



Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.



A Large Neighbourhood Search Metaheuristic for the Contagious Disease Testing Problem

David Wolfinger^{a,b,*}, Margaretha Gansterer^c, Karl F. Doerner^{a,b}, Nikolas Popper^d

^a Department of Business Decisions and Analytics, University of Vienna, Oskar-Morgenstern-Platz 1, Vienna 1090, Austria

^b Data Science @ Uni Vienna, University of Vienna, Kolingasse 14–16, Vienna 1090, Austria

^c Department of Operations, Energy and Environmental Management, University of Klagenfurt, Universitätsstraße 65–67, Klagenfurt am Wörthersee 9020, Austria

^d Institute of Information Systems Engineering, TU Wien, Favoritenstraße 9–11, Vienna 1040, Austria

ARTICLE INFO

Article history:

Received 17 February 2021

Accepted 13 October 2021

Available online xxx

Keywords:

Routing

Facility location

Location routing

Large neighbourhood search

COVID-19

ABSTRACT

In late 2019 a new coronavirus disease (COVID-19) emerged, causing a global pandemic within only a few weeks. A crucial factor in the public health response to pandemics is achieving a short turnaround time between a potential case becoming known, specimen collection and availability of a test result. In this article we address a logistics problem that arises in the context of testing potential cases. We assume that specimens can be collected in two ways: either by means of a mobile test-team or by means of a stationary test-team in a test-centre. After the specimens have been collected they must be delivered to a laboratory in order to be analysed. The problem we address aims at deciding how many test-centres to open and where, how many mobile test-teams to use, which suspected cases to assign to a test-centre and which to visit with a mobile test-team, which specimen to assign to which laboratory, and planning the routes of the mobile test-teams. The objective is to minimise the total cost of opening test-centres and routing mobile test-teams. We introduce this new problem, which we call the contagious disease testing problem (CDTP), and present a mixed-integer linear-programming formulation for it. We propose a large neighbourhood search metaheuristic for solving the CDTP and present an extensive computational study to illustrate its performance. Furthermore, we give managerial insights regarding COVID-19 test logistics, derived from problem instances based on real world data.

© 2021 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

In late 2019 a new coronavirus disease (COVID-19) emerged in Wuhan, China. COVID-19 is a contagious respiratory and vascular disease that is caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). Within only a few weeks it spread all over the globe, causing the most recent pandemic in human history. As of October 2021, more than 235 million people have been infected worldwide, with over 4.8 million deaths attributed to the disease.¹

The world health organisation (WHO) identifies five phases (or stages) for an effective response to epidemics and pandemics: *anticipation, early detection, containment, control and mitigation*, and

elimination or eradication (WHO, 2018). A key strategy for affected countries to implement during the *containment* phase and the *control and mitigation* phase is the so-called *find, test, trace, isolate, and support* (FTTIS) strategy. Particularly in the COVID-19 pandemic this strategy is an essential component of the public health response (Matukas, Dhalla, & Laupacis, 2020; Rajan, Cylus, & McKee, 2020; SAGE, 2020). The idea is simple: anyone with symptoms is tested and, if positive, their (close) contacts are traced and advised or instructed to isolate (with support by the state if needed). Implementing the strategy such that it is effective is, however, anything but simple. It requires a complex system with many interlinking components (Rajan et al., 2020). One vital factor for the FTTIS strategy to be effective, is that test results are made available quickly (Matukas et al., 2020). That is, achieving a short turnaround time between a potential case becoming known, specimen collection and availability of a test result is crucial.

In this article we address a logistics problem that arises in the context of clarifying potential cases. Clarifying a potential case (also called a suspected case) means to collect a specimen from

* Corresponding author at: Department of Business Decisions and Analytics, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria.

E-mail addresses: david.wolfinger@univie.ac.at (D. Wolfinger), margaretha.gansterer@aau.at (M. Gansterer), karl.doerner@univie.ac.at (K.F. Doerner), nikolas.popper@tuwien.ac.at (N. Popper).

¹ <https://coronavirus.jhu.edu/map.html>. Accessed on October 5, 2021.

the potentially infected person – i.e., have the person take an infection detection test (IDT) – and analyse the specimen in a laboratory. In the case of COVID-19, for example, this would mean to take a SARS-CoV-2 polymerase chain reaction (PCR) test and analysing the swab in a laboratory. We focus on clarifying potential cases of infection that have been identified by an official (public health) institution (e.g., through contact-tracing or a screening process for people who show symptoms) and are officially instructed to take an IDT. We do not consider the testing of people who have not received such an instruction, like it is done for prevalence-testing or for obtaining a “certificate of non-infection”. We assume that there are two ways in which specimens can be collected. First, by means of a mobile test-team. Such a test-team visits the test-person at their home to take the IDT. And second, by means of a stationary test-team in a (walk-/drive-through) test-centre. In this case, the test-person must travel to the test-centre in order to take the IDT there. After the specimens have been collected (either by a mobile- or a stationary test-team), they must be delivered to a laboratory in order to be analysed. Using a combination of mobile test-teams and dedicated test-centres to meet the demand for IDTs is a common approach, which has been implemented in various countries during the COVID-19 pandemic; examples are Austria, Belgium, Canada, England, Finland, Israel and Spain.² One of the problems decision makers face when using this strategy, and the one we address in this article, is to decide how many test-centres to open and where, how many mobile test-teams to use, which suspected cases to assign to a test-centre and which to visit with a mobile test-team, which specimen to assign to which laboratory, and planning the routes of the mobile test-teams.

We call this problem the contagious disease testing problem (CDTP). In the CDTP the above mentioned decisions must be made while fulfilling a number of (practical) constraints. Most importantly, in order to guarantee a quick availability of test results, we assume a limit on the allowed amount of time between the occurrence of a potential case and the taking of the IDT (*time-to-test* limit), as well as on the time between the taking of the IDT and the attainment of its result (*time-to-result* limit). Furthermore, we consider that some people must be visited by a mobile test-team. This is for example the case when they are too sick to take the journey to a test-centre or when they have no access to a private means of transportation (potentially infected people taking public transportation must be avoided). Additional constraints concern the capacities of test-centres and laboratories, their opening hours, a limit on the time a person is expected to travel to a test centre (i.e., the test-centre's coverage range), and classical vehicle routing aspects for the mobile test-teams (such as time windows or start and end locations of the routes). We assume that test-centres in the CDTP are exclusively used by public health institutions (to clarify potential cases of infection) and are not available to the general public. That is, they do not accept bookings or drop-in patients, but only service cases that have been assigned by a public health institution. Such a situation is often the case at the beginning of an epidemic or pandemic when IDTs are (still) scarce and must be used efficiently (e.g., for contact-tracing). Furthermore, it helps preventing the spread of the disease, as it separates potentially infected people from people who are likely not infected (when taking the IDT). Opening a test-centre and using a vehicle (for the mobile test-team) involves paying fixed costs. The objective of the CDTP is to minimise the total cost of opening test-centres and routing vehicles.

Given the temporal characteristics of epidemics and pandemics – rapid onset but limited in time – the infrastructure used for test-

ing potential cases is often makeshift and intended to be only temporary. That is, test-centres typically consist of temporarily built test-stations in (large) public spaces such as convention centres, parking lots or town squares. As a consequence, and contrary to facilities in the classical location-routing problem (LRP), once built, test-centres can be opened and closed on a day-by-day basis. More precisely, once the infrastructure is in place, opening a test-centre requires virtually no setup activities and simply amounts to sending personnel on site. Thus, and because the main cost factor for operating a test-centre is the personnel, it can be decided each day anew whether to open a test-centre (i.e., staff it) or keep it closed for the day. Indeed, this is even true for individual test-stations within a test-centre. In the CDTP, we assume that the test-centres are already built and we only decide which test-centres to staff on a particular day. Hence, solving the CDTP on a daily basis is both possible and reasonable.

Note, the CDTP is not limited to COVID-19. Indeed, it arises in the context of any contagious disease for which an IDT exists. Furthermore, the CDTP is NP-hard. It reduces to the traveling salesperson problem (TSP) if the *time-to-test* and *time-to-result* constraints are (sufficiently) relaxed and there is no potential test-centre, only one uncapacitated laboratory and only one mobile test-team (i.e., vehicle). Since we know that the TSP is NP-hard (Garey & Johnson, 1979), so is the CDTP.

The contribution of our article is threefold. First, we introduce a new problem which we call the CDTP and present an arc-based mixed-integer linear programming formulation for it. Second, we propose a large neighbourhood search (LNS) metaheuristic for solving the CDTP and illustrate its performance with an extensive computational study. Since the CDTP is NP-hard, addressing it with a metaheuristic solution method allows us to obtain good results for real-world sized problem instances in a reasonable amount of time. Third, we analyse problem instances based on real-world data from two Austrian provinces, from which we draw and present managerial insights regarding COVID-19 test logistics. We find that significant cost savings can be achieved when the *time-to-test* and/or *time-to-result* limit is disregarded, albeit for the price of equally significant increases in the clarification times.

In this article we address a deterministic and static version of the CDTP. Clearly, the real-world problem is stochastic and dynamic in nature. Potential cases of infection occur dynamically throughout the day, according to some spatial-temporal probability distribution. Nevertheless, given the novelty of the problem, solving the deterministic and static variant of the problem is still an important contribution as it provides us with valuable (managerial) insights.

The remainder of this article is structured as follows. In Section 2, we discuss the related literature. In Section 3, we give a formal description of the CDTP and present the mathematical formulation for it. Section 4 is devoted to the LNS. The computational study and the managerial insights are presented in Section 5. Concluding remarks are presented in Section 6.

2. Literature Review

The CDTP is closely related to the location-or-routing problem (LoRP). Like in the CDTP, the goal in the LoRP is to cover each customer either by an open facility or by a vehicle route, and the objective is to minimise the total cost of opening facilities and routing vehicles. Contrary to the CDTP, in the LoRP all customers located within the coverage range of an open facility are considered to be covered (by this facility), while in the CDTP being located within the coverage range of an open facility merely grants the possibility of being covered by this facility. Furthermore, in the LoRP facilities simultaneously serve as vehicle depots and each route must depart from an open facility, whereas in the CDTP vehi-

² <https://www.covid19healthsystem.org/mainpage.aspx> Accessed on July 7, 2021. For detailed information, choose the desired country and go to subchapter 1.5 *Testing of the Policy responses*.

cle depots are separate from the facilities. Lastly, customers in the LoRP have a positive demand, and vehicles are subject to capacity constraints. In the CDTP, each test-person supplies only one IDT, and due to their small size vehicle capacities can be neglected. The LoRP was only recently introduced by Arslan (2020). They consider uncapacitated facilities and a maximum route length constraint. To solve the problem, they propose a branch-and-price algorithm; which is able to solve to optimality instances with up to 40 customers and 5 facilities within 3 hours of run time.

