Scheduling non-urgent patient transportation while maximizing emergency coverage. *Transportation Science*.

Bertsimas, D. J., & Van Ryzin, G. (1993). Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles. *Operations Research, 41*, 60–76.

Charikar, M., Guha, S., Tardos, É., & Shmoys, D. B. (2002). A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences, 65*, 129–149.

Cohn, A. M., & Barnhart, C. (2003). Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research, 51*, 387–396.

Colmenar, J. M., Hoff, A., Martí, R., & Duarte, A. (2018). Scatter search for the bi-criteria p-median p-dispersion problem. *Progress in Artificial Intelligence, 7*, 31–40.

Crainic, T. G., Perboli, G., Mancini, S., & Tadei, R. (2010). Two-echelon vehicle routing problem: a satellite location analysis. *Procedia-Social and Behavioral Sciences, 2*, 5944–5955.

Daskin, M. S., & Maass, K. L. (2015). The p-median problem. In *Location Science* (pp. 21–45). Springer.

Doitsidis, L., Weiss, S., Renzaglia, A., Achtelik, M. W., Kosmatopoulos, E., Siegwart, R., & Scaramuzza, D. (2012). Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision. *Autonomous Robots, 33*, 173–188.

Drexl, M., & Schneider, M. (2015). A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research, 241*, 283–308.

van Ee, M., & Sitters, R. (2014). Routing under uncertainty: The a priori traveling repairman problem. In *International Workshop on Approximation and Online Algorithms* (pp. 248–259). Springer.

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering, 57*, 1472–1483.

El Amrani, M., Benadada, Y., & Gendron, B. (2016). Generalization of capacitated p-median location problem: modeling and resolution. In *2016 3rd International Conference on Logistics Operations Management (GOL)* (pp. 1–6). IEEE.

Erdougan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review, 48*, 100–114.

Fontecha, J. E., Guaje, O. O., Duque, D., Akhavan-Tabatabaei, R., Rodríguez, J. P., & Medaglia, A. L. (2019). Combined maintenance and routing optimization for large-scale sewage cleaning. *Annals of Operations Research*, 1–34.

Garone, E., Naldi, R., Casavola, A., & Frazzoli, E. (2010). Cooperative mission planning for a class of carrier-vehicle systems. In *49th IEEE Conference on Decision and Control (CDC)* (pp. 1354–1359). IEEE.

Gaudioso, M., & Paletta, G. (1992). A heuristic for the periodic vehicle routing problem. *Transportation Science, 26*, 86–92.

Goodson, J. C., Ohlmann, J. W., & Thomas, B. W. (2013). Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. *Operations Research, 61*, 138–154.

Gopalan, R., & Talluri, K. T. (1998). The aircraft maintenance routing problem. *Operations Research, 46*, 260–271.

Groër, C., Golden, B., & Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & Service Operations Management, 11*, 630–643.

Haouari, M., Shao, S., & Sherali, H. D. (2012). A lifted compact formulation for the daily aircraft maintenance routing problem. *Transportation Science, 47*, 508–525.

Huizing, D., Schäfer, G., van der Mei, R. D., & Bhulai, S. (2020). The median routing problem for simultaneous planning of emergency response and non-emergency jobs. *European Journal of Operational Research, 285*, 712–727.

Huizing, D., Schäfer, G., van der Mei, R. D., & Bhulai, S. (2022). Distance-preserving graph sparsification with bounded loss for network problems (under review).

Ichoua, S., Gendreau, M., & Potvin, J.-Y. (2000). Diversion issues in real-time vehicle dispatching. *Transportation Science, 34*, 426–438.

Irawan, C. A., Ouelhadj, D., Jones, D., Stålhane, M., & Sperstad, I. B. (2017). Optimisation of maintenance routing and scheduling for offshore wind farms. *European Journal of Operational Research, 256*, 76–89.

