Omega 135 (2025) 103305

Contents lists available at ScienceDirect

Omega

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

Managing equitable contagious disease testing: A mathematical model for resource optimization

Peiman Ghasemi [a,*], Jan Fabian Ehmke [b], Martin Bicher [c]

[a] University of Vienna, Department of Business Decisions and Analytics, Kolingasse 14-16, 1090 Vienna, Austria
[b] University of Vienna, Department of Business Decisions and Analytics, Vienna, Austria
[c] Institute of Information Systems Engineering, TU Wien, Favoritenstraße 11, 1040 Vienna, Austria

A R T I C L E I N F O

Editor: Dr J Zhu

Keywords:
Contagious disease testing
Team orienteering problem
Multi objective variable neighborhood search
Non-dominated sorting genetic algorithm
Equitable testing

A B S T R A C T

All nations in the world were under tremendous economic and logistical strain as a result of the advent of COVID-19. Early in the epidemic, getting COVID-19 diagnostic tests was a significant difficulty. Furthermore, logistical challenges arose from the restricted transportation infrastructure and disruptions in international supply chains in the distribution of these testing kits. In the face of such obstacles, it is critical to give patients' needs top priority in order to provide fair access to testing. In order to manage contagious disease testing, this work proposes a bi-objective and multi-period mathematical model with an emphasis on mobile tester route plans and testing resource allocation. In order to optimize patient scores and reduce the likelihood of patients going untreated, the suggested team orienteering model takes into account issues like resource limitations, geographic clustering, and testing capacity limitations. To this aim, we present a comparison between quarantine and non-quarantine scenarios, introduce an equitable categorization based on disease backgrounds into "standard" and "risky" groups, and cluster geographical locations according to average age and contact rate. We use a Multi-Objective Variable Neighborhood Search (MOVNS) and a Non-Dominated Sorting Genetic Algorithm II (NSGA-II) to solve our problem. Due to the superiority of MOVNS, it is applied to a case study in Vienna, Austria. The results demonstrate that, over the course of several weeks, the average number of unserved risky patients in the prioritizing scenario is consistently lower than the usual number of patients. In the absence of prioritization, the average number of high-risk patients who remain untreated rises sharply and exceeds that of regular patients, though. Furthermore, it is clear that waiting times are greatly impacted by demand volume when comparing scenarios with and without quarantine.

1. Introduction

In late 2019, a new virus, COVID-19, emerged in Wuhan, China, rapidly spreading worldwide within weeks. As of January 18, 2024, statistics indicate that 701,980,740 people were infected, and 6,970,519 lives were lost. For example, the number of infected people in the United States, Austria, and Germany was 110,610,761, 6,081,287, and 38,796,602, respectively. The total number of deaths in these countries was 1,192,813, 22,542, and 182,375, respectively[1].

An important strategy in keeping COVID-19 from spreading was the appropriate assignment of testing resources to individuals that showed symptoms of COVID-19. For instance, Austria's government managed the pandemic with lockdowns, vaccination campaigns, and widespread testing. In April 2021, the "Alles Gurgelt" campaign launched in Vienna, offering PCR testing twice a week for residents. Individuals with COVID-19 symptoms contacted a hotline, and contact tracing was initiated based on disease records. Subsequently, the laboratory dispatched mobile testers, prioritized according to urgency, to collect samples from patients in patients' homes [45]. Results were promptly communicated, allowing for swift treatment and contact tracing. Depending on the test outcome, individuals could initiate treatment and inform other potentially infected individuals with whom they have recently been in contact, such as family members, neighbors, or coworkers [32].

From a logistics point of view, mobile testers depart from depots,

Area: Innovative Applications. This manuscript was processed by Associate Editor Jay Rosenberger.

* Corresponding author.
E-mail addresses: peiman.ghasemi@univie.ac.at (P. Ghasemi), jan.ehmke@univie.ac.at (J.F. Ehmke), martin.bicher@tuwien.ac.at (M. Bicher).

[1] https://www.worldometers.info/coronavirus/

https://doi.org/10.1016/j.omega.2025.103305
Received 3 May 2024; Accepted 22 February 2025
Available online 25 February 2025

visit patients, and collect samples before returning to the laboratories. Following Wolfinger et al. [46], we call this the Contagious Disease Teasing Problem. We model the problem as Team Orienteering Problem (TOP). By optimizing routes and resource allocation, the TOP facilitates efficient delivery of testing kits while minimizing exposure risks, which is essential in navigating the challenges of a pandemic. The TOP was first described by Chao et al. [10], considering multiple vehicles. Its main goal is to maximize the total collected reward by selecting a subset of customers for each vehicle and ordering their visits. In the TOP, the fleet size is limited, and vehicles have a maximum tour length or service time limit. This means that only a subset of customers in the network can be served, which corresponds to the situation of limited resources as observed in the COVID-19 pandemic. The main objective is to maximize the total collected reward from customer visits. Because the TOP is NP-hard, most of the solution approaches in the literature make use of metaheuristics such as tabu search algorithms and the variable neighborhood search algorithm introduced by Archetti et al. [2].

In the face of the overwhelming challenges posed by the COVID-19 pandemic, prioritizing patients based on urgency became paramount in the efficient allocation of limited testing resources. This is especially for equitable testing, ensuring that those most in need receive timely care and attention. Identifying and swiftly attending to individuals exhibiting symptoms not only ensures timely diagnosis and treatment but also plays a crucial role in containing the spread of a virus. By focusing on those most in need of testing and medical attention, healthcare systems can effectively manage their resources and mitigate the strain on hospitals and laboratories, ultimately safeguarding public health. So, prioritizing patients in the logistical management of a pandemic such as emerging with COVID-19 is a critical strategy in optimizing equitable delivery of essential healthcare services.

Therefore, in this paper, a bi-objective and multi-period mathematical model for the equitable contagious disease testing problem is investigated. The proposed model optimizes the equitable delivery of testing kits from depots to patients' locations and, finally, their delivery to laboratories. The main goal of this research is to maximize the score of the collected patients and minimize the number of unserved patients. The main contributions of this paper include: Firstly, we introduce a multi-period and bi-objective TOP for managing the equitable contagious disease testing problem and address it with a mixed-integer linear programming formulation (MILP). Secondly, we cluster geographical areas based on the age of patients and their contact rate, and prioritize patients based on their level of risk considering resource limitations. Thirdly, we investigate the Multi-Objective Variable Neighborhood Search (MOVNS) and the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) metaheuristics to tackle this problem and validate them through extensive computational experiments. Given the NP-hard nature of the multi-period team orienteering problem, employing a metaheuristic approach enables satisfactory results for real-world scenarios within reasonable time frames. Fourthly, we examine instances of the problem using real-world data from Austria, providing managerial insights into COVID-19 testing procurement.

The remainder of this paper is as follows. In Section 2, the problem description is discussed. Section 3 covers the literature review and research gaps, while Section 4 addresses assumptions and the mathematical model. Then, in Section 5, the solution methodology is described, followed by computational results, managerial insights (Section 6), and conclusions (Section 7).

2. Problem description

We aim to discover an effective routing for a group of mobile testers who gather patient specimens from various locations. The network features mobile testers originating from depots, efficiently collecting specimens from patient locations, and ensuring delivery to laboratories. Starting from a designated depot in the testing area, each route taken by a mobile tester concludes at the laboratories, after a day's tour. Due to the limited testing capacities of the laboratories, determined by the availability of testing kits and the working time of mobile testers, the number of specimens collected at the laboratories should not surpass this limit.

The patient's service depends on the laboratories and mobile testers' capacity as well as the working hour limitations of resources. The overarching objective of this paper is to optimize patient allocation and routing plans for mobile testers. We seek to maximize the overall patient scores by strategically prioritizing risky and standard patients in the process. This prioritization is crucial as it ensures equitable resource allocation and improves the overall quality of testing provided. Conversely, another objective is to address the critical need to minimize the number of unserved patients within the system. By considering strategically prioritizing patient groups based on factors such as age and recent contacts, the model aims to mitigate the potential negative consequences of unserved patients while equitably managing available resources.