The LoRP conjoins two well-known optimisation problems which also share similarities with the CDTP: the LRP and the covering location problem (CLP). In the LRP, facilities do not cover customers, but rather, every customer must be visited by a vehicle departing from an open facility. The objective is to minimise the total cost of opening facilities and routing vehicles. In the CLP, on the other hand, no vehicles are used, which means that customers must be covered exclusively by opening facilities. In this problem, typically, one of two objectives is pursued; either, trying to minimise the location cost while satisfying all demand (*set covering location problem*), or, trying to maximise the amount of covered demand with a given fixed number of facilities (*maximal covering location problem*). Both problems are well studied in the scientific literature. We therefore discuss in this section only the works most closely related to our study, and refer the interested reader to the recent surveys of Prodhon & Prins (2014), Drexel & Schneider (2015) and Schneider & Drexel (2017) for a detailed review on the LRP (and its many variants), as well as, the survey of Farahani, Asgari, Heidari, Hosseiniinia, & Goh (2012) for a thorough discussion of the CLP and the recent surveys of Ahmadi-Javid, Seyed, & Syam (2017) and Güneş, Melo, & Nickel (2019) for a detailed discussion of healthcare facility location problems in general.

Martínez-Reyes, Quintero-Araújo, & Solano-Charris (2020) investigate for the city of Bogotá, Colombia, locating distribution centres and designing corresponding vehicle routes to supply intensive care units with personal protective equipment in the context of the COVID-19 pandemic. They formulate their problem as a capacitated LRP (CLRP) – capacity constraints on facilities and vehicles – with stochastic demands. To solve this problem they propose a hybrid approach that combines an iterated local search metaheuristic with Monte Carlo simulation.

Lee, Chen, Pietz, & Benecke (2009) develop an emergency-response decision-support tool which aids public-health emergency coordinators to set up efficient and large-scale dispensing of critical medical countermeasures (i.e., drugs, vaccines, or therapeutics) in the event of public-health emergencies, such as bioterrorist attacks or pandemics. In order to determine the locations for the point-of-dispensing facilities, they solve a set covering location problem by means of a heuristic approach combining features of a genetic algorithm and an adaptive greedy search.

Another related problem class is the vehicle routing problem with intra-route facilities (VRP-IF). Intra-route facilities are different from regular customers in the sense that visiting them is optional and typically depends on the state of the vehicle with respect to the load and/or fuel (battery charging) level. In the CDTP, the laboratories are intra-route facilities. They can be visited whenever needed, however, one visit (immediately before returning to the depot) is required in each vehicle route. Schneider, Stenger, & Hof (2015) address a generic version of the VRP-IF, abstracting away the actual purpose of the intermediate stop (i.e., replenishment of goods, refuelling/recharging of the vehicle or disposal of load). They develop an adaptive variable neighbourhood search (VNS), combining features of VNS and adaptive large neighbourhood search (LNS) (ALNS), to solve their problem. Schiffer & Walther (2018) study vehicle routing with intra-route facilities in combination with the LRP. In this newly introduced problem, the locations of the depot facilities are known, instead, the locations

of facilities for intermediate stops need to be determined in order to keep vehicles operational (for example, recharging stations for electric vehicles). They propose an ALNS algorithm to solve this problem. For an in-depth discussion of the VRP-IF, including also in combination with the LRP, we refer to the recent review by Schiffer, Schneider, Walther, & Laporte (2019).

The *time-to-test* and *time-to-result* limits are crucial aspects of the CDTP. The *time-to-test* limit can be expressed in the form of a time window for each suspected case, within which the IDT must take place. Suspected cases that are tested in test-centres must be scheduled accordingly and vehicles must adhere to these time windows when visiting a suspected case. Likewise, the *time-to-result* limit requires a scheduling of cases in laboratories. To the best of our knowledge, there are no studies that combine facility location, vehicle routing and scheduling decisions. Our paper is a first step towards closing this research gap.

3. Problem Description

In this section, we give a formal description of the CDTP and present an arc-based mixed-integer linear programming formulation for it.

Let P be the set of suspected cases which need to be clarified. With each suspected case $p \in P$ is associated a time window $[a_p, b_p]$. The start of the time window, a_p , represents the point in time when case p becomes known (for example, by a call to a hot-line). The end of the time window, b_p , is set such that $b_p = a_p + \psi^{\text{test}}$, where ψ^{test} denotes the *time-to-test* limit (i.e., the maximum allowed amount of time between the occurrence of a case and the time an IDT is taken). Let $\Gamma_p \in \{0, 1\}$ indicate for each suspected case p whether it must be clarified by a mobile test-team ($\Gamma_p = 1$) or can also be clarified in a test-centre ($\Gamma_p = 0$).

Let K be a homogeneous fleet of uncapacitated vehicles (representing the mobile test-teams). Using a vehicle causes fixed costs, denoted with f^V . Each vehicle $k \in K$ is associated with a depot at which it must start and end its route. For a more convenient representation in the mathematical formulation we split up the depot of each vehicle k into an origin depot k^+ and a destination depot k^- . The set of depots is denoted with $N = \bigcup_{k \in K} \{k^+, k^-\}$. A time window $[a_i, b_i]$ is also associated with each depot $i \in N$.

The set of test-centres is denoted with T . Each test-centre $t \in T$ is associated with a number of test-stations q_t which are operated in parallel, as well as a time window $[a_t, b_t]$ representing the opening time (a_t) and closing time (b_t) of the test-centre. Opening a test-centre causes fixed costs, denoted with f_t^T . We denote with τ the test-centre's coverage range, representing the maximum amount of time a person is expected to travel to a test-centre. Any person living further away from the closest open test-centre than τ must be visited by a mobile test-team.

Let L be the set of laboratories. Laboratories use dedicated machines to analyse and evaluate specimens. Processing a batch of specimens is called performing an evaluation run. With each laboratory $l \in L$ is associated a set of evaluation runs E_l . We denote with ε_{le} the start time of evaluation run $e \in E_l$ of laboratory $l \in L$. The number of specimens which can be processed in one evaluation run differs between laboratories and is denoted with r_l , where $l \in L$. The duration of an evaluation run is denoted with v and assumed to be the same for all laboratories. A time window $[a_l, b_l]$ is also associated with each laboratory $l \in L$.

At least once, possibly several times, throughout the opening hours of a test-centre the so far collected specimens are brought to an associated laboratory for evaluation. We denote with $\eta_t \in L$ the associated laboratory of test-centre $t \in T$, and with H_t the set of time points at which the so far collected specimens are transported from t to η_t . We do not take these transports into account when generating the vehicle routes, but rather assume that they

are performed by additional personnel and vehicles. The specimens obtained by a mobile test-team, on the other hand, must be delivered to a laboratory by the test-team itself and are therefore part of the (routing) problem.

Note that the set $H_t = \{h_1, h_2, \dots, h_n\}$ together with the opening time a_t naturally forms a set of time slots for each test-centre t . In this set of time slots, which we denote with \mathcal{H}_t , the first time slot starts at a_t and ends at h_1 , the second time slot starts at h_1 and ends at h_2 , the third time slot starts at h_2 and ends at h_3 , and so on, until the n th (and last) time slot which starts at h_{n-1} and ends at h_n . We assume that $h_n = b_t$, which is reasonable, since otherwise specimens would stay in the test-centre overnight. We do not generate an explicit schedule of potential cases (i.e., a sequence of suspected cases) within a single time slot. Each time slot $h \in \mathcal{H}_t$ of a test-centre $t \in T$ is therefore associated with a single time point $\omega_{th} = \xi_{th} + d_{th}/2$ at which the IDT is assumed to be taken; where ξ_{th} and d_{th} denote the start time and duration of time slot h of test-centre t , respectively.

Let ψ^{result} denote the *time-to-result* limit, i.e., the maximum allowed amount of time between taking an IDT and attaining its result. We assume that the result of an IDT is known immediately after the evaluation run which contains the corresponding specimen has finished. Therefore, we use the finishing times of the evaluation runs to calculate the elapsed time between taking an IDT and attaining its result.

The problem is defined on a directed graph $G = (V, A)$, where $V = N \cup P \cup T \cup L$ is the vertex set, and $A = \bigcup_{k \in K} A^k$ is the arc set, with $A^k = \{(i, j) \mid i \in \{k^+\} \cup P \cup L, j \in \{k^-\} \cup P \cup L, (j \neq k^- \vee i \in L)\}$ representing the feasible movements of each vehicle $k \in K$. With each arc $a \in A$ is associated a travel cost c_a and a travel time λ_a . We assume that the triangular inequality holds for both. Let σ^T and σ^P denote the time it takes to perform an IDT at a test-centre and at a test-person's home, respectively. Furthermore, let σ^L denote the time it takes for a mobile test-team to unload the obtained specimens at a laboratory. Then, with each location $i \in V \setminus T$ is associated a service time σ_i , where $\sigma_i = 0$ for all $i \in N$, $\sigma_i = \sigma^P$ for all $i \in P$ and $\sigma_i = \sigma^L$ for all $i \in L$.

For each suspected case, the IDT must be taken within its time window. This means that either the case must be assigned to a test-centre time slot with an associated time point of taking the IDT (ω_{th}) that lies within the time window, or, a mobile test-team must visit the corresponding location within the time window. A vehicle can arrive prior to the beginning of the time window, but must await its beginning before starting the service. Furthermore, a vehicle may visit several laboratories on its route, or the same laboratory several times, or both. We assume that at each visit all specimens that are currently on the vehicle are unloaded.

The objective of the CDTP is to find a feasible assignment of suspected cases to test-centre time slots, a set of feasible vehicle routes which serves all cases that are not assigned to a test-centre, and a feasible assignment of suspected cases to laboratory evaluation runs, such that all suspected cases are clarified and the sum of fixed costs and travel costs is minimised.

An assignment of suspected cases to time slots of test-centres is feasible if cases are only assigned to open test-centres, if the assigned cases are located within the test-centres' coverage range, if for each assigned case the time point of taking the IDT lies within its time window, and if the test-centre's capacity in each time slot (which depends on σ^T , q_t , and d_{th}) is not exceeded. A route for a vehicle corresponds to a path from its origin depot to its destination depot in the graph G . The path is feasible if the time windows at all visited locations are respected and the capacities of the visited laboratories are not exceeded. An assignment of suspected cases to laboratory evaluation runs is feasible if for each assigned case the elapsed time between taking the IDT and attaining its result is not larger than ψ^{result} and the capacity of each evaluation

run is respected. A solution is feasible if all assignments and vehicle routes are feasible.

3.1. Mixed-Integer Linear Programming Formulation

In the CDTP laboratories may be visited more than once by the same vehicle. In order to work on a graph where each vertex is visited at most once, we need to use location-copies; one copy for each potential visit by the same vehicle. We assume that we are given an explicit number of copies m_l for each laboratory $l \in L$. m_l represents the maximum number of allowed visits by the same vehicle at laboratory l . Note, however, that the value of m_l can be set arbitrarily large such that it does not impose any limitation on the construction of vehicle routes. Let \tilde{L}_l be the set of m_l copies of laboratory $l \in L$. Then, we replace the set L in graph G with $\tilde{L} = \bigcup_{l \in L} \tilde{L}_l$ for the mathematical formulation.