Jackson, L. E., Rouskas, G. N., & Stallmann, M. F. (2007). The directional p-median problem: Definition, complexity, and algorithms. *European Journal of Operational Research, 179*, 1097–1108.

Jaillet, P., Bard, J. F., Huang, L., & Dror, M. (2002). Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science, 36*, 292–300.

Jain, K., & Vazirani, V. V. (2001). Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM (JACM), 48*, 274–296.

Joubert, J. W. (2007). *An integrated and intelligent metaheuristic for constrained vehicle routing* (Doctoral dissertation). University of Pretoria.

Kergosien, Y., Gendreau, M., Ruiz, A., & Soriano, P. (2014). Managing a fleet of ambulances to respond to emergency and transfer patient transportation demands. In *Proceedings of the International Conference on Health Care Systems Engineering* (pp. 303–315). Springer.

Kiechle, G., Doerner, K. F., Gendreau, M., & Hartl, R. F. (2009). Waiting strategies for regular and emergency patient transportation. In *Operations Research Proceedings* (pp. 271–276). Springer.

Kraster, D. (2020). Heuristic approaches for a multi-period job scheduling and emergency coverage problem. URL: www.ubvu.vu.nl/pub/index_oclc.cfm?SearchObjectId=8&objectid=109&ordering=1&openitem=197615.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research, 59*, 345–358.

López-Santana, E., Akhavan-Tabatabaei, R., Dieulle, L., Labadie, N., & Medaglia, A. L. (2016). On the combined maintenance and routing optimization problem. *Reliability Engineering & System Safety, 145*, 199–214.

Maróti, G., & Kroon, L. (2005). Maintenance routing for train units: the transition model. *Transportation Science, 39*, 518–525.

Palma-Behnke, R., Benavides, C., Lanas, F., Severino, B., Reyes, L., Llanos, J., & Sáez, D. (2013). A microgrid energy management system based on the rolling horizon strategy. *IEEE Transactions on Smart Grid, 4*, 996–1006.

Penicka, M., Strupchanska, A. K., & Bjørner, D. (2003). Train maintenance routing. In *FORMS'2003: Symposium on Formal Methods for Railway Operation and Control Systems*.

Pillac, V., Guéret, C., & Medaglia, A. L. (2012). An event-driven optimization framework for dynamic vehicle routing. *Decision Support Systems, 54*, 414–423.

Pillac, V., Gueret, C., & Medaglia, A. L. (2013). A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters, 7*, 1525–1535.

ProRail BV (2022). Hinder | ProRail reports. https://prestaties.prorail.nl/hinder/. Accessed 14 January 2022.

Psaraftis, H. N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science, 14*, 130–154.

ReVelle, C. S., & Swain, R. W. (1970). Central facilities location. *Geographical Analysis, 2*, 30–42.

Sarac, A., Batta, R., & Rump, C. M. (2006). A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research, 175*, 1850–1869.

Savelsbergh, M. W., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science, 29*, 17–29.

Shu, L., Wang, W., Lin, F., Liu, Z., & Zhou, J. (2013). A sweep coverage scheme based on vehicle routing problem. *Indonesian Journal of Electrical Engineering and Computer Science, 11*, 2029–2036.

Solomon, M. M. (1984). Vehicle routing and scheduling with time window constraints: Models and algorithms. Technical Report.

Talluri, K. T. (1998). The four-day aircraft maintenance routing problem. *Transportation Science, 32*, 43–53.

Toth, P., & Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications*. SIAM.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society, 35*, 797–809.

VRP-REP (2021). Problem variants. http://www.vrp-rep.org/variants.html. Accessed 7 April 2021.

Wang, Y., & Hussein, I. I. (2007). Cooperative vision-based multi-vehicle dynamic coverage control for underwater applications. In *IEEE International Conference on Control Applications* (pp. 82–87). IEEE.

Zhang, S., Ohlmann, J. W., & Thomas, B. W. (2014). A priori orienteering with time windows and stochastic wait times at customers. *European Journal of Operational Research, 239*, 70–79.