The score of a patient is determined as follows. The patients are divided into risky and standard patients based on the concept of *hazard ratio*. The patients with a hazard ratio equal to 1 are considered *standard* patients, while those with a hazard ratio greater than 1 are considered *risky* patients. To calculate the hazard ratio for each patient, we utilize the methodology proposed by Jahn et al. [23], who calculate hazard ratios for various patient groups based on specific health conditions. For instance, patients with kidney disease have a hazard ratio of 2, while those with diabetes have a hazard ratio of 1.95. For heart disease, the hazard ratio is 1.17, for respiratory disease, it is 1.63, for liver disease, it is 1.75, and for cancer, it is 1.72. If a patient has several diseases simultaneously, the corresponding coefficients are multiplied. For example, a patient who has both cancer and hypertension has a hazard ratio value equal to $1.59 * 1.75 * 1.72 = 4.7859$. The literature indicates that regions with greater contact rates and higher average ages face increased exposure to infectious diseases. Therefore, residents of these areas should receive higher priority for services. Consequently, priority groups are defined to include those least susceptible to extremely susceptible individuals.

The prioritization of patients is based on their disease records obtained from contact tracing. We also consider clustering of geographical areas based on contact rate and a higher average age. First and foremost, clustering facilitates equitable resource allocation and enables the targeted distribution of limited healthcare resources to priority groups based on susceptibility and geographic location. This strategic allocation ensures that areas and individuals at the highest risk receive timely and adequate attention. Additionally, clustering allows for targeted interventions by identifying high-risk geographic locations characterized by a higher contact rate and an older average age. The idea is that prioritized services for the most vulnerable populations contribute to the reduction of overall mortality and morbidity rates.

The priority group utilized to categorize all infected into different segments. In computing the priority group, two parameters are used: age and the number of recent contacts. Patients are categorized based on age into the following intervals: 1–18, 18–40, 40–50, 50–60, 60–70, 70–80, 80+. Each interval has a hazard ratio following Jahn et al. [23]. By multiplying the hazard ratio for age and contact rate, the importance of that cluster is obtained, and results are normalized to 100 %.

3. Literature review

The literature review comprises three parts: the first part covers the contagious disease testing problem, the second part addresses the TOP in healthcare applications, and the third part discusses approaches involving priorities and clustering patients in a pandemic situation.

Table 1 presents an overview of work related to the contagious disease testing problem, focusing on diverse vehicle types, solution procedures, and modeling approaches. Vehicle types are categorized as *heterogeneous* and *homogeneous*, while solution procedures encompass

Table 1
Literature on the contagious disease testing problem.

Author	Vehicle type		Solution procedure			Type of Uncertainty					Classification type		Type of Model		Objective Function					Period		Type of planning		Decision				Transportation phase	
	Heterogenous	Homogeneous	Exact	Heuristic	Meta Heuristic	Fuzzy	Scenario Base	Stochastic	Probabilistic	Deterministic	Clustering	Prioritizing	VRP	TOP	Covering	Cost	Serving	Profit	time	Single	Multi	Simulation	Mathematical model	Routing	Number of infected	Scheduling	Allocation	Deliver	Collect
Aringhieri et al. (2024)		•		•						•		•		•				•			•		•	•					•
Bish et al. (2024)			•							•						•					•		•		•		•	•	•
Navaei et al. (2023)		•	•		•					•			•		•	•					•		•	•			•		•
Chen et al. (2023)				•				•				•							•	•		•					•	•	
Thul and Powell (2023)		•	•					•								•					•		•				•		•
Wolfinger et al. (2023)		•			•			•					•			•					•		•	•			•	•	•
Vahdani et al. (2023)		•	•						•			•	•			•					•		•	•			•	•	
Arbabian and Rikhtehgar (2023)				•				•		•						•					•		•				•	•	
Colajanni et al. (2022)		•	•							•			•			•		•			•		•				•	•	
Shahnejat-Bushehri et al. (2022)		•		•						•			•			•				•			•			•	•	•	
Ozdemir et al. (2022)	•			•						•			•		•					•			•	•					•
Tlili et al. (2022)		•			•					•				•	•					•			•	•		•		•	•
Jahn et al. (2021)			•				•					•			•					•		•			•			•	
Foy et al. (2021)			•							•		•			•						•		•				•	•	
Rao and Brandeau (2021)			•						•			•			•					•		•					•	•	
Hosseini-Motlagh et al. (2021)		•		•		•		•									•				•		•	•			•		•
Santini. (2021)		•	•							•									•		•		•				•		•
Singgih et al. (2020)		•		•					•				•		•						•		•	•			•		•
Roozbeh et al. (2020)		•			•					•		•		•				•		•			•	•				•	
Jin and Thomas (2019)		•			•			•				•		•				•		•			•	•		•		•	
Yücel et al. (2018)		•	•							•		•		•				•		•			•	•				•	
This Research	•		•		•					•	•	•		•			•	•			•	•	•	•	•		•	•	•

exact, heuristic, and *metaheuristic* methods. Modeling techniques include *nonlinear, fuzzy,* and *scenario-based* models, addressing stochastic, probabilistic, and deterministic scenarios. Patient categorization through clustering and prioritizing, along with commodity types (single and multi), are examined alongside objective functions covering *cost, risk, profit,* and *time* optimization. Period and type of planning are differentiated as *single* or *multi*, encompassing *simulation, conceptual,* and *mathematical* models. Decision-making phases involve *routing*, consideration of the number of infected individuals, *scheduling*, and *allocation*, within the broader transportation phases of delivery and collection.

3.1. Contagious disease testing problem

Bish et al. [7] developed a multi-period mathematical model for swab test allocation, using real-case data from 2018 to 2021. Their objective was to minimize the overall cost of the supply chain while ensuring efficient test distribution. A key contribution of their study was the introduction of a multi-disease testing design model, which improved resource allocation.

A mathematical model for assigning vaccinations and diagnostic kits to patients has been presented by Thul and Powell [41]. The suggested model can capture passive information processes and active learning for diagnostic kit allocation. A parameterized rolling horizon approach is utilized to solve a multi-agent model. The case study is situated in Nevada, USA, and the findings show that the suggested model is resilient to resource scarcity. Another mathematical model is put out by Wolfinger et al. [46] to minimize the expenses associated with reopening testing sites and allocating the COVID-19 testing teams. One of their objectives is the routing and assignment of the suspicious cases to the laboratories. A large neighborhood search approach has been applied to solve the proposed model with two Austrian case studies for Upper Austria and the City of Vienna. Navaei et al. [30] proposed a distribution-location-allocation multi-period, multi-objective model for designing the testing kit supply chain. A key contribution of their work was incorporating sustainability into the supply chain design while leveraging the Internet of Things (IoT) for improved efficiency. They solved the model using NSGA-II and the Augmented ε-Constraint2 (AUGMECON2) method, and applied it to a real-case scenario in Iran. Arbabian and Rikhtehgar [1] focused on swab inventory management in the supply chain during a pandemic, aiming to determine the production capacity of swab suppliers. They solved the problem using a heuristic approach. A key contribution of their work was considering disruptions in the supply chain. The results compared stationary and stochastic demand, providing insights into how different demand scenarios affect the supply chain's performance.

Colajanni et al. [12] examined the swab test supply chain using unmanned aerial vehicles (UAVs) during the pandemic. Their objective was to minimize costs and maximize profits by optimizing the allocation between patient locations and laboratories. They focused on the delivery decision, leveraging UAVs to improve efficiency and reduce the time and cost associated with swab test transportation. A mathematical model for the assignment of Covid-19 tests is proposed by Shahnejat-Bushehri et al. [37]. They consider time windows and workload balancing. Their primary objectives are to reduce the overall penalty of the maximum deviation in the testers' workload and the costs of transportation. An adaptive large neighborhood search is used to solve the model. Ozdemir et al. [31] optimize the coverage level and reduce the walking distance for the retrieval of COVID-19 diagnosis kits for temporary testing locations. A total number of 99 hospitals in Seoul, South Korea, and Istanbul, Turkey are included in the model's analysis. Four heuristics – NodeSelection, Node Potential, Set Covering, and CoEC-NodePotentialr – are applied to solve the problem.

Santini [36] presented a mathematical model for swab test assignment to minimize patients' waiting time in the laboratory during a viral epidemic. Among the contributions of their model is the consideration of bottlenecks for testing, including the lack of chemical reagents. The proposed model has been tested with real data from Italy and solved using hierarchical multi-objective optimization.

A multi-stage mathematical model for the logistics management of the COVID-19 sample routing is presented by Hosseini-Motlagh et al. [22]. This involves the use of molecular, diagnostic, and antibody assays. By allocating appropriate test kits, their primary objective is to reduce the possibility of testing errors while also decreasing the expenses associated with providing the service. Fuzzy goal programming is presented as a solution to this problem. The case study of Tehran, Iran shows that the system's costs rise sharply with increased demand. The modeling of a mobile laboratory movement route for the purpose of collecting COVID-19 tests is surveyed by Singgih [38]. Their goal is to cover as much ground as possible for an Indonesian case study. They consider the capacity in each time period to collect the completed tests. The findings suggest that changing the allocation over different time periods may raise the coverage level to an advantageous level.