The following mathematical variables are used to formulate the CDTP:

$$\begin{aligned} s_i^k & \text{ start time of service of vehicle } k \in K \text{ at node } i \in V \\ x_a^k & \begin{cases} 1 \text{ iff vehicle } k \in K \text{ traverses arc } a \in A \\ 0 \text{ otherwise} \end{cases} \\ u_{ap}^k & \begin{cases} 1 \text{ iff the specimen of case } p \in P \text{ is transported over} \\ \text{arc } a \in A \text{ by vehicle } k \in K \\ 0 \text{ otherwise} \end{cases} \\ g_p^{le} & \begin{cases} 1 \text{ iff suspected case } p \in P \text{ is assigned to evaluation run} \\ e \in E_l \text{ of laboratory } l \in L \\ 0 \text{ otherwise} \end{cases} \\ y_t & \begin{cases} 1 \text{ iff test-centre } t \in T \text{ is opened} \\ 0 \text{ otherwise} \end{cases} \\ z_p^{th} & \begin{cases} 1 \text{ iff suspected case } p \in P \text{ is assigned to time slot} \\ h \in \mathcal{H}_t \text{ of test-centre } t \in T \\ 0 \text{ otherwise} \end{cases} \end{aligned}$$

To abbreviate notation, for any subset $S \subseteq V$, we write $\delta_k^+(S) = \{(i, j) \in A^k \mid i \in S, j \notin S\}$ and $\delta_k^-(S) = \{(i, j) \in A^k \mid i \notin S, j \in S\}$ for the set of arcs belonging to vehicle k which leave, respectively enter, set S . Also, for any subset $F \subseteq A^k$, we $x^k(F) = \sum_{a \in F} x_a^k$ and $u_p^k(F) = \sum_{a \in F} u_{ap}^k$.

Then, the CDTP can be formulated as the following mixed-integer linear program.

$$\min \sum_{k \in K} \sum_{j \in V \setminus \{k^-\}} f^V x_{(k^+, j)}^k + \sum_{k \in K} \sum_{a \in A^k} c_a x_a^k + \sum_{t \in T} f_t^T y_t \quad (1)$$

subject to

$$\sum_{k \in K} x^k(\delta_k^+(p)) + \sum_{t \in T} \sum_{h \in \mathcal{H}_t} z_p^{th} = 1 \quad \forall p \in P \quad (2)$$

$$\sum_{k \in K} x^k(\delta_k^+(p)) \geq \Gamma_p \quad \forall p \in P \quad (3)$$

$$\sum_{p \in P} \sum_{h \in \mathcal{H}_t} z_p^{th} \leq |P| y_t \quad \forall t \in T \quad (4)$$

$$z_p^{th} = 0 \quad \forall p \in P, t \in T : \lambda_{(p, t)} > \tau, h \in \mathcal{H}_t \quad (5)$$

$$z_p^{th} = 0 \quad \forall p \in P, t \in T, h \in \mathcal{H}_t : \omega_{th} < a_p \vee \omega_{th} > b_p \quad (6)$$

$$\sum_{p \in P} z_p^{th} \leq \left\lfloor \frac{d_{th}}{\sigma^T} \right\rfloor q_t \quad \forall t \in T, h \in \mathcal{H}_t \quad (7)$$

$$\sum_{l \in L} \sum_{e \in E_l} g_p^{le} = 1 \quad \forall p \in P \quad (8)$$

$$\sum_{h \in \mathcal{H}_t} (\xi_{th} + d_{th} + \lambda_{(t, \eta_t)}) z_p^{th} \leq \sum_{e \in E_{\eta_t}} \varepsilon_{\eta_t e} g_p^{le} \quad \forall p \in P, t \in T \quad (9)$$

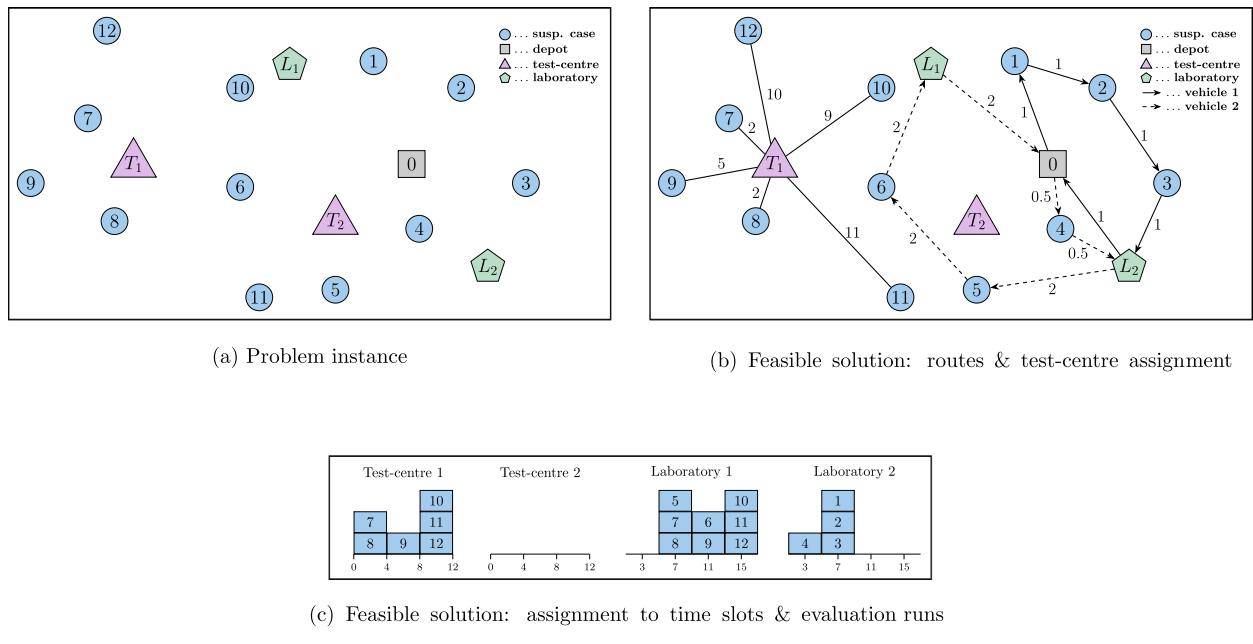


Fig. 1. Problem instance (1a) together with a feasible solution (1b & 1c).

$$s_i^k + \sigma_i - M(1 - u_p^k(\delta_k^-(i))) \leq \sum_{e \in E_l} \varepsilon_{le} g_p^{le} \quad \forall p \in P, l \in L, i \in \bar{L}_l, k \in K \quad (10)$$

$$\sum_{l \in L} \sum_{e \in E_l} (\varepsilon_{le} + \nu) g_p^{le} - \sum_{t \in T} \sum_{h \in H_t} \omega_{th} z_p^{th} - \sum_{k \in K} s_p^k \leq \psi^{\text{result}} \quad \forall p \in P \quad (11)$$

$$\sum_{p \in P} g_p^{le} \leq r_l \quad \forall l \in L, e \in E_l \quad (12)$$

$$x^k(\delta_k^+(k^+)) = 1 \quad \forall k \in K \quad (13)$$

$$x^k(\delta_k^-(k^-)) = 1 \quad \forall k \in K \quad (14)$$

$$x^k(\delta_k^-(i)) - x^k(\delta_k^+(i)) = 0 \quad \forall i \in V \setminus N, k \in K \quad (15)$$

$$s_i^k + \sigma_i + \lambda_{(i,j)} - M(1 - x_{(i,j)}^k) \leq s_j^k \quad \forall k \in K, (i, j) \in A^k \quad (16)$$

$$x^k(\delta_k^+(i)) a_i \leq s_i^k \quad \forall k \in K, i \in V \setminus \{k^-\} \quad (17)$$

$$s_i^k \leq x^k(\delta_k^-(i)) b_i \quad \forall k \in K, i \in V \setminus \{k^+\} \quad (18)$$

$$x_a^k \leq u_{ap}^k \quad \forall p \in P, k \in K, a \in \delta_k^+(p) \quad (19)$$

$$u_p^k(\delta_k^-(i)) - u_p^k(\delta_k^+(i)) = 0 \quad \forall p \in P, i \in P \setminus \{p\}, k \in K \quad (20)$$

$$\sum_{p \in P} u_{ap}^k \leq |P| x_a^k \quad \forall k \in K, a \in A^k \quad (21)$$

$$\sum_{k \in K} \sum_{l \in L} \sum_{i \in \bar{L}_l} \sum_{p \in P} u_p^k(\delta_k^-(i)) = 0 \quad (22)$$

$$g_p^{le}, u_{ap}^k, x_a^k, y_t, z_p^{th} \in \{0, 1\}; s_i^k \geq 0 \quad (23)$$

$$\forall p \in P, l \in L, e \in E_l, t \in T, h \in H_t, k \in K, a \in A^k, i \in V$$

The objective function (1) minimises the total cost, where the first term represents the fixed costs for using vehicles, the second term represents the routing costs and the third term represents the cost for opening test-centres. Constraints (2) guarantee that every suspected case is covered, either by a mobile test-team or by a test-centre. Constraints (3) ensure that all suspected cases

which must be tested by a mobile test-team are visited by a vehicle. Constraints (4) make sure that cases are only assigned to open test-centres, while constraints (5) forbid that cases are assigned to test-centres which are located further away than the test-centre's coverage range. Constraints (6) ensure that the time point of taking an IDT in a test-centre lies within the case's time window. Constraints (7) guarantee that the capacity of a time slot in a test-centre is not exceeded. Constraints (8) ensure that every suspected case is evaluated in a laboratory. Constraints (9) make sure that the specimens of suspected cases which are assigned to a test-centre are evaluated in the associated laboratory of the test-centre. Furthermore, they ensure that the specimens are assigned to an evaluation run which can feasibly be reached in time. Similarly, constraints (10) ensure that cases which are covered by a mobile test-team are assigned to an evaluation run which can feasibly be reached in time by the vehicle. Note, specimens can be part of an evaluation run only if they arrive before the start time of the evaluation run (not during it). Constraints (11) guarantee that the elapsed time between taking the IDT and attaining its result does not exceed the maximum allowed time ψ^{result} . Constraints (12) make sure that the capacity of an evaluation run is not exceeded. Constraints (13) and (14) guarantee that every vehicle starts and ends its tour at the depot. Constraints (15) regulate the flow conservation of vehicles. Time consistency is ensured by constraints (16), where M is a sufficiently large value. Constraints (17) and (18) guarantee that the time windows are respected. Constraints (19) make sure that a vehicle leaving a test-person's home carries the corresponding specimen. Constraints (20) regulate the flow conservation of specimens which were obtained by a mobile test-team, while constraints (21) ensure that all specimen movements are covered by a vehicle flow. Constraints (22) forbid that specimens are transported out of a laboratory. Finally, constraints (23) set the domain of the decision variables.

Fig. 1 illustrates a problem instance of the CDTP as formulated above (Fig. 1a) together with a feasible solution (Fig. 1b and 1c). The instance consists of twelve suspected cases (1, ..., 12), one depot (0), two test-centres (T_1, T_2) and two laboratories (L_1, L_2). For both test-centres the collected specimens are transported to the associated laboratories (laboratory 1 for test-centre 1, laboratory

2 for test-centre 2) at time points 4, 8 and 12, resulting in three time slots. In both laboratories evaluation runs are started at time points 3, 7, 11 and 15. Each time slot and each evaluation run has a capacity of three suspected cases. To keep it simple, the time of occurrence is equal to 0 for all cases. Likewise, the service time for vehicles is also equal to 0 at all locations.

Fig. 1b shows a feasible solution which uses two vehicles (mobile test-teams) and one test-centre. The numbers on the arcs and edges are the travel times. Cases 1, 2 and 3 are visited by vehicle 1 (solid line) and their specimens are delivered to laboratory 2. Cases 4, 5 and 6 are visited by vehicle 2 (dashed line), where the specimen of case 4 is delivered to laboratory 2 and the specimens of cases 5 and 6 are delivered to laboratory 1. Cases 7–12 are assigned to test-centre 1. Test-centre 2 is not opened.

Fig. 1c shows the corresponding assignment of the suspected cases to the time slots and evaluation runs in the test-centres and laboratories, respectively. Cases 7 and 8 are assigned to the first time slot in test-centre 1 (which starts at time point 0 and ends at time point 4). The time point of taking the IDT is therefore equal to 2 for both cases. Furthermore, their specimens are transported to laboratory 1 at time point 4, where they are assigned to the evaluation run which starts at time point 7. Case 9 is assigned to the second time slot in test-centre 1 and to the evaluation run that starts at time point 11 in laboratory 1. Cases 10, 11 and 12 are assigned to the third time slot in test-centre 1 and to the fourth evaluation run in laboratory 1. Given that vehicle 1 arrives at laboratory 2 at time point 4, cases 1, 2 and 3 are assigned to the evaluation run which starts at time point 7. The time points of taking the IDT for cases 1, 2 and 3 are 1, 2, and 3, respectively. Case 4 is evaluated in laboratory 2 in the evaluation run which starts at time point 3. Cases 5 and 6 are evaluated in laboratory 1. Due to the capacity restriction of at most three cases per evaluation run, only case 5 can still be assigned to the evaluation run which starts at time point 7. Case 6 is assigned to the evaluation run which starts at time point 11.

4. LNS for the CDTP

In this section, we describe the LNS metaheuristic developed for the CDTP. The main idea of LNS is to obtain a good solution by iteratively destroying and repairing it in different ways. The method was originally introduced by Shaw (1997) for the capacitated VRP (CVRP) with time windows. For a detailed description of LNS, we refer to Pisinger & Ropke (2019).

Algorithm 1 gives the pseudocode for the proposed LNS. It takes as input an initial solution s (which does not need to be feasible). In each iteration, one destroy heuristic and one repair heuristic are randomly chosen (line 3). Using the chosen heuristics, the current solution s is first destroyed, by removing a number of suspected cases, and then repaired again, by reinserting the removed cases (line 4). If the newly found solution s' is within θ percent of the current best solution s^* , it is subjected to local search (line 6). The rest of the algorithm deals with updating the best solution (line 7) and deciding whether the newly found solution should be accepted as the new incumbent solution (line 9). The algorithm runs until a certain stopping criterion is met, which is checked at the beginning of each new iteration (line 2), and in the end returns the best solution found (line 11).

In the following subsections, we review the main components of the proposed LNS.

4.1. Destruction Heuristics

Each destruction heuristic removes n suspected cases from the solution. As is common in LNS algorithms, n depends on the size of the problem instance and is determined in each iteration anew

Algorithm 1: LNS.

```

Input: initial solution  $s$ 
1  $s^* \leftarrow s$ 
2 while stopping criterion not met do
3   randomly select a destroy and a repair heuristic
4    $s' \leftarrow \text{Repair}(\text{Destroy}(s))$ 
5   if  $s' < (1 + \theta) \cdot s^*$  then
6      $s' \leftarrow \text{LocalSearch}(s')$ 
7   if  $s' < s^*$  then
8      $s^* \leftarrow s'$ 
9   if accept( $s', s^*$ ) then
10     $s \leftarrow s'$ 
11 return  $s^*$ 
```

(cf. Ropke & Pisinger, 2006; Wolfinger, 2021). Specifically, it is randomly drawn from the interval $[\alpha \cdot |P|, \beta \cdot |P|] \cap \mathbb{N}$ in each iteration, where α and β are input parameters which represent the minimum and maximum percentage of suspected cases to be removed. Removing a suspected case from the solution means to remove it from the vehicle route or the test-centre time slot it is currently covered by, as well as removing it from the laboratory evaluation run it is currently assigned to. We implemented six destruction heuristics; three well-known general purpose heuristics from the literature, one heuristic which specifically addresses the vehicle routing aspect in our problem (*smallest route removal*), and two heuristics that specifically address the facility location aspect in our problem (*least utilized test-centre removal* and *test-centre opening removal*). Where applicable, the destruction heuristics do not take into account whether the suspected case to be removed is currently assigned to a test-centre or visited by a vehicle.

1. *Random removal.* This heuristic removes n randomly selected suspected cases from the current solution, whereby all cases have equal probability of being selected.
2. *Worst removal.* First, for each suspected case the cost savings produced by its removal are computed. Note, that cases which are assigned to test-centres do not directly cause costs. Thus, their cost savings are equal to zero. Then, n cases are selected at random to be removed. The probability of a case to be selected increases with the magnitude of the cost savings. See Ropke & Pisinger (2006) for a detailed description of this heuristic. We use the same method as proposed in their article for the random selection.
3. *Related removal.* The main idea of this heuristic is to remove related, respectively similar, suspected cases. This heuristic was originally introduced by Shaw (1998) for the CVRP. We adopt the basic idea and adapt the heuristic to our problem. To define the similarity of two suspected cases $p, p' \in P$ we use a relatedness measure $R(p, p')$. It consists of three terms – whereby each term is weighted using the weights ρ_1, ρ_2 and ρ_3 , respectively – and is computed as

$$R(p, p') = \rho_1(c_{(p, p')}/c^{\max}) + \rho_2((|\Lambda_p - \Lambda_{p'}|)/b^{\max}) + \rho_3(|\Gamma_p - \Gamma_{p'}|),$$

where $c^{\max} = \max_{p, p' \in P}\{c_{(p, p')}\}$, $b^{\max} = \max_{p \in P}\{b_p\}$ and Λ_p represent the longest arc in the graph, the latest end of a time window of a suspected case and the time point of taking the IDT of case p , respectively. The relatedness measure, thus, depends on the distance between the locations of the two cases (first term), the difference between the time of taking the IDT (second term) and the difference in whether the cases must be visited by a mobile test-team (third term). The lower $R(p, p')$ is, the more related are the two suspected cases. Notice that

due to the normalization of the first two terms $0 \leq R(p, p') \leq \sum_{k=1}^3 \rho_k$. Like in the *worst removal* heuristic, n suspected cases are selected at random to be removed, whereby the probability of being selected increases with lower values of $R(p, p')$. We use the same values for the weights ρ_1 , ρ_2 , and ρ_3 as proposed in [Ropke & Pisinger \(2006\)](#) (where they are called φ , χ , and ω , respectively).

4. *Smallest route removal*. This heuristic aims at reducing the number of used vehicles. It selects the vehicle which visits the smallest number of suspected cases and removes all the cases that are visited by the selected vehicle. The idea is that these cases might be reinserted in the remaining vehicles, thereby creating new, perhaps better, solutions (which use fewer vehicles). In case of a tie, the first encountered vehicle is selected for removal.
5. *Least utilised test-centre removal*. This heuristic follows the same idea as the *smallest route removal* heuristic. But, instead of considering vehicles it considers test-centres. That is, it first selects the test-centre which has the smallest number of suspected cases assigned to it. Then, it removes all cases which are assigned to the selected test-centre. Again, in case of a tie, the first encountered test-centre is selected for removal.
6. *Test-centre opening removal*. This heuristic is inspired by the *satellite opening* heuristic proposed by [Hemmelmayr, Cordeau, & Crainic \(2012\)](#). The heuristic first selects a test-centre among those that are currently closed and opens it. The selection process is such that with a probability of γ the test-centre which covers the most suspected cases is selected, and with a probability of $(1 - \gamma)$ a random test-centre is selected. Then, n suspected cases are randomly selected to be removed, whereby a smaller distance to the opened test-centre corresponds to a larger probability of being selected.

4.2. Repair Heuristics

Each insertion heuristic iteratively reinserts the previously removed suspected cases into the solution until all cases are covered or none can be inserted any more. An insertion of a suspected case into the solution consists of inserting it into either a vehicle route or a test-centre time slot, and simultaneously inserting it into a laboratory evaluation run. We implemented three repair heuristics; two well-known general purpose heuristics from the literature, which are only capable of handling insertions into vehicle routes, and one heuristic which is designed to handle only insertions into test-centres (i.e., assignments to time slots, *time-greedy insertion into test-centre*). Because any insertion into a vehicle route or a test-centre requires the simultaneous assignment of the considered case to an evaluation run in a laboratory, all three heuristics are capable of handling insertions into laboratories. In all heuristics, suspected cases are always assigned to the earliest feasible evaluation run (i.e., they are inserted in a time-greedy manner). Furthermore, in order to speed up the insertion heuristics, we store (and continuously update) for each suspected case the best position in each vehicle route for inserting it, whereby *best position* refers to the position at which the insertion cost is the smallest. We denote with U the set of uncovered suspected cases.

1. *Random best insertion*. In each iteration, this heuristic inserts a randomly selected uncovered suspected case at its over all best position. [Algorithm 2](#) gives the pseudocode for this heuristic. In each iteration, the heuristic first randomly selects an uncovered case p from the set U (line 7) and removes it (line 8). Then, it evaluates the insertion of p into every vehicle that is currently used in the solution (lines 9–16). For each of these vehicles, the heuristic first evaluates the insertion of p by itself (line 10). In case p cannot be feasibly inserted by itself, the heuristic also

evaluates inserting it together with an additional visit to a laboratory (line 12). Furthermore, for all vehicles that are currently not used in the solution, an insertion of p combined with a laboratory visit is evaluated as well (lines 17–20). Finally, if a feasible insertion was found, p is inserted at its best position (possibly together with an additional laboratory visit) and assigned to the best evaluation run (lines 21–25). The algorithm runs until the set U is empty (line 6).

Algorithm 2: Random best insertion.

```

Input: destroyed solution  $S$ ; set of uncovered cases  $U$ 
1 best cost  $c^* \leftarrow \infty$ 
2 best position of case  $y_p^* \leftarrow -1$ 
3 best laboratory  $l^* \leftarrow -1$ 
4 best position of laboratory  $y_l^* \leftarrow -1$ 
5 best evaluation run  $e^* \leftarrow -1$ 
6 while  $U \neq \emptyset$  do
7    $p \leftarrow$  randomly selected suspected case from set  $U$ 
8    $U \leftarrow U \setminus \{p\}$ 
9   foreach vehicle  $k \in K$  used in  $S$  do
10    |  $c, y_p, l, e \leftarrow$  bestPosition( $p, k, S$ )
11    | if no feasible insertion position found then
12    |   |  $c, y_p, l, e, y_l \leftarrow$  bestPositionWithLab( $p, k, S$ )
13    | if feasible insertion pos. found and  $c < c^*$  then
14    |   |  $c^* \leftarrow c$ ;  $y_p^* \leftarrow y_p$ ;  $l^* \leftarrow l$ ;  $e^* \leftarrow e$ 
15    |   | if insertion includes laboratory visit then
16    |     |  $y_l^* \leftarrow y_l$ 
17   foreach vehicle  $k \in K$  not used in  $S$  do
18    |  $c, y_p, l, e, y_l \leftarrow$  bestPositionWithLab( $p, k, S$ )
19    | if feasible insertion pos. found and  $c < c^*$  then
20    |   |  $c^* \leftarrow c$ ;  $y_p^* \leftarrow y_p$ ;  $l^* \leftarrow l$ ;  $e^* \leftarrow e$ ;  $y_l^* \leftarrow y_l$ 
21    | if feasible insertion position found then
22    |   | insert  $p$  at position  $y_p^*$ 
23    |   | insert  $p$  into evaluation run  $e^*$  of laboratory  $l^*$ 
24    |   | if insertion includes laboratory visit then
25    |     | insert  $l^*$  at position  $y_l^*$ 

```

2. *Regret-k insertion*. In this heuristic, the decision of which suspected case to insert is based on the concept of regret. Let Δf_p^j be the insertion cost of the suspected case $p \in U$ (possibly together with an additional laboratory visit) in the j th best route at its best position. Then, in each iteration, the case $p^* = \operatorname{argmax}_{p \in U} (\sum_{j=2}^k \Delta f_p^j - \Delta f_p^1)$ is inserted into the solution at its best position. A detailed description of this heuristic is given in [Ropke & Pisinger \(2006\)](#). We use three regret-k heuristics with $k \in \{2, 3, 4\}$ in our implementation.