3.2. Team orienteering problem in medical problems

We model the problem at hand as TOP. For more information related to the general TOP and its variants, we refer to Wu et al. [47], Zhang et al. [49], Hanafi et al. [18], Lou et al. [28], and Lin and Vincent [27].

Aringhieri et al. [3] developed a team orienteering model for swab test collection in the city of Turin. They proposed a heuristic algorithm based on machine learning to optimize routing decisions while considering working time limitations, improving efficiency in test collection logistics. The logistics management for distributing and gathering COVID-19 testing samples is surveyed by Tlili et al. [42]. The primary objective of the model, which they view as a multi-origin-destination team orienteering problem, is to schedule and route ambulances. A case study for the City of Tunis is used to customize the mathematical model, which makes use of both hybrid genetic algorithms and memetic algorithms. According to the findings, the coverage maintains a stable level as demand rises. Roozbeh et al. [35] present a mixed integer model for routing and positioning in emergency response situations. They use the Cooperative Orienteering Problem with Time Windows, and their goal is to maximize the reward in each location. Finally, their model is solved using Adaptive Large Neighborhood Search, and the results are compared with simulated annealing. Jin and Thomas [24] present a mathematical model for intrahospital phlebotomist routing using the TOP for the routing and scheduling of hospital personnel. They consider uncertainty in rewards and service times. They customize their model for the University of Iowa Hospitals and Clinics, and finally, the model is solved with Variable Neighborhood Search. Yücel et al. [48] present a mathematical model for locating and routing mobile medical facilities based on the team orienteering problem. Their main goal is to maximize the difference between the sum of fully and partially covered scores and the total traveling cost. Finally, the proposed model for the City of Istanbul was tested and solved using CPLEX.

3.3. Priorities and clustering patients equitably in pandemics

Vahdani et al. [44] investigate the fair-split distribution of vaccines during the outbreak of COVID-19. Their main goal is to minimize the costs of distribution and transportation and to control the inventory of vaccines. The results show that prioritizing the elderly, regardless of model variables such as vaccine effectiveness, will lead to the greatest reduction in mortality. Chen et al. [11] applied Monte Carlo stochastic simulation to the testing kit supply chain, focusing on inventory allocation and delivery of testing kits. Their main contribution was the development of a fair allocation strategy. They solved the model using a Perturbation Search heuristic, which improved the efficiency of the allocation process. The results showed a fair solution while minimizing the feasible time for allocation.

Jahn et al. [23] prioritize individuals for the administration of the COVID-19 vaccine based on its limitations. This prioritization is determined by risk factors including age group and comorbidities such as cancer, diabetes, chronic liver disease, etc. Additionally, they utilize a dynamic agent-based model to compare various vaccine distribution strategies. The findings indicate that elderly individuals with multiple diseases should be given higher priority. Foy et al. [17] investigate the effectiveness of prioritizing people in the distribution of COVID-19 vaccines. Their prioritization is based on the age of people and their call rate, and therefore, older people with a higher call rate have higher priority. Their main goal is to maximize the level of vaccine coverage. Rao and Brandeau [34] address the allocation of limited vaccine to eligible individuals in New York state. They prioritize young people to reduce the likelihood of new infections and older people to reduce the likelihood of death, thus aiming to minimize the loss of life-years or quality-adjusted life years.

3.4. Research gaps

Many studies overlook the importance of prioritizing patients based on factors such as severity of symptoms or other relevant criteria. Our paper addresses this gap by introducing an approach to prioritize patients within the contagious disease testing management. We categorize patients based on specific disease backgrounds as either ’risky’ or ’standard’ patients. This facilitates the strategic allocation of limited healthcare resources and prompts consideration of equity in allocating resources to patients. Additionally, our paper considers the clustering of geographical locations based on contact rate and average age. This clustering enables the targeted allocation of resources to priority groups, ensuring timely attention to high-risk areas and individuals. Another research gap is the disregard for real-world scenarios in the context of COVID-19 conditions. In many existing studies, there is a notable absence of consideration for the intricacies and complexities that arise in actual pandemic situations. In this paper, we aim to address this gap by delving into the comparison between scenarios such as ‘quarantine’ and ‘non-quarantine’. Furthermore, we apply a real case study for Vienna, Austria, to ground our analysis in concrete, localized data, thus offering a more comprehensive understanding of a pandemic’s dynamics. Finally, waiting time and the minimization of unserved patients are often ignored. Therefore, in this paper, we maximize the score of the collected patients and minimize the number of unserved patients, along with routing and allocation decisions.

The most related paper to ours is the one by Wolfinger et al. [46]. One of their assumptions is that every suspected case can be covered. However, this assumption is not always feasible in the peak conditions of a pandemic. Hence, we use a TOP-based formulation, where it is not necessary to cover all patients, and coverage should be based on their priority. Furthermore, we employ clustering of geographical locations based on contact rate and a higher average age. To better reflect real-world scenarios, we consider multiple periods and divide patients into two categories: high-risk and standard. In each period, priority is given to high-risk patients, and unserved patients are transferred to the next period. Also, the function considered by Wolfinger et al. [46] aims to minimize the total costs, which includes the cost of using vehicles, the cost of routing, and the establishment of laboratories. Considering that in crisis situations, the priority is to minimize service time at the expense of considering costs, we thus consider the two goals of maximizing the reward of patients and minimizing the number of unserved patients.

4. Mathematical modeling

The problem at hand is conceptualized as a multi-period TOP. We model two echelon networks between depots, patient nodes, and laboratories. The main objectives are to maximize the total collected reward and to minimize the number of unserved patients. The reward is considered as a dynamic parameter. Patients are divided into risky and standard categories based on their disease background, and they have different service times. The score of standard patients is divided by the period number, while the score of risky patients is multiplied by the period number. Dividing the score by the period number is a method to reflect the impact of time on the patient’s condition appropriately, considering their risk category. Standard patients, by definition, have a more stable background. As time progresses, their condition is expected to stabilize or improve gradually due to ongoing treatment and management. Dividing their score by the period number reflects this stabilization or improvement over time, reducing the impact of their initial score as they progress through the periods. Risky patients have a more severe or volatile disease background. As time progresses, their condition may deteriorate or become more complicated without significant intervention. Multiplying their score by the period number reflects the increasing risk and potential complications that can arise over time if not managed aggressively. Given the limitations in testing capacity due to resource constraints such as testing kits and personnel availability, it becomes imperative to ensure that the number of collected specimens at any laboratory remains within its capacity bounds. The mathematical model is as follows.

4.1. Notations

Indices	
i,j	Index of patient’s nodes
t, t'	Index of time periods
v	Index of mobile testers
o	Index of depots
l	Index of laboratories
r	Index of priority groups
Sets	
N_c	Set of patient’s nodes $\{1, \ldots, NN\}$
O	Set of depots
V	Set of mobile testers
L	Set of testing laboratories
b	Number of all nodes
T	Time period
R	A set of priority groups
Parameters	
d_{it}	1, if there is demand of node i at period t, o.w 0
λ_r	The weight of a priority group r at period t. $\sum_{r=1}^{R} \lambda_r = 1$
C_l	Capacity of testing laboratory l
p_{it}	Score of patient i in period t
k_o	Number of mobile testers in depot o
t_{ij}	Travel time from i to j
t'_{oi}	Travel time from o to i
t''_{il}	Travel time from i to l
s_i	Service time for patient i
NN	Total number of the nodes
T_{max}	Maximum total travel time for a mobile tester
M	Big number
Variables	
x_{ijvt}	1, if node i is visited after node j by mobile tester v at period t, o.w 0
x'_{oivt}	1, if mobile tester v travels from depot o to node i at period t, o.w 0
x''_{ilvt}	1, if mobile tester v travels from node i to laboratory l at period t, o.w 0
y_{ivt}	1 if patient i is served by mobile tester v at period t, o.w 0
z_{lvt}	The number of demands from mobile tester v go to laboratory l at period t.
ud_{it}	The new demand received in node i for period t
NT_i	The number of periods that the demand of patient i has not been served
u_{ivt}	Auxiliary decision integer variable is used in the sub-tour elimination constraints

$$OBJ_1 = Max \sum_{t\in T} \sum_{r\in R} \sum_{i\in N_c} \sum_{v\in V} (\lambda_r p_{it} y_{ivt}) \tag{1}$$

$$OBJ_2 = Min \sum_{i\in N_c} \sum_{t\in T} \left(ud_{it} - \sum_{v\in V} y_{ivt} \right) \tag{2}$$