3. *Time-greedy insertion into test-centre*. In each iteration, this heuristic inserts a suspected case at its closest feasible test-centre into the earliest feasible time slot. For each case that can be assigned to a test-centre (i.e., $\Gamma = 0$), first, all test-centres are listed in ascending order with respect to their distance to the suspected case. Then, starting from the front, the heuristic traverses this list and evaluates the insertion of the suspected case into the currently considered test-centre; whereby the time slots of the test-centre are evaluated in chronological order. The case is inserted into the first test-centre where an insertion is feasible. [Algorithm 3](#) gives the pseudocode for this heuristic.

Algorithm 3: Time-greedy insertion into test-centre.

```

Input: destroyed solution  $S$ ; set  $U$ 
1 foreach uncovered case  $p \in U$  where  $\Gamma_p = 0$  do
2   List  $\Omega \leftarrow$  All test-centres in  $T$ , sorted by ascending
      distance to  $p$ 
3   foreach test centre  $t \in \Omega$  and time slot  $j \in \mathcal{H}_t$  do
4     if isInsertionFeasible( $p, t, j, S$ ) then
5       foreach evaluation run  $e \in E_{\eta_t}$  do
6         if isInsertionFeasible( $p, t, j, S, e$ ) then
7           insert  $p$  into time slot  $j$  of test ctr.  $t$ 
8           insert  $p$  into eval. run  $e$  of lab.  $\eta_t$ 
9           break and continue with next  $p$ 

```

4.3. Local Search

In order to further exploit promising new solutions we perform a local search on every new solution that is within θ percent of the best found solution. For that purpose, we use the well-known *Or-opt* operator (Or, 1976), which relocates sequences of consecutive vertices (i.e., paths) in vehicle routes. In our implementation, we use a sequence length of up to four consecutive vertices. In case one of the vertices in the sequence represents a laboratory visit, we omit this vertex and include its successor instead. That is, we do not consider the relocation of laboratory visits. The operator is applied intra- and inter-route, and moves are performed in a first-improvement manner as long as improvements can be found.

4.4. Selection of Heuristics and Initial Solution

In each iteration of the LNS algorithm, both the destruction heuristic and the repair heuristic are randomly selected according to a uniform probability distribution. We consider only the first two repair heuristics for this random selection. Assigning a suspected case to a test-centre does not increase the cost of a solution. We therefore call the *time-greedy insertion into test-centre* heuristic at the beginning of each repair phase. If afterwards there are still uncovered cases left, we call the previously randomly selected repair heuristic.

We implemented the strategy proposed in Wolfinger (2021) to generate an initial solution from scratch. The idea is to use the insertion heuristics one by one until a feasible solution is found or all heuristics have been tried; in which case the solution with the least amount of uncovered suspected cases is used as the initial solution. We begin with *random best insertion*, and continue with *regret-k insertion* for $k = 2 \dots 4$. Whenever a heuristic does not yield a feasible solution, we start again from an empty solution. Like in the repair phase, we always first call the *time-greedy insertion into test-centre* heuristic before using any of the other heuristics.

4.5. Acceptance and Stopping Criteria

In order to allow occasional deterioration of solutions throughout the search, which can be useful for escaping local optima, we use threshold acceptance (originally introduced by Dueck & Scheuer, 1990). Given the so far best found solution s^* , a newly found solution s' is accepted if $c(s') \leq (1 + \chi) \cdot c(s^*)$, where $c(\cdot)$ represents the cost of a solution and χ is the threshold (in percent) for s' to still be accepted. The algorithm terminates when a certain amount of time has passed. We denote with v the maximum run time in seconds.

4.6. Penalty

There is a chance that a repair heuristic cannot reinsert all removed suspected cases. When this happens, instead of immediately rejecting this infeasible solution, we increase its cost with a penalty \mathcal{F} and allow it to be accepted as the new incumbent solution. The penalty is computed as

$$\mathcal{F} = (\zeta / |P|)\mu + \zeta^{\frac{\pi}{v}},$$

where μ is the number of uncovered suspected cases in the current solution, π is the current runtime in seconds and ζ represents the cost of the initial solution (cf. Wolfinger, 2021).

5. Computational Study

In this section, we summarize the computational experiments that we performed to assess the performance of the LNS and present managerial insights obtained from analysing real-world based problem instances. The LNS was implemented in C++ and each experiment was executed on a single thread and performed on an Intel Xeon Platinum 8174 3.1 GHz CPU with 2 GB of RAM.

Table 1 summarizes the parameter values for the LNS. To determine those values we first performed a set of experiments testing, *ceteris paribus*, different values for α , β , and χ . We considered the following three sets of values: {0.1, 0.15, 0.2, 0.25}, {0.2, 0.25, 0.3, 0.35} and {0.005, 0.01, 0.025, 0.05} for α , β , and χ , respectively. Then, we performed a second set of experiments to determine the values for γ , θ , and the sequence length of the Or-Opt operator, using the sets {0.5, 0.9, 0.95}, {0.005, 0.01, 0.025} and {2, 3, 4}, respectively. Both sets of experiments have been performed on a small subset of the below described real-world based problem instances; consisting of 14 instances in total with between ca. 200 and 1400 suspected cases per instance.

In the following we first present the problem instances that we used for the computational experiments (Section 5.1). We then evaluate the performance of the LNS in Section 5.2 and present the managerial insights in Section 5.3.

5.1. Test Instances

We generated problem instances for two Austrian provinces, Vienna and Upper Austria. Vienna has a small, almost rectangular shaped area (ca. 415 square kilometres) with a high population density (ca. 4,600 people per square kilometre), whereas Upper Austria has a large, contorted area (ca. 12,000 square kilometres) with a comparatively low population density (ca. 120 people per square kilometre). We replicated the chosen strategies of the Austrian Red Cross – which is the responsible institution for clarifying suspected COVID-19 cases in Austria – for both provinces in our problem instances. That is, a small number of large test-centres and only one vehicle depot with a large vehicle fleet in Vienna, and a large number of small test-centres and several small vehicle fleets in Upper Austria. The Austrian Red Cross provided us with the majority of the necessary data and information to generate our problem instances, such as the number of parallel test stations and the throughput of each test-centre, the service time of mobile test-teams, the duration of an evaluation run in a laboratory, and approximate numbers regarding the capacity and frequency of evaluation runs in laboratories. Specific data relating to laboratories is generally hard to come by, since the majority of utilised laboratories are private businesses and therefore reluctant to provide detailed information. Some of the additional data we used to generate our instances is publicly available, specifically the test-centre locations, the total number of test-teams operating in a province, and the opening hours of test-centres and laboratories.

Table 1

Parameter values for the LNS.

Parameter	Value	Description
α	0.1	min. percentage of suspected cases removed in each iteration of the LNS
β	0.3	max. percentage of suspected cases removed in each iteration of the LNS
γ	0.9	probability of selecting the test-centre which covers the most suspected cases in <i>test-centre opening removal</i>
θ	0.005	threshold value for the local search
ρ_1	9	weight of distance-term for <i>related removal</i>
ρ_2	3	weight of time-term for <i>related removal</i>
ρ_3	5	weight of mandatory-vehicle-visit-term for <i>related removal</i>
X	0.005	threshold value for accepting s' as the new incumbent solution

Only for a few input data we had to make assumptions: the locations of the vehicle depots and the fleet size at each depot, and the aforementioned data related to laboratories. We set the coverage range of each test-centre to 60 minutes and the length of a single test-centre time slot to 90 minutes in all problem instances. Furthermore, we set the maximum allowed duration of a vehicle route equal to the duration of a work shift in the Austrian Red Cross, which is 12 hours, and assume a *time-to-test* and *time-to-result* limit of 24 hours each (which is the current value in Austria). Lastly, in all problem instances, we assumed that 30% of the suspected cases must be clarified by a mobile test-team (i.e., $\Gamma_i = 1$).

We used an epidemiological COVID-19 model, that was designed to forecast the spread of the disease in Austria, to generate the demand data. The model is an agent-based SEIR type simulation model that depicts every inhabitant of Austria as one model agent (see Bicher, Rippinger, Urach, Brunmeir, & Popper, 2020a). The model is stochastic, population-dynamic and based on a validated population model of Austria. The input data for the model comes from the official Austrian COVID-19 disease reporting system (EMS³). The model has been used since early on in the COVID-19 pandemic to support Austrian decision makers, such as ministries and public health agencies (Bicher et al., 2020b).

During the simulation, infected agents progress through a number of different states. One of these states, called *preConfirmed*, represents the situation of *being infected, showing symptoms and awaiting a positive test result*; meaning that an agent in this state requires a COVID-19 IDT. We used the agents who enter this state on a given day as the demand for that day (i.e. as the set of suspected cases P). Specifically, the information we obtained from the simulation model consists of the number of newly *preConfirmed* agents per postal district per day. Clearly, this number represents rather a lower-bound on the actual number of required, respectively demanded, IDTs, as not only infected people require an IDT.

To generate the demand data, we first simulated two scenarios for both provinces: one with high numbers of infections (based on the situation in October/November 2020) and one with low(er) numbers of infections (based on the lock-down phase in December/January 2020/2021). This resulted in 61 days per province, with demands ranging from 211 to 1681 suspected cases. We then grouped the days into three size categories, 0–499, 500–999, and 1000+. For Upper Austria, 36 days fell into the first category and 25 into the second, none into the third. For Vienna, 30, 6 and 25 fell into the first, second and third size category, respectively. We then randomly chose 20 days for each province: ten days from each available size category for Upper Austria, and 4, 6 and 10 days from the first, second and third size category, respectively, for Vienna.

For each of the selected days we generated five problem instances, resulting in 200 problem instances in total (70 in category 0–499, 80 in category 500–999 and 50 in category 1000+). In

all instances, the coordinates of each suspected case are randomly chosen within the corresponding postal district, according to a uniform probability distribution. Likewise, the time of occurrence for each suspected case is chosen randomly as well. All data regarding test-centres and laboratories (such as locations, capacities and opening hours) is the same in all instances belonging to the same province and corresponds to the real-world data as described above. The travel costs and travel times are equal to the real-world street distances and travel times for a car. The distance- and travel time-matrices were generated with the Open Source Routing Machine C++-library (Luxen & Vetter, 2011) using data from the OpenStreetMap (OpenStreetMap contributors, 2017). In order to favour solutions with fewer vehicles – a stated preference of the Red Cross – we set the fixed cost for vehicles rather high compared to the routing cost. Specifically, we set $f^V = (|V| + 1)a^{\max}$, where $a^{\max} = \text{argmax}_{a \in A}\{c_a\}$ is the largest arc in the graph, which can never be exceeded by the routing cost of any tour (f^V is even larger than the length of a hamiltonian cycle in a modified graph in which all arcs are as long as the longest arc in the original graph). This guarantees that, all else unchanged, using less vehicles is always cheaper and therefore preferred. The fixed costs for a test-centre t depend on its size, indicated by the number of test-stations q_t it operates. As it takes the same amount of people to operate a test-station like it does to operate a mobile test-team, we set $f_t^T = f^V \cdot q_t$.

To evaluate the LNS with respect to the solution quality, we compare it with Gurobi 9.11, a commercial MIP solver. For that purpose, we additionally generated a set of small-sized random problem instances. In particular, we generated 5 problem instances for both provinces, with 10, 15, 20, 25 and 30 suspected cases each. In these instances the number of available test-centres, laboratories and vehicles is drastically reduced compared to the real-world instances, such that they can still be handled by the MIP solver.

All instances are publicly available in Wolfinger, Gansterer, Doerner, & Popper (2021).

5.2. Evaluating the Performance of the LNS

In all experiments, we perform 20 runs for each problem instance. The run time limits are set to 1800, 3600 and 7200 seconds for the instances of size 0–499, 500–999 and 1000+, respectively. For the small problem instances we set the run time limit to 30 seconds.

First, we evaluate the quality of the solutions produced by the LNS. For that purpose, we solve the set of small problem instances and calculate for each instance the percentage gap between the best solution found by the LNS and the solution produced by Gurobi ($\text{gap} = 100 \cdot (z_{LNS} - z_{MIP})/z_{MIP}$, where z_{LNS} and z_{MIP} denote the value of the solution produced by the LNS and Gurobi, respectively). We give Gurobi a time limit of 24 hours. Table 2 presents the minimum, average and maximum gap for each size category. As can be seen, for all instances with ten suspected cases Gurobi finds the optimal solution, and so does the LNS. For larger instances, Gurobi could not find any proven optimal solution any

³ <https://www.ages.at/en/wissen-aktuell/publikationen/epidemiologische-parameter-des-covid19-ausbruchs-oesterreich-2020/>. Accessed on February 5, 2021.

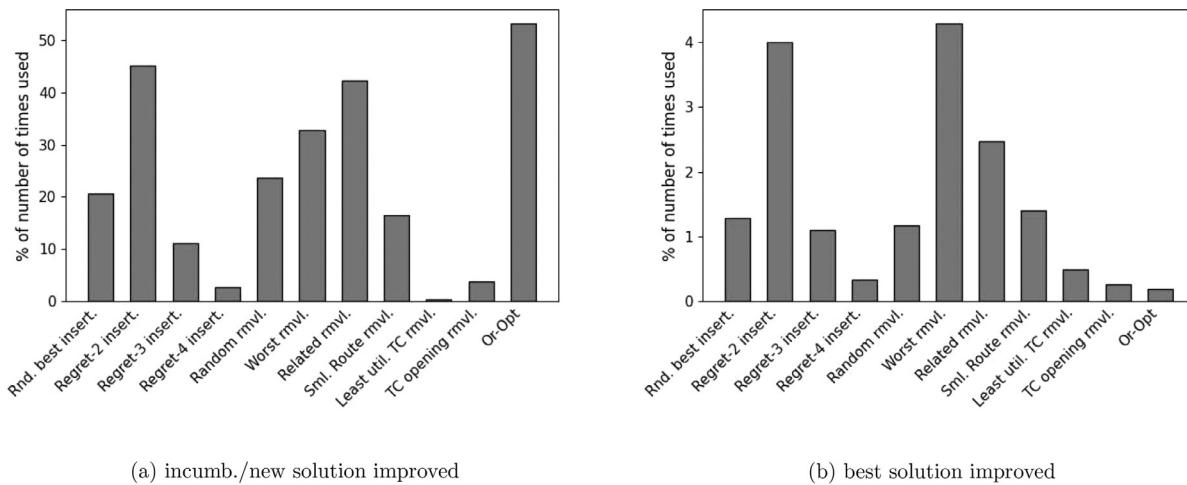


Fig. 2. Efficiency of the destroy/repair heuristics and the local search operator.

Table 2

Comparison LNS with MIP solver (Gurobi). Time limit LNS: 30 seconds per run.

Size	MIP CPU (hours)	LNS Gap (%)		
		Min	Avg	Max
10	4.79	0.00	0.00	0.00
15	24.00	-43.16	-4.80	0.00
20	24.00	-43.92	-8.78	0.00
25	24.00	-63.84	-33.78	0.00
30	24.00	-57.38	-38.20	0.00
Average	20.16	-41.66	-17.11	0.00

more. What is more, for 18 instances Gurobi could not even find a feasible solution (not included in the gap calculations). For all instances for which Gurobi found a solution, the LNS found an equally good or better solution; indicated by the negative minimum and average gaps, respectively the maximum gaps of 0.0%. From these results we conclude that the solution quality of the LNS is very good, and, that a specialized solution method is indeed necessary to solve the CDTP, given the performance of the MIP solver.

We now measure the performance of the destroy and repair heuristics, as well as of the local search operator. For that purpose, we calculate the number of times a heuristic or the local search operator improved the incumbent solution (newly found solution), respectively the best solution. Both indicators are expressed as the percentage over the number of times the heuristic/local search operator was used. Fig. 2 shows the values of both indicators; which are averages over all instances. The best insertion heuristic is *regret-2 insertion*. It clearly outperforms all other insertion heuristics in both measures. *Random best insertion* performs significantly better than *regret-3 insertion* with respect to improving the incumbent solution, but only marginally better for improving the best solution. *Regret-4 insertion* ranks last, with distinctly smaller values for both indicators. The *time-greedy insertion into test-centre* heuristic is not included in the comparison, since it is used in each iteration of the LNS algorithm. The best destruction heuristics are *related removal*, with respect to improving the incumbent solution, and *worst removal*, with respect to improving the best solution. It is worth mentioning, that *worst removal* significantly outperforms all other destroy heuristics, improving the best solution at least almost twice as often as any other destroy heuristic. Following (in order) are *random removal*, *smallest route removal*, *test-centre opening removal*, and lastly *least utilised test-centre removal*. The destroy heuristics which specifically consider test-centres rank last and perform considerably worse than the re-

Table 3

Percentage change in solution cost when destroy/repair heuristic is not used (all else unchanged).

Heuristic	Min	Avg	Max
Random best insertion	-23.49	-0.59	33.23
Regret-2 insertion	-27.72	1.08	33.22
Regret-3 insertion	-16.67	1.94	39.91
Regret-4 insertion	-27.73	2.37	45.38
Random removal	-29.37	2.37	36.32
Worst removal	-26.56	3.54	33.33
Related removal	-26.58	0.58	36.77
Smallest route removal	-29.38	0.83	29.35
Least utilised test-centre removal	-19.96	61.15	371.54
Test-centre opening removal	-33.25	3.78	50.05

maining heuristics. This is to be expected, given that they alter the structure of a solution to a much larger extent than the other destroy heuristics. Indeed, this is in fact a desired behaviour in order to cover more and different areas of the solution space. The local search operator, *Or-Opt*, performs well; improving the newly found solution roughly 50% of the time and the best solution ca. 0.2% of the time.

In order to further analyse the contribution of the heuristics, we removed them one at a time and re-solved all problem instances. Table 3 shows for each heuristic the percentage change in the solution cost when it is not used compared to the full LNS. It can be seen that all heuristics positively contribute to the solution quality. That is, even though cost reductions of up to ca. 16–33% can be achieved by not using a heuristic (depending on the heuristic, see column *Min*), on average, for each heuristic, not using it leads to an increase of the solution cost. The only exception being *random best insertion*. However, due to the significantly worse worst-case (cost increase of ca. 33%) compared to the best-case (cost reduction of ca. 23%), it is reasonable to include it in the LNS. It is also worth mentioning that the *least utilised test-centre removal* heuristic – which performs worst, respectively second worst, with respect to improving the incumbent and best solution – significantly contributes to the solution quality. Not using it increases the cost of a solution by roughly 60%, on average.

5.3. Managerial Insights

In this section, we analyse the solutions of the real-world problem instances. In all analyses, we take the best run for each instance as a basis.

Table 4

Principal solution characteristics. The numbers in brackets next to 'Vehic.' and 'Test-Ctr.' represent the number of available vehicles and test-centres, respectively.

Size	Upper Austria					Vienna				
	# required		T2TD	T2RD	TCT	# required		T2TD	T2RD	TCT
	Vehic. (20)	Test-Ctr. (13)				Vehic. (29)	Test-Ctr. (3)			
0–499	9.54	3.18	1.02	6.99	8.01	4.95	0.90	1.14	6.81	7.95
500–999	16.34	5.24	0.91	6.96	7.87	8.73	1.00	0.90	6.95	7.85
1000+	—	—	—	—	—	17.58	1.02	1.15	8.43	9.57
Average	12.94	4.21	0.97	6.97	7.94	10.42	0.97	1.06	7.40	8.46

Table 5

Further solution characteristics.

Size	Upper Austria					Vienna				
	Distribution (%)		Vehicle routes		Test-Ctr.	Distribution (%)		Vehicle routes		Test-Ctr.
	Mobile	Test-Ctr.	Len. (km h)	# Stops	Util. (%)	Mobile	Test-Ctr.	Len. (km h)	# Stops	Util. (%)
0–499	38.40	61.60	270.96 11.92	14.44	14.65	37.18	62.82	98.17 11.95	22.49	16.80
500–999	34.77	65.23	273.70 11.94	17.21	22.45	30.67	69.33	105.01 11.96	27.29	45.60
1000+	—	—	— —	—	—	42.84	57.16	108.65 11.97	35.31	66.57
Average	36.59	63.41	272.33 11.93	15.82	18.55	36.90	63.10	103.94 11.96	28.36	42.99

First, we report some interesting indicators, summarising the principal characteristics of the solutions. Table 4 shows for each province and size category the number of required vehicles and test-centres. The depicted values are averages over all instances of the same size category. The results show that, on average, a significantly larger number of both vehicles and test-centres is used in Upper Austria compared to Vienna. Given the fairly larger area of Upper Austria combined with the test-centre coverage range and the *time-to-test* and *time-to-result* limits, this is of course expected. Furthermore, it can be seen that in both provinces the number of used vehicles and test-centres is well under the number of available vehicles and test-centres. This provides an indication that the strategies implemented by the Austrian Red Cross were chosen well, and that they can potentially even handle significantly higher numbers of suspected cases. The table additionally shows the average *time-to-test* duration (T2TD), *time-to-result* duration (T2RD) and total clarification time (TCT), in hours. We can see that all three values are, again, much smaller than the maximum allowed values (of 24 and 48 hours, respectively), corroborating the presumed resilience of the chosen strategies. Contrary to before, there is no significant difference between both provinces, with average T2TD, T2RD and TCT values of roughly 1, 7 and 8 hours, respectively.