Constraints

$$\sum_{v\in V} \sum_{i\in N_c} x'_{oivt} \le k_o \qquad o \in O, t \in T \tag{3}$$

$$\sum_{i\epsilon N_c} x'_{oivt} \leq 1 \qquad o \in O, v \in V, t \in T \tag{4}$$

$$\sum_{o\epsilon O} x'_{ojvt} + \sum_{i\epsilon N_c} x_{ijvt} = y_{jvt} \qquad j \in N_c, v \in V, t \in T \tag{5}$$

$$\sum_{o\epsilon O} x'_{ojvt} + \sum_{i\epsilon N_c, i\neq j} x_{ijvt} = \sum_{l\epsilon L} x''_{jlvt} + \sum_{i\epsilon N_c,\ i\neq j} x_{jivt} \qquad j \in N_c, v \in V, t \in T \tag{6}$$

$$u_{ivt} - u_{jvt} + NN \times x_{ijvt} \leq NN - 1 \qquad i,j \in N_c, v \in V, t \in T \tag{7}$$

$$ud_{it} = d_{it} + \sum_{t'=1}^{t-1} \left(ud_{it'} - \sum_{v\in V} y_{ivt'} \right) i \in N_c, t \geq 2 \tag{8}$$

$$ud_{i1} = d_{i1} \ i \in N_c \tag{9}$$

$$\sum_{v\in V} y_{ivt} \leq d_{it} i \in N_c,\ \ t \in T \tag{10}$$

$$NT_i = \sum_{t\in T} \left(d_{it} - \sum_{v\in V} y_{ivt} \right) i \in N_c \tag{11}$$

$$\sum_{i\epsilon N_c} \left(t'_{oi} + s_i \right) \times x'_{oivt} + \sum_{i\epsilon N_c} \sum_{j\epsilon N_c} \left(t_{ij} + s_j \right) \times x_{ijvt} + \sum_{l\epsilon L} \sum_{i\epsilon N_c} t''_{il} \times x''_{ilvt} \leq T_{max} \qquad o \in O, v \in V, t \in T \tag{12}$$

$$\sum_{v\in V} z_{lvt} \leq C_l \qquad l \in L, t \in T \tag{13}$$

$$z_{lvt} = \sum_{i\epsilon N_c} y_{ivt} \times \sum_{i\epsilon N_c} x''_{ilvt} \qquad l \in L, v \in V, t \in T \tag{14}$$

$$x_{ijvt}, x'_{oivt}, x''_{ivt},\ y_{ivt} \in \{0,1\}$$

$$NT_i, ud_{it}, z_{lvt} \in integer \qquad i,j \in N_c, l \in L,\ o \in O, v \in V, t \in T \tag{15}$$

The objective function (1) seeks to maximize the overall collected patient scores. Based on various factors such as age and number of recent contacts, a risk group with greater weight receives higher priority for the service. Therefore, λ_r represents the *susceptibility* of each risk group based on the priority group weighting. The second objective function (2) minimizes the number of unserved patients. Constraint (3) sets the upper bound of the number of mobile testers. Constraint (4) guarantees that at most one patient can be served from each depot each period. In other words, the output from each depot in each tour and period should be at most one. Constraint (5) depicts the relationship between the allocation variable and the routing variables in each period. This constraint indicates whether patient j is visited by mobile tester v in period t or not. In other words, this constraint shows the inputs to node j. If there is an input to this node, the variable y_{jvt} must be equal to 1. In fact, if the variable y_{jvt} becomes equal to 1, either the mobile tester goes from the depot directly to node j ($\sum_{o\epsilon O} x'_{ojvt} = 1$), or it goes to node j from another node ($\sum_{i\epsilon N_c} x_{ijvt} = 1$). Constraint (6) serves as a balance constraint, ensuring equilibrium between input and output at patient nodes. Constraint (7) utilizes the subtour elimination method by Bektaş and Gouveia [4]. This follows the Miller–Tucker–Zemlin (MTZ) subtour elimination constraint that generally used for classic vehicle routing problems (VRPs). This constraint ensures that if patient i is visited before patient j, the position of j in the tour must be greater than the position of i. If $x_{ijvt} = 1$ (i.e., the mobile tester travels directly from i to j), then u_{jvt} must be greater than u_{ivt} by at least 1. The term $NN \times x_{ijvt}$ allows the constraint to be relaxed if i and j are not consecutive in the tour. Constraint (8) calculates the updated demand for each period in a way that ensures it equals the initial demand for the period plus the uncovered demand. Constraint (9) specifies that the updated demand in the first period equals the initial demand. Constraint (10) indicates that if there is a demand for a patient in a given period, at most one mobile tester should serve that patient. Conversely, if the patient does not have a demand in a given period, none of the mobile testers should serve the patient. Constraint (11) calculates the number of periods during which the demand for patient i has not been met. This is determined by the difference between the number of new patients demands in each period and the number of demands served. Constraint (12) ensures that the time taken by each mobile tester to complete their tour does not exceed the maximum allowed time, denoted as T_{max}. Constraint (13) indicates that the sum of collected specimens at testing laboratory should be less than or equal to its capacity, denoted by C_l. Constraint (14) specifies the relationship between the allocation and routing variables. If the mobile tester does not visit the hospitals ($\sum_{i\epsilon N_c} x''_{ilvt} = 0$), the patient is not assigned to them. However, if the mobile tester visits the hospitals ($\sum_{i\epsilon N_c} x''_{ilvt} = 1$), the number of allocated patients is equal to the total number of patients that the mobile testers transferred to that hospital. Given that this constraint is nonlinear, its linearization is provided in Appendix E. Constraint (15) describes the type of variables in the model.

5. Solution methodology

In this research, we apply the Epsilon-Constraint Method (ECM) to solve the proposed model exactly for small instances. Furthermore, we propose MOVNS and NSGA-II to solve the model for large instance sizes.

5.1. Epsilon-constraint method

The ECM is commonly used in multi-objective optimization problems and it is one of the most applicable exact multi-objective methods [16]. Suppose that $f_j(x)$ represents the *jth* objective function. If $j \in \{1, \ldots k\}$, then the multi-objective optimization transforms into the following single-objective optimization. Thus, S is a feasible solution in the solution space, and ε_i equals the upper bound of the *ith* objective function.

$$Min\ f_j(x) \qquad \forall j \in \{1, \ldots k\} \tag{16}$$

s.t.

$$f_i(x) \leq \varepsilon_i \qquad \forall i \in \{1, \ldots k\},\ i \neq j \tag{17}$$

$$x \in S \tag{18}$$

In our paper, the implementation of the ε-constraint method is as follows. First, we choose the first objective function as our main objective function. Next, we solve the problem for each of the objective functions individually to obtain their optimal values. We then divide the interval between the optimal values of the sub-objective functions into a predetermined number of buckets and create a table of values for ε_2, …, ε_n. Subsequently, we solve the problem using the original objective function with each of these ε-values. Finally, we report the Pareto solutions found. Furthermore, by adjusting the values on the right side of the constraints (ε_i), we obtain efficient solutions for the problem. Therefore, the second objective function is considered as a constraint, and its value is considered to be less than ε_1. Then, we introduce a new constraint for that objective function with a lower bound of ε_2, denoted as $OBJ_2 \leq \varepsilon_2$. If the obtained solutions are not Pareto-optimal, we adjust the value of ε_2 and repeat the optimization process until a satisfactory set of Pareto-optimal solutions has been obtained (see Eqs. (19)–(21).

$$OBJ_1 \tag{19}$$

s.t.

$$OBJ_2 \leq \varepsilon_2 \tag{20}$$

$$Constraint\ 3 - 15 \tag{21}$$

5.2. Multi-objective variable neighborhood search (MOVNS)

The Variable Neighborhood Search (VNS) proposed by Mladenović & Hansen [29] is a metaheuristic approach used for solving optimization problems heuristically. It is based on systematically changing the neighborhood structure during the search [19]. It shifts from its present location in the solution space to a new one if an enhancement has occurred or specific acceptance criteria are satisfied [8].

So far, many successful applications of the multi-objective VNS approach have been recorded [15,25]. Geiger [20] introduced the first multi-objective VNS algorithm, aiming to address the Permutation Flow Shop Scheduling Problem by minimizing various combinations of criteria. The adaptation of the VNS framework to multi-objective optimization problems introduces significant innovations in how solutions are conceptualized and improved. By redefining the solution as an approximate set of efficient points found during the search process, the VNS can effectively handle multiple objectives simultaneously. Hence, the VNS can be seen under the umbrella of population-based metaheuristics, which traditionally maintain a set of solutions akin to the Pareto front. By viewing the approximate Pareto front as the incumbent solution, we can naturally extend various VNS variants to multi-objective problems. Fig. 1 shows the pseudocode of our proposed MOVNS algorithm. This approach is applicable and well-established for solving the TOP. The VNS not only addresses the complexities of optimizing multiple routes to maximize the total score but also leverages adaptive strategies to balance exploration and exploitation throughout the search process, thus yielding high-quality solutions to the TOP [43].

The process starts by generating an initial solution using Variable Neighborhood Descent (VND), and this solution is set as the best-found solution, denoted as X^*. The main loop of the algorithm continues until a predefined maximum number of iterations has been reached. Within this loop, the algorithm iterates through each neighborhood structure, applying a shaking mechanism to perturb the current solution and exploring the new solution X that results. If X dominates X^* (meaning X is better in at least one objective and no worse in others), X^* is updated with X, and the search process reverts to the first neighborhood structure. If not, X is stored in a repository of non-dominated solutions, and the algorithm proceeds to the next neighborhood structure. In addition to the core MOVNS operations, the algorithm incorporates mechanisms for managing and selecting multiple non-dominated solutions. Once the inner loop over neighborhood structures is complete, the algorithm forms a Pareto frontier from the stored non-dominated solutions. It then divides this frontier into grid cells based on objective values, calculates the probability of each cell, and uses a roulette wheel selection mechanism to choose a solution from these grid cells. The selected solution is then used to update X^*. If no significant improvement is observed according to predefined criteria (e.g., SM, MID and HV), the algorithm terminates early. This approach balances exploration and exploitation by iteratively refining solutions and incorporating diversity through grid-based selection and non-dominated sorting.

5.2.1. Neighborhood structures

First, the MOVNS generates an initial solution. Fig. 2 shows an instance of the multi-period TOP formulated above along with a feasible solution. The instance includes 10 patients (nodes 1 to 10), two depots (nodes 11 and 12), one laboratory (node 13), and two periods ($t = 1,2$). Additionally, two mobile testers provide service in two periods. The routes marked with blue are for the first period, and the routes marked with red are for the second period. The demand of nodes 6, 1, 4, 2, 10, and 7 is for the first period, and the demand of nodes 8, 3, 5, and 9 is for the second period.

There are three types of strings in the designed neighborhood. The first string, called the *visiting order string,* has a length of $\sum_{i\in N_c}\sum_{t\in T}d_{it} + (|t|-1)$, which represents the visiting order for all periods and clusters and $|t|$ represents the total number of periods. In this string, the visiting order of each period is separated from other periods by a 0. To examine

```
Generate the initial solution using VND
X*=best solution
while (iter < maximum number of iterations)
    k = 1;
    while (k < k_max)
        Apply Shaking
        X=the best solution from shaking
        if the objectives function of X dominates X*
          X* = X
        Else if X* do not dominate X
          store non-dominated solutions in repository
          k = k + 1
        Endif
   Endwhile
   Form pareto frontier based on non-dominated sorting from repository
   Define grid boundaries and resolution
   Assign each Pareto solution to a grid cell based on objective function values
   Calculate probability of each cell
   Create a roulette wheel with segments sized according to the probabilities of each grid cell,
   then choose a grid cell
   Select a solution from the chosen grid cell
   Update X*
   if improvement criteria (MID,SM,HV) is not better in most iterations
      break
   endif
   iter = iter + 1
end while
```

Fig. 1. Pseudocode of MOVNS algorithm.

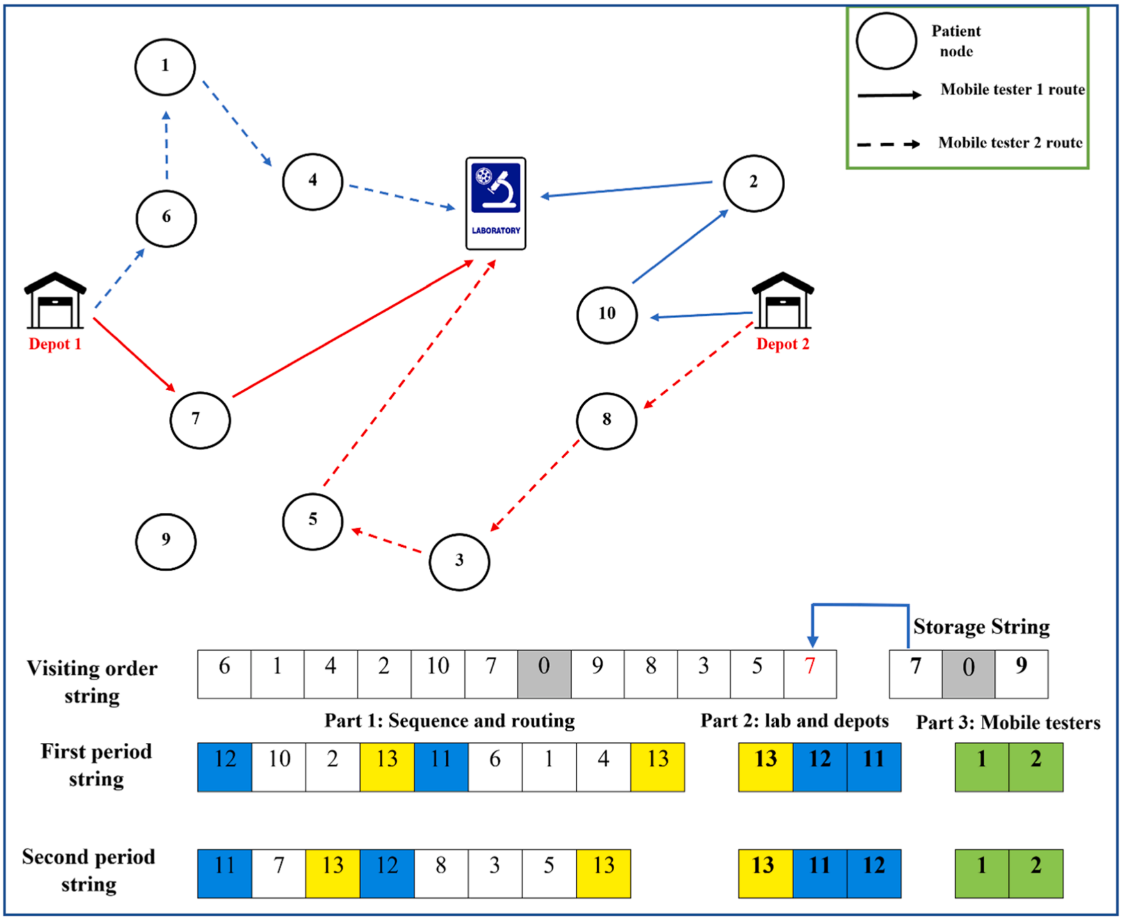

Fig. 2. Problem instance together with a feasible solution.

the solution space more effectively, we randomly select a series of points instead of choosing the entire solution space from the visiting order string. This approach allows the algorithm to produce more diverse solutions and explore the solution space better. Additionally, it helps reach a locally optimal solution more efficiently. The second neighborhood is called the *period string,* and it is divided into three parts. The first part of this neighborhood consists of a string that represents the routing and movement sequence of mobile testers. The first cell of this string is the depot node, the next cells are the locations of patients or demand points. Due to the existence of only one laboratory in the problem, the first cell of the second string is the laboratory, and the next cells in the second string are depots. The third part of the string is the number of the mobile testers, and the number of cells in this part is always one less than the second string. Because by removing the first cell of the second string, each depot is connected to its corresponding mobile tester in the third part.