Table 5 presents additional solution characteristics, providing further insights into the solution structure. It gives information on the average length and number of stops of a vehicle route and shows the average utilisation of open test-centres. Additionally, the table depicts the percentage of suspected cases that is clarified with a mobile test-team, respectively in a test-centre. All values are again averages over all instances of the same size category. The results show that, on average, the mobile test-teams are fully utilised in both provinces; which is indicated by the average vehicle route duration of close to 12 hours. (Remember, the maximum allowed duration of a vehicle route is 12 hours). As expected, the vehicle routes in Upper Austria are much longer in terms of distance and contain noticeably fewer stops compared to the vehicle routes in Vienna; roughly 2.7 times longer and only 60% of the number of stops. On average, a mobile test-team in Upper Austria travels roughly 270 kilometres and visits around 14 suspected cases (the average number of laboratories visited on a route is 1), while a mobile test-team in Vienna travels roughly 104 kilometres and visits 27 suspected cases on its tour. Contrary to the mobile test-teams, the degree of utilisation of test-centres is rather low. This is for the most part, however, simply due to the overall large

capacity provided by them. Regarding the distribution of suspected cases between mobile test-teams and test-centres, we observe that the vast majority of suspected cases that can be clarified in a test-centre, is clarified in a test-centre. In both provinces, on average, only roughly 6% of the "optional" suspected cases are clarified with a mobile test-team.

In the next set of experiments we analyse the influence of the *time-to-result* limit on the solution cost and structure. For that purpose, we solve each instance again, but without the *time-to-result* limit. Because of the considerably larger solution space for this version, we doubled the run time limit for the LNS. Table 6 presents the percentage change in the solution cost, the percentage of suspected cases which would violate the *time-to-result* limit (including the average and maximum violation in hours), and the absolute change in the principal solution characteristics, averaged over each size category. The results show that relaxing the *time-to-result* limit can have a considerable impact on the cost. For large instances, cost reductions of up to roughly 10%, on average, could be achieved when not adhering to the *time-to-result* limit. For smaller instances, the cost savings are much smaller, amounting to only 1–2% on average. This is to be expected, since in larger instances the *time-to-result* limit is much more constraining. As can be seen in the table, the cost reductions are mainly caused by the potential to reduce the number of used vehicles. On days with more than 1000 suspected cases, not considering the *time-to-result* limit can, on average, save up to almost 3 vehicles. The number of required test-centres, on the other hand, stays approximately the same, no matter the magnitude of suspected cases. While for days with a low or medium amount of potential cases, relaxing the *time-to-result* limit has virtually no impact on the average T2RD, we can observe an increase of, on average, more than 5 hours (ca. 60%) in the average T2RD on days with a high number of potential cases. This then translates to an increase of the average TCT of roughly the same amount. The average percentage of suspected cases that would violate the *time-to-result* limit is negligible in small and medium sized instances, but amounts to more than 24% of the suspected cases, on average, in large instances; with average and maximum violations of approximately 4 and 12 hours, respectively.

Note that since all suspected cases in our problem instances occur on the same day, the *time-to-test* limit is not restrictive and has therefore no impact on the cost and structure of a solution. An analysis regarding the impact of relaxing the *time-to-test* limit is thus obsolete.

Table 6

Percentage change in cost, *time-to-result* limit violations, and absolute change in principal solution characteristics when no *time-to-result* limit is in place.

Size	Upper Austria										Vienna					
	Cost	# required		T2R violation			T2RD	TCT	Cost	# required		T2R violation			T2RD	TCT
		Vehic.	Test-Ctr.	% Cases	Avg (h)	Max (h)				Vehic.	Test-Ctr.	% Cases	Avg (h)	Max (h)		
0–499	−0.73	−0.06	−0.08	0.00	0.00	0.00	0.02	0.02	−1.10	−0.10	0.00	0.02	0.08	0.08	0.04	−0.01
500–999	−1.07	0.06	−0.26	0.00	0.00	0.00	0.00	0.00	−1.88	−0.27	0.00	3.65	1.79	4.87	2.09	2.08
1000+	−	−	−	−	−	−	−	−	−9.65	−2.98	0.18	24.23	3.90	12.70	5.25	5.10
Average	−0.90	0.00	−0.17	0.00	0.00	0.00	0.01	0.01	−4.21	−1.12	0.06	9.30	1.93	5.88	2.46	2.39

Table 7

Percentage change in both cost and number of suspected cases which are clarified in a test-centre, and absolute change in principal solution characteristics when the length of the test-centre time slots is halved (from 90 to 45 minutes).

Size	Upper Austria								Vienna					
	Cost	# required		Test-Ctr. clarification	T2TD	T2RD	TCT	Cost	# required		Test-Ctr. clarification	T2TD	T2RD	TCT
		Vehic.	Test-Ctr.						Vehic.	Test-Ctr.				
0–499	−3.19	0.08	−0.24	−1.16	0.02	−0.24	−0.21	−0.55	−0.05	0.00	0.00	−0.01	−0.34	−0.34
500–999	0.39	0.30	−0.18	−0.76	0.02	−0.29	−0.28	−0.48	−0.07	0.00	0.49	−0.01	−0.37	−0.38
1000+	−	−	−	−	−	−	−	−1.09	−0.44	0.00	−0.16	−0.01	−0.04	−0.05
Average	−1.40	0.19	−0.21	−0.96	0.02	−0.26	−0.25	−0.71	−0.19	0.00	0.11	−0.01	−0.25	−0.26

Table 8

Percentage change in both cost and number of suspected cases which are clarified in a test-centre, and absolute change in principal solution characteristics when the length of the test-centre time slots is doubled (from 90 to 45 minutes).

Size	Upper Austria								Vienna					
	Cost	# required		Test-Ctr. clarification	T2TD	T2RD	TCT	Cost	# required		Test-Ctr. clarification	T2TD	T2RD	TCT
		Vehic.	Test-Ctr.						Vehic.	Test-Ctr.				
0–499	−2.25	0.08	−0.04	−0.29	−0.01	0.59	0.58	−1.07	−0.11	0.00	0.00	−0.04	0.58	0.54
500–999	1.05	0.26	−0.06	−0.27	0.01	0.61	0.62	1.32	0.17	0.00	−1.65	0.04	1.15	1.18
1000+	−	−	−	−	−	−	−	5.40	−0.06	0.10	−1.66	−0.14	1.16	1.01
Average	−0.60	0.17	−0.05	−0.28	0.00	0.60	0.60	1.88	−0.00	0.03	−1.10	−0.05	0.96	0.91

In practice, the length of test-centre time slots may be subject to a variety of factors, such as the availability of resources (e.g., staff and vehicles) or customer (in)convenience. We therefore now analyse the impact of the length of the time slots in test-centres, both on the cost and on the solution structure. For that purpose, we re-solve all instances, once with the length of all time slots halved (from 90 to 45 minutes) and once with the length doubled (from 1.5 to 3 hours). Tables 7 and 8 present the results for both versions, respectively, depicting the percentage change in both the cost and the number of suspected cases which are clarified in a test-centre, and the absolute change in the principal solution characteristics; averaged over each size category. The results show that halving the duration of the time slots has no significant impact on the cost, nor on the solution structure in general. The cost decrease by ca. 0.7–1.4% on average, depending on the province, and the change in the number of suspected cases which are clarified in a test-centre is less than 1% on average. Furthermore, all clarification times – T2TD, T2RD and TCT – change by less than 0.5 hours. Doubling the duration of the time slots has a comparatively bigger impact, but it is still not large. For small and medium sized problem instances the cost change on average by at most 2.5%, while for large problem instances the cost increase by roughly 5%, on average. The change in the number of suspected cases which are clarified in a test-centre is again negligible – less than 1% on average – and the clarification times increase by 0.5–1 hour, on average. Summarized, the results indicate that the duration of the test-centre time slots has no significant impact on the cost, nor on the number of suspected cases which are clarified in a test-centre, and neither on the clarification times

(T2TD, T2RD and TCT); which gives some wiggle room to practitioners.

Up until now, in all our experiments the implicit assumption was that we know in advance when and where the suspected cases will occur. Obviously, this assumption does not hold in practice. However, what can be done in practice is to collect all the suspected cases that occur throughout the day and then clarify them on the next day. Using this strategy, our solution method can be directly applied in practice, as is. We now evaluate the impact of this strategy on the solution cost and structure, as well as its cost in terms of uncovered suspected cases due to violation of the *time-to-test* and *time-to-result* limits. For that purpose, we subtract 24 hours from the time of occurrence of each suspected case in the problem instances – to simulate the situation of clarifying yesterday's suspected cases – and solve them again. To give an example, assume that in the original instance the time of occurrence of suspected case p is minute 180, i.e., $a_p = 180$. Then, the time of occurrence of p in the modified instance is $a_p = 180 - 1440 = -1260$. Table 9 shows that when we use the proposed strategy in Vienna, on average, only 3% of the suspected cases cannot be covered within the *time-to-test* and/or *time-to-result* limit. For Upper Austria this value is twice as high, which is, however, still rather low. Overall, the vast majority of suspected cases (between 93% and 97%, on average) can still be clarified within the required time limits. As can be expected, we observe significant cost increases compared to the situation when we have perfect information. In Vienna, the costs are between 36% and 87% higher, on average, while the cost increase in Upper Austria is only around 16–26%, on average. Similar to relaxing the *time-to-result* limit, the change

Table 9

Uncovered suspected cases, percentage change in cost, and absolute change in number of required vehicles and test centres when clarifying yesterday's suspected cases.

Size	Upper Austria				Vienna			
	% uncov.	Cost	# required		% uncov.	Cost	# required	
			Vehic.	Test-Ctr.			Vehic.	Test-Ctr.
0–499	4.89	16.01	7.80	-1.50	2.81	36.48	6.55	-0.85
500–999	8.43	26.40	3.44	2.42	2.45	86.86	15.13	-0.90
1000+	-	-	-	-	3.74	83.31	9.88	1.40
Average	6.66	21.21	5.62	0.46	3.00	68.88	10.52	-0.12

Table 10Percentage change in cost, time limit violations, and absolute change in principal solution characteristics when clarifying yesterday's suspected cases and either the *time-to-test* limit, or the *time-to-result* limit, or both limits are *not* in place; compared to when both limits are in place.