Consider this example for better understanding: In period 1, mobile tester 2 (the first cell of the third part) starts moving from depot 12 (the second cell of the second part) and after visiting patients 10 and 2, it delivers the specimens to laboratory 13 (the first cell of the second part). Next, mobile tester 1 (the second cell of the third part) moves from depot 11 and goes to laboratory 13 after visiting patients 6, 1, and 4. The point is that at the end of each period, the cells related to the location of the patients are compared with the visiting order string cells. If the visiting order cells exceed the serviced cells, the surplus cells (representing unserved patients) are stored in a string called the *storage string*. The space between periods is marked with cell 0. This string is responsible for transferring the surplus cells to the demand string for the next period. Here, because the visiting order of node 7, which was for the first period, was not served, this node is stored in the storage string and transferred to the second period. In the second period, mobile tester 1 moves from depot 11 and after visiting node 7, delivers the specimens to the laboratory (13). Also, mobile tester 2 moves from depot 12, and after visiting nodes 8, 3, and 5, it delivers the samples to the laboratory (13), respectively. In this period, the cells related to the location of the patients are again compared with the cells of the visiting order string, and cell 9 is stored as an unserved cell in the storage string. Given that there were only two periods in this instance, this node is reported as an unserved patient.

5.2.2. *Variable neighborhood descent (VND)*

After the initialization phase, the initial solution is improved in different steps. At the core of our MOVNS implementation, there is a VND procedure combining *2-Opt* and *3-Opt*. The steps of the proposed VND algorithm are shown in Fig. 3. The process begins with the creation of an initial solution, after which the 2-Opt operator is applied to generate all possible improved solutions. These solutions are then evaluated and a Pareto frontier is established through non-dominated sorting within the repository. To manage the search space effectively, grid boundaries and resolution are defined, and each Pareto solution is assigned to a grid cell based on its objective values. The probability of selecting each of the Pareto points in each cell is calculated, and a roulette wheel is constructed with segments proportional to these probabilities. The solution X^* is selected from the chosen grid cell, and the 3-opt operator is applied to further refine the solution. This process of forming the Pareto frontier, defining grid boundaries, and recalculating probabilities is repeated to continuously improve the solution. The heuristic leverages the grid and roulette wheel mechanism to systematically explore and exploit the solution space, optimizing the solution through a sequence of neighborhood changes.

Step 1	Create an initial solution
Step 2	Apply 2-opt operator to create all possible improved solution,
Step 3	Form pareto frontier based on non-dominated sorting from repository
Step 4	Define grid boundaries and resolution
Step 5	Assign each Pareto solution to a grid cell based on objective function values
Step 6	Calculate probability of each cell
Step 7	Create a roulette wheel with segments sized according to the probabilities of each grid cell, then choose a grid cell
Step 8	Select a solution X^* from the chosen grid cell
Step 9	Apply 3-opt operator to create all possible improved solution
Step 10	Form pareto frontier based on non-dominated sorting from repository
Step 11	Define grid boundaries and resolution
Step 12	Assign each Pareto solution to a grid cell based on objective function values
Step 13	Calculate probability of each cell
Step 14	Create a roulette wheel with segments sized according to the probabilities of each grid cell, then choose a grid cell
Step 15	Select a solution X^* from the chosen grid cell

Fig. 3. Proposed VND Algorithm.

5.2.3. *Change neighborhood (shaking)*

In this study, we utilize the *cross-exchange* method, a robust neighborhood structure widely recognized for addressing multi-period VRPs, initially proposed by Taillard et al. [39]. The essence of cross-exchange lies in interchanging two segments from different routes while maintaining their respective sequences. Let us denote NR as the number of patients in a given route R. In the neighborhood, the maximum length of a segment is given by $min(NR,\ k)$. The minimum segment length is set at 1 for the first route and 0 for the second route, enabling the possibility of relocating nodes between routes. However, a challenge arises when a patient is randomly relocated to a route with insufficient remaining working hours, resulting in an infeasible solution. To mitigate this issue, patients are transferred only to routes where adequate working hours are available for their travel.

Within each neighborhood, all possible segments of equal lengths are considered. Following a cross-exchange operation, new routes are generated. However, due to constraints such as working hour limits and time restrictions, these routes may lead to infeasible solutions. These issues are addressed during the iterative improvement phase. By ensuring that segment lengths are constrained to 1 and 0, we maintain nested neighborhoods consistently. This approach encourages the exploration of shorter segments, focusing on refining the solution space and aligning more closely with the current optimal solution. The way this operator works is that first, two nodes (X_1, X'_1) and (Y_1, Y'_1) are cut from route 1, and (X_2, X'_2) and (Y_2, Y'_2) are also cut from route 2. Then parts $X'_1 - Y_1$ and $X'_2 - Y_2$ which include an arbitrary number of patients, are swapped, and new routes are generated as (X_1, X'_2), (Y_2, Y'_1), (X_2, X'_1) and (Y_1, Y'_2) .

To handle multiple periods, we designate the demand of patients entering the calculation from period 1 onward as *new request* and the demand of period 1 as *current request*. Therefore, to handle the new request, we utilize the second type of operator, called *Exchange 1,* which involves selecting a route and a position within this route and removing the amount p unserved patients found at this position. To calculate the value of p, the amount of travel time and service time of the patients are added cumulatively until this value exceeds the working hour limitation. Therefore, we transfer patients whose cumulative time exceeds the working hour limitation (unserved patients) to the next period and remove them from the current period. For example, if in the third period, according to the number of available mobile testers, the cumulative time of the 20th patient exceeds the working hour limitations, we transfer this patient to the fourth period and remove this request (which is equivalent to $p = 1$ in this case) from the third period. Next, the number of q patients (which is equal to the updated request ud_{it}) is replaced by those demands (d_{it}) (please see Constraint 8).

To prioritize patients, we introduce the *Exchange 2* operator, which handles patient selection. This operator involves selecting a route and a position on it, and then replacing a certain number of requests based on priority within this route. The way this operator works is that if there are new risky requests within a period, these requests replace the standard requests, and the standard requests are transferred to the next period in case of a working hour limitation as long as there are no risky requests.

If, in the stage of Exchange 1, the patient transferred to the next period is a risky patient, we need an Exchange 2 operator. The Exchange 2 operator checks the patients assigned to the route in the previous period if the patient transferred to the next period is of a risky type. If all patients were of the risky type, then no change is needed. However, if there is a standard patient, the risky patients of the next period (who were unserved patients of the previous period) will replace these standard patients. This process is repeated until either all risky patients return to the previous period or there are no more standard patients in the previous period.

Fig. 4 shows the pseudo-code for shaking. Steps 1 to 8 display the code for the Cross-Exchange operator. The Exchange 1 operator and the Exchange 2 operator are coded in steps 9 to 13 and 14 to 20, respectively. Finally, steps 21 to 27 show the Iterative Improvement.

Continuing with this operator, the iterative improvement approach is balanced, aiming to manage the new demand in subsequent periods to prevent the route from becoming infeasible.

5.2.4. *Iterative improvement*

At this point, the local solutions obtained in the previous step are improved. Therefore, to enhance the solutions of the operators, Exchange 1, Exchange 2 and Cross-Exchange, we employ the *3-opt approach* to optimize the paths that became infeasible in the previous step. The 3-opt operator, introduced by Lin [26], works by considering three edges of a route and exploring alternative ways to reconnect them while preserving the overall tour structure.

By evaluating the objective functions, the algorithm can determine whether the move improves the current solution. The change made in this approach is that the movement chain is limited to no more than three nodes. Due to the working hour limitations that can render the algorithm infeasible, we only consider moves that do not invert this chain and are feasible. This strategy represents the initial improvement approach that restarts iteratively immediately following each improving move.

5.2.5. *Acceptance criterion*

We decide whether to accept the solution obtained from the new neighborhood. This may involve considering factors such as the solution

```
Shaking
% Cross-Exchange operator
for each period:
      Choose two random segments (subsets of elements) from the current random solution
      Ensure the segments chosen are not overlapping or adjacent
      Swap the positions of these two segments
End
for each period
  for each route
        Calculate cumulative travel and service time
% Exchange 1 operator
        if cumulative time exceeds working hour limitation:
            Remove number of p patients from current period
            Transfer the patients whose time exceeds the working limitation to the next period.
            Update request of q (i.e., q = p + d_it)
        End
% Exchange 2 operator
         if the patient transferred to the next period (i.e., p patient) are risky patients
             if the remaining patients are standard
                 Exchange risky patient to standard patient
            End
        End
   End
End
% Iterative Improvement (i.e., Local solutions improvement for Cross-Exchange Exchange 1
and Exchange 2)
Best solution=the best solution obtained after shaking
for the maximum improvement iteration
       Apply 3-opt to the best solution
       if new random solution better than best solution
           Best solution=new random solution
       End
End
```

Fig. 4. Pseudocode of shaking algorithm.

quality, improvement, or other criteria specific to the problem. We apply a multi-objective function improvement criterion in our research, i.e., a solution is accepted only if it is non-dominated relative to the current set of solutions. This means the solution does not have any other solutions that are better in all objectives and at least as good in one objective.

5.2.6. Stopping criterion

In this study, two types of stopping criteria are used. If any of these two conditions occur, the algorithm will stop. These two criteria are as follows:

- We stop after the determined number of iterations.
- We stop when a determined number of iterations have been performed without improvement.

If the stopping criterion is not met, we start the initial improvement process again.

5.3. Non-dominated sorting genetic algorithm (NSGA-II)

NSGA-II is an updated version of traditional Genetic algorithms, specifically designed for multi-objective optimization. First proposed by Deb et al. [14], NSGA-II has become a widely recognized and applicable method due to its efficiency and ability to generate a diverse set of solutions. The algorithm utilizes non-dominated sorting to rank individuals in the population, coupled with crowding distance to maintain diversity in the Pareto front. This makes NSGA-II particularly effective for solving problems with multiple conflicting objectives, as it produces a set of Pareto-optimal solutions [21].

Fig. 5 presents the NSGA-II pseudo-code, outlining key steps: initialization (steps 1–4), non-dominated sorting and ranking (steps 5–6), crowding distance calculation (steps 7–10), crossover and mutation (steps 11–20), merging and sorting of populations (steps 21–23), and final selection for the next generation (steps 24–28).

5.3.1. Chromosome representation

The proposed chromosome structure is similar to the MOVNS structures shown in Fig. 2. The first, known as *the visiting order chromosome*, has a length of $\sum_{i\in N_c}\sum_{t\in T}d_{it} + (|t| - 1)$ and separates the visiting order of each period using a 0. The second, *the period chromosome*, is divided into three parts: the first represents the routing and movement sequence of mobile testers, the second starts with a laboratory followed by depots, and the third indicates the number of mobile testers, with one fewer cell than the second string to maintain depot-to-tester connections. At the end of each period, patient locations are compared with the visiting order chromosome. Any unserved patients are stored in a separate string called the *storage chromosome* (See Fig. 2).

5.3.2. Crossover operator

In this paper, a double-point crossover operator is employed in

```
%Initialize Population
For pop = 1 to Npop
   Generate an initial population P of size N randomly
   Calculate the objective functions for each individual in P
End
% Non-dominated Sorting
Divide the population into different Pareto fronts using non-dominated sorting
Assign ranks to chromosomes based on their front levels
% Compute Crowding Distance
For each front
   Calculate the crowding distance for each chromosome based on Equation (22)
   Sort the solutions based on crowding distance
End
For it = 1 to Maxit
   % Apply Crossover
   For Popc = 1 to round(Pc * Npop)
      Select two chromosomes from the parent population using the roulette wheel selection
method
      Perform double-point crossover to generate offspring
      Calculate the objective functions for the offspring
   End
   % Apply Mutation
   For Popm = 1 to round(Pm * Npop)
      Apply the mutation operator based on Subsection 5.3.3
      Evaluate the objective functions for the mutated chromosome
   End
   % Combine Parent and Offspring Populations
   Merge the parent population P with the offspring from crossover and mutation
   % Non-dominated Sorting
   Divide the new population into different Pareto fronts using non-dominated sorting
   Assign ranks to chromosomes based on their front levels
   % Compute Crowding Distance
   For each front
      Calculate the crowding distance for each chromosome based on Equation (22)
      Sort the solutions based on crowding distance
   End
   % Select Next Generation
   Remove excess sorted chromosomes exceeding Npop
End
```

Fig. 5. Pseudocode of NSGA-II algorithm.

NSGA-II. This operator involves randomly selecting two parent chromosomes and determining two crossover points within each. The genes between these points are then swapped, producing two offspring that inherit traits from both parents. We applied the double-point crossover to the visiting order chromosome, period chromosome, and storage chromosome. For instance, Fig. 6 illustrates the double-point crossover applied to the visiting order chromosome.

5.3.3. Mutation operator

In this paper, we use the reverse operator for mutation. The reverse mutation involves reversing a segment of the chromosome, altering the sequence of genes. The steps for mutation are as follows: first, select a segment within the chromosome; second, reverse the selected segment; and finally, replace the original segment with the reversed one. We apply the reverse operator mutation to the visiting order chromosome, period chromosome, and storage chromosome. For instance, Fig. 7 illustrates the reverse mutation applied to the visiting order chromosome. We assumed that the number of cells to be mutated should be half of the total number of cells. The obtained value is rounded up. For the visiting order chromosome, the number of mutated cells would be 6. Therefore, 6 cells are randomly selected, and the selected segment is reversed.

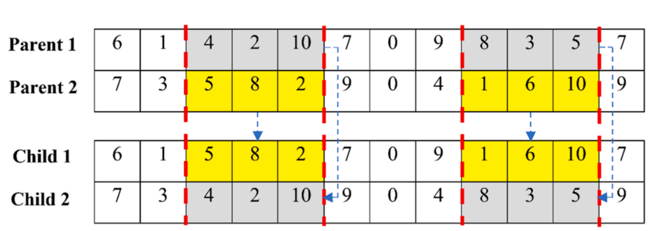

Fig. 6. Double-point crossover for visiting order chromosome.

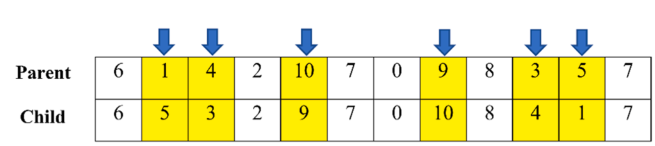

Fig. 7. Reverse mutation for visiting order chromosome.

5.3.4. Crowding distance

In this paper, we use the crowding distance sorting mechanism in NSGA-II to maintain diversity among solutions in the population while exploring the Pareto front. The crowding distance (CD_i) for a solution i in a bi-objective optimization problem is defined as:

$$CD_i = \frac{f_{i+1}^{(1)} - f_{i-1}^{(1)}}{f_{max}^{(1)} - f_{min}^{(1)}} + \frac{f_{i+1}^{(2)} - f_{i-1}^{(2)}}{f_{max}^{(2)} - f_{min}^{(2)}} \tag{22}$$

where $f^{(1)}$and $f^{(2)}$represent the two objective functions (1) and (2). The terms $f_{i+1}^{(m)}$ and $f_{i-1}^{(m)}$ denote the objective values of the next and previous solutions in the sorted list for each objective m, respectively. Additionally, $f_{max}^{(m)}$ and $f_{min}^{(m)}$are the maximum and minimum values of the objective m within the front. To ensure the selection of boundary solutions, they are assigned an infinite crowding distance.

5.3.5. Stopping criterion and parameter tuning

For the stopping criterion, two conditions are used to determine when the algorithm should stop. The algorithm stops either after a predetermined number of iterations or when a specific number of iterations have been completed without any improvement. If neither criterion is met, the initial improvement process is restarted.

For parameter tuning, we follow the approach of Navaei et al. [30]. They implemented the NSGA-II algorithm for a multi-period, multi-objective mathematical model for the distribution of testing kits. To adjust the parameters, they used the Taguchi method [40]. So, based on Navaei et al. [30], we set the population size (*Npop*) to 200, the crossover rate (*Pc*) to 0.4, the mutation rate (*Pm*) to 0.3, and the maximum number of iterations (*Maxit*) to 450.

6. Computational experiments

Section 6 is organized into two main subsections. In Section 6.1, the we evaluate or solution methods validation, including the computational setup and the evaluation of model performance at small, medium, and large scales using artificial data. In Section 6.2, the computational setup and managerial insights for the case study are described.

6.1. Validation of solution method

6.1.1. Computational setup

To evaluate the effectiveness of the multi-objective algorithms, three assessment metrics are employed:

- Spacing Metric (SM): measures the standard deviation of the distances between solutions of the Pareto front [50].

$$SM = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N-1}(\bar{d} - d_i)^2} \tag{23}$$

where d_i is equal to the Euclidean distance between two adjacent Pareto points, and $\bar{d}$ is the average Euclidean distance. The lower the value of this metric, the smaller the distance between the Pareto points and the better the performance of the algorithm. For example, if this value is exactly equal to zero, the distance between neighboring Pareto's is the same. Also, N is the number of Pareto points.