		Size	Cost	# required			T2T violation			T2R violation			T2TD	T2RD	TCT
				Vehic.	Test-Ctr.	% Cases	Avg (h)	Max (h)	% Cases	Avg (h)	Max (h)				
Upper Austria	no <i>time-to-test</i> limit	0–499	-55.26	-10.86	0.84	35.06	3.05	10.11	0.00	0.00	0.00	1.25	0.19	1.44	
		500–999	-49.02	-9.44	-4.64	33.78	3.18	10.60	0.00	0.00	0.00	2.17	0.96	3.13	
		1000+	-	-	-	-	-	-	-	-	-	-	-	-	
		Average	-52.14	-10.15	-1.90	34.42	3.12	10.36	0.00	0.00	0.00	1.71	0.58	2.29	
	no <i>time-to-result</i> limit	0–499	-0.87	-0.24	0.08	0.00	0.00	0.00	0.00	0.00	0.00	-0.02	-0.04	-0.05	
		500–999	-0.82	0.00	-0.18	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.04	0.05	
		1000+	-	-	-	-	-	-	-	-	-	-	-	-	
		Average	-0.85	-0.12	-0.05	0.00	0.00	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	
	neither limit	0–499	-55.38	-10.54	0.52	36.01	3.09	9.95	0.00	0.00	0.00	1.41	0.29	1.70	
		500–999	-49.97	-9.28	-4.64	35.78	3.29	10.98	0.00	0.00	0.00	2.40	0.97	3.37	
		1000+	-	-	-	-	-	-	-	-	-	-	-	-	
		Average	-52.68	-9.91	-2.06	35.90	3.19	10.46	0.00	0.00	0.00	1.90	0.63	2.54	
Vienna	no <i>time-to-test</i> limit	0–499	-52.69	-7.60	0.35	41.03	3.39	10.52	0.00	0.00	0.00	1.70	0.95	2.66	
		500–999	-64.16	-19.27	0.97	34.60	2.81	10.42	0.00	0.00	0.00	0.54	-0.90	-0.36	
		1000+	-70.39	-21.64	-0.84	50.75	3.95	11.81	0.00	0.00	0.00	3.42	2.59	6.01	
		Average	-62.41	-16.17	0.16	42.13	3.38	10.92	0.00	0.00	0.00	1.89	0.88	2.77	
	no <i>time-to-result</i> limit	0–499	-1.29	-0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	-0.09	-0.02	
		500–999	-0.89	-0.23	0.00	0.00	0.00	0.00	3.00	0.90	1.98	0.02	1.03	1.06	
		1000+	-0.38	0.00	-0.04	0.00	0.00	0.00	5.23	1.78	4.08	0.03	1.40	1.43	
		Average	-0.86	-0.13	-0.01	0.00	0.00	0.00	2.74	0.89	2.02	0.04	0.78	0.82	
	neither limit	0–499	-52.69	-7.60	0.35	40.74	3.39	10.44	2.77	0.89	1.65	1.69	1.87	3.56	
		500–999	-64.42	-19.33	0.97	34.70	2.81	10.44	3.08	1.71	3.34	0.55	0.23	0.79	
		1000+	-70.71	-21.78	-0.84	51.01	3.96	11.88	22.61	4.40	14.43	3.45	6.64	10.09	
		Average	-62.60	-16.24	0.16	42.15	3.39	10.92	9.49	2.33	6.47	1.90	2.91	4.81	

in cost is mainly due to a change in the number of used vehicles. On average, clarifying yesterday's suspected cases, requires 5 to 10 vehicles more, as well as, half a test-centre more.

In the last set of experiments, we now analyse the impact of both time limits when clarifying yesterday's suspected cases. For that purpose, we solve each instance again, once without the *time-to-test* limit, once without the *time-to-result* limit, and once without both limits; assuming in all three versions that we clarify yesterday's suspected cases. Table 10 shows again the percentage change in the solution cost, the percentage of suspected cases which would violate a time limit, and the absolute change in the principal solution characteristics, averaged over each size category and separately for each version. We observe a significant impact of the *time-to-test* limit on the cost. Through relaxing these constraints the cost could be reduced by 50–70% on average, depending on the magnitude of suspected cases and the province. The cost reductions are again driven by a reduction in the number of used vehicles. As can be expected, the percentage of suspected cases that are not clarified within the *time-to-test* limit is rather high, 30–50%; with average and maximum violations of roughly 3 and 10 hours. The average TCT increases by approximately 2–3 hours, on days with more than 1000 cases by ca. 6 hours, compared to

when the *time-to-test* limit is respected. The impact of relaxing the *time-to-result* limit is considerably smaller. The cost reductions are negligible and the impact on the clarification time is also small, 0.8 hours on average. Consequently, the version which relaxes both time limits differs only marginally from the version which relaxes only the *time-to-test* limit. We can conclude that not adhering to duration limits can be very lucrative in terms of cost, but is bought for a (potentially high) price in terms of a significant increase in the total clarification time.

6. Conclusion

In this article we introduced a novel logistics problem, called the contagious disease testing problem (CDTP), which arises in the context of the public health response to epidemics and pandemics. The task in this problem is to simultaneously decide the location of test-centres and design vehicle routes for mobile test-teams, such that a number of potentially infected people can be clarified, i.e., tested for the disease, either by visiting an open test-centre or by being visited by a mobile test-team. The objective is to minimise the total cost of opening test-centres and routing mobile test-teams.

We presented an arc-based mixed-integer formulation for the CDTP and proposed a large neighbourhood search (LNS) algorithm for solving the problem. The performance of the proposed LNS was validated through extensive computational experiments, using a newly created benchmark collection which contains both small and real-world (sized) problem instances. The results showed that the algorithm is efficient to solve the CDTP.

Additionally, we analysed and drew insights from the solutions of the real-world instances. We found that restricting the duration between the time of occurrence of a potential case and the time of taking an IDT, as well as restricting the time between taking an IDT and attaining its result, can cause a significant cost increase. However, large cost savings are typically linked with equally significant increases of the time it takes to clarify a suspected case. Furthermore, we showed that our method can be applied in practice as is, with only a small percentage of suspected cases not being clarified within the desired amount of time.

As we already mentioned in the introduction, the real-world problem is of course stochastic and dynamic. Therefore, an interesting and important direction for future research would be to address the CDTP in a stochastic-dynamic context. Developing methods that can schedule dynamically occurring suspected cases is of great value and importance, especially for practice. Indeed, we plan to adapt the proposed LNS to exactly this context. Nevertheless, we hope that the presented research can already contribute to the COVID-19 crisis management.

Acknowledgements

This research was funded in whole by the [Austrian Science Fund \(FWF\)](#) (Grant number P-34151-N). We are very thankful to the dwh GmbH, and in particular Martin Bicher, for their help with generating the problem instances and their valuable input in general. We also wish to extend our gratitude to the Austrian Red Cross for helping us to properly understand and specify the real-world problem, and for their provided data. Lastly, we would like to thank the two anonymous referees for their helpful comments.

References

- Ahmadi-Javid, A., Seyed, P., & Syam, S. S. (2017). A survey of healthcare facility location. *Computers and Operations Research*, 79, 223–263.
- Arlan, O. (2020). The location-or-routing problem. *Technical Report*. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT).
- Bicher, M., Rippinger, C., Urach, C., Brunmeir, D., & Popper, N. (2020a). Agent-based simulation for evaluation of contact-tracing policies against the spread of SARS-CoV-2. medRxiv. 10.1101/2020.05.12.20098970
- Bicher, M., Zuba, M., Rainer, L., Bachner, F., Rippinger, C., & Ostermann, H. et al. (2020b). Supporting Austria through the COVID-19 epidemics with a forecast-based early warning system. medRxiv. 10.1101/2020.10.18.20214767
- Drexl, M., & Schneider, M. (2015). A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, 241(2), 283–308.
- Dueck, G., & Scheuer, T. (1990). Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, 90(1), 161–175. [https://doi.org/10.1016/0021-9991\(90\)90201-B](https://doi.org/10.1016/0021-9991(90)90201-B)
- Farahani, R. Z., Asgari, N., Heidari, N., Hosseiniinia, M., & Goh, M. (2012). Covering problems in facility location: A review. *Computers and Industrial Engineering*, 62(1), 368–407.
- Garey, M., & Johnson, D. (1979). Computers and intractability: A guide to the theory of NP-completeness. *Mathematical sciences series*. W. H. Freeman.
- Güneş, E. D., Melo, T., & Nickel, S. (2019). Location problems in healthcare. In G. Laporte, S. Nickel, & F. Saldanha da Gama (Eds.), *Location science* (pp. 657–686). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-32177-2_23.
- Hemmelmayr, V. C., Cordeau, J.-F., & Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers and Operations Research*, 39(12), 3215–3228.
- Lee, E. K., Chen, C.-H., Pietz, F., & Benecke, B. (2009). Modeling and optimizing the public-health infrastructure for emergency response. *Interfaces*, 39(5), 476–490.
- Luxen, D., & Vetter, C. (2011). Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM sigspatial international conference on advances in geographic information systems* (pp. 513–516). New York, NY, USA: ACM. <https://doi.org/10.1145/2093973.2094062>.
- Martínez-Reyes, A. F., Quintero-Araújo, C. L., & Solano-Charris, E. L. (2020). A decision support tool for the location routing problem during the COVID-19 outbreak in Colombia. *Technical Report*.
- Matukas, L. M., Dhalla, I. A., & Laupacis, A. (2020). Aggressively find, test, trace and isolate to beat COVID-19. *Canadian Medical Association Journal*, 192(40), E1164–E1165. <https://doi.org/10.1503/cmaj.202120>.
- OpenStreetMap contributors (2017). *Planet dump retrieved from planet.osm.org*. <https://www.openstreetmap.org>
- Or, I. (1976). *Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking*. Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL. Ph.D. thesis.
- Pisinger, D., & Ropke, S. (2019). Large neighborhood search. In M. Gendreau, & J.-Y. Potvin (Eds.), *Handbook of metaheuristics* (pp. 99–127). Springer.
- Prodhon, C., & Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1), 1–17.
- Rajan, S., Cylus, J. D., & McKee, M. (2020). What do countries need to do to implement effective 'find, test, trace, isolate and support' systems? *Journal of the Royal Society of Medicine*, 113(7), 245–250. <https://doi.org/10.1177/0141076820939395>. PMID: 32663428
- Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4), 455–472.
- SAGE (2020). Final integrated find, test, trace, isolate, support (FITTIS) response to the pandemic.
- Schiffer, M., Schneider, M., Walther, G., & Laporte, G. (2019). Vehicle routing and location routing with intermediate stops: A review. *Transportation Science*, 53(2), 319–343.
- Schiffer, M., & Walther, G. (2018). An adaptive large neighborhood search for the location-routing problem with intra-route facilities. *Transportation Science*, 52(2), 331–352.
- Schneider, M., & Drexl, M. (2017). A survey of the standard location-routing problem. *Annals of Operations Research*, 259(1–2), 389–414.
- Schneider, M., Stenger, A., & Hof, J. (2015). An adaptive VNS algorithm for vehicle routing problems with intermediate stops. *Or Spectrum*, 37(2), 353–387.
- Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK.
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher, & J.-F. Puget (Eds.), *Principles and practice of constraint programming—CP98* (pp. 417–431). Springer Berlin Heidelberg.
- WHO (2018). *Managing epidemics: Key facts about major deadly diseases*. Geneva: World Health Organization. Licence: CC BY-NC-SA 3.0 IGO
- Wolfinger, D. (2021). A large neighborhood search for the pickup and delivery problem with time windows, split loads and transshipments. *Computers and Operations Research*, 126, 105110. <https://doi.org/10.1016/j.cor.2020.105110>.
- Wolfinger, D., Gansterer, M., Doerner, K. F., & Popper, N. (2021). "CDTP", mendeley data, v2. 10.1763/4363n6n9nm.2