- Mean ideal distance (MID): measures the convergence rate of Pareto fronts up to a certain point (0, 0) [51].

$$MID = \frac{\sum_i^n \sqrt{\left(\frac{f_{1i}-f_1^*}{f_{1.global}^{max}-f_{1.global}^{min}}\right)^2 + \left(\frac{f_{2i}-f_2^*}{f_{2.global}^{max}-f_{2.global}^{min}}\right)^2}}{n} \tag{24}$$

Here, f_{ji} equals the *jth* objective function for the *ith* Pareto front. Also, $f_{j.global}^{min}$ and $f_{j.global}^{max}$ are the minimum and maximum of the *jth* objective function among the Pareto points. Additionally, f_1^* and f_2^* are the ideal values of the first and second objectives, respectively and n is the number of Pareto points. According to this definition, the algorithm with a lower value of the MID has a better performance.

- The Hypervolume metric (HV):

The hypervolume indicator, first introduced by Zitzler and Thiele [52], is a robust measure in multi-objective optimization that assesses solution quality by quantifying the volume of the objective space dominated by a set of solutions relative to a reference point. This metric aids in evaluating how closely a solution set approximates the Pareto front, guiding optimization towards optimal and diverse solutions [13]. Its unique ability to capture both convergence and diversity combined with properties like scale independence and sensitivity to solution efficiency, positions it as a premier metric for comparing diverse optimization algorithms. This indicator is defined as Eq. (25):

$$H\left(v^{(1)},\ldots,v^{(\mu)};R\right) = \Lambda_k\left(\bigcup_{i=1}^{\mu}\left[v^{(i)},R\right]\right) \tag{25}$$

with $v^{(1)},\ldots,v^{(\mu)} \in \mathbb{R}^k$ as non-dominated set and $R \in \mathbb{R}^k$ such that $v^{(i)} < R$ for all $i = 1,\ldots,\mu$. The reference point ($R = (r_1, r_2, \ldots, r_k)^T$) can be taken as the vector of the worst values of the objective functions. Also, $\Lambda_k(.)$ represents the Lebesgue measure in $\mathbb{R}^k$. Due to the two objectives ($k = 2$) of our model, the calculation of (25) reduces to Eq. (26):

$$H\left(v^{(1)},\ldots,v^{(\mu)};R\right) = \left[r_1 - v_1^{(1)}\right].\left[r_2 - v_2^{(1)}\right] + \sum_{i=2}^{\mu}\left[r_1 - v_1^{(i)}\right].\left[v_2^{(i-1)} - v_2^{(i)}\right] \tag{26}$$

6.1.2. Evaluation of model performance

In this section, MID, SM and HV metrics are employed to compare the effectiveness of the MOVNS and NSGA-II with the exact solution approach. Therefore, for each sample, these three metrics are reported in Table 2. We consider 30 samples in three different sizes: S1 to S10 ("small scale"), M1 to M10 ("medium scale"), and L1 to L10 ("large scale"). The lower the values of MID and SM, the better the algorithm has performed, and the larger the value of the hypervolume metric, the better.

Considering that the calculated MID and SM metric values are below 1, we can trust the quality of the solutions. Furthermore, the comparison of solution times for the proposed algorithms shows that the MOVNS algorithm outperforms both ECM and NSGA-II, making it the most efficient choice for solving the case study. HV values consistently highlight the quality and efficiency of the solutions, with higher values indicating superior performance. For instance, in instances S1 through S10, the HV values for the ECM, NSGA-II, and MOVNS methods are closely matched, indicating that the algorithms perform comparably well in terms of approximating the Pareto front. Also, the metric values show that MOVNS performs better than NSGA-II in all metrics, including CPU time. Moreover, the ECM is unable to solve the mathematical model due to the long solution time resulting from the NP-hardness of the model; therefore, the model on a large scale is solved only with the MOVNS approach. This trend indicates that MOVNS has an edge in delivering higher convergence and more diverse solutions for more complex scenarios compared to NSGA-II. For more information on the

Table 2
Assessment metrics for the effectiveness of MOVNS, NSGA-II and exact solution approach.

Instances	*MID*			*SM*			HV			CPU time (s)		
	ECM	*MOVNS*	NSGA-II	*ECM*	*MOVNS*	NSGA-II	*ECM*	*MOVNS*	NSGA-II	*ECM*	*MOVNS*	NSGA-II
S1	0.136	0.136	0.136	0.038	0.038	0.038	0.823	0.823	0.823	3	3	3
S2	0.129	0.133	0.133	0.031	0.035	0.037	0.845	0.841	0.841	4	4	4
S3	0.142	0.145	0.149	0.059	0.061	0.061	0.811	0.807	0.805	5	4	5
S4	0.220	0.223	0.227	0.078	0.079	0.083	0.852	0.850	0.844	155	23	28
S5	0.208	0.213	0.218	0.086	0.092	0.095	0.836	0.831	0.822	188	31	39
S6	0.260	0.265	0.270	0.105	0.109	0.114	0.818	0.811	0.805	356	44	47
S7	0.283	0.290	0.293	0.098	0.106	0.115	0.854	0.845	0.842	426	65	71
S8	0.291	0.296	0.302	0.120	0.125	0.131	0.845	0.832	0.821	505	80	88
S9	0.311	0.314	0.317	0.167	0.173	0.175	0.861	0.842	0.838	588	106	118
S10	0.342	0.351	0.363	0.187	0.200	0.205	0.855	0.851	0.847	912	132	143
M1	0.407	0.414	0.422	0.312	0.318	0.329	0.764	0.755	0.750	2018	191	205
M2	0.343	0.354	0.355	0.210	0.221	0.228	0.741	0.737	0.724	2790	237	251
M3	0.186	0.194	0.209	0.388	0.397	0.410	0.757	0.754	0.737	3951	317	331
M4	0.457	0.471	0.477	0.157	0.162	0.180	0.732	0.725	0.711	5548	511	535
M5	0.309	0.321	0.323	0.173	0.187	0.198	0.754	0.747	0.744	6779	719	743
M6	0.215	0.218	0.224	0.281	0.286	0.292	0.719	0.711	0.701	8375	902	952
M7	0.263	0.285	0.298	0.193	0.209	0.214	0.783	0.772	0.746	10566	1035	1063
M8	0.195	0.202	0.217	0.284	0.290	0.294	0.759	0.745	0.729	12561	1282	1311
M9	0.403	0.416	0.422	0.211	0.224	0.232	0.776	0.755	0.748	14073	1315	1332
M10	0.445	0.456	0.471	0.230	0.245	0.252	0.729	0.724	0.704	20423	1550	1586
L1	-	0.315	0.318	-	0.158	0.173	-	0.695	0.684	-	1629	1629
L2	-	0.341	0.353	-	0.219	0.222	-	0.647	0.638	-	1811	1876
L3	-	0.294	0.308	-	0.161	0.186	-	0.593	0.543	-	2102	2148
L4	-	0.317	0.338	-	0.162	0.179	-	0.626	0.615	-	2293	2317
L5	-	0.302	0.316	-	0.278	0.301	-	0.673	0.634	-	2405	2439
L6	-	0.313	0.353	-	0.290	0.311	-	0.645	0.630	-	2732	2809
L7	-	0.297	0.309	-	0.265	0.275	-	0.684	0.655	-	3104	3186
L8	-	0.405	0.427		0.321	0.334	-	0.690	0.662	-	3527	3563
L9	-	0.351	0.363	-	0.307	0.316	-	0.581	0.563	-	3770	3819
L10	-	0.419	0.426	-	0.310	0.336	-	0.628	0.604	-	4208	4316

payoff of single-objective versus multi-objective optimization using the MOVNS algorithm, please see Appendix D. Also, test instances for further research are provided in Appendix C.

6.2. Case study

After validating the model performance using GAMS 42.1.0 with the CPLEX 11.0 solver, we solve the model using MOVNS and NSGA-II in MATLAB on an Intel(R) Core(TM) i7 CPU running at 1.60 GHz with 16 GB RAM. Due to the superiority of MOVNS compared to NSGA-II and the epsilon constraint method in terms of CPU time and performance metrics, we solve the real case study using MOVNS.

6.2.1. Computational setup

In the following, we investigate a case study for the city of Vienna to validate the mathematical model and investigate its impact under real-life conditions. The data collected spans from 2020–02–06 to 2020–03–04. The synthetic caller data, which represents the demand for the MOVNS algorithm, comes from an agent-based epidemic model that was developed and used in the course of the COVID-19 pandemic with the aim of being able to compare different measures and their influence [5]. The model is population dynamic, with each model agent interpreted as a statistical representative of a real person. More details on the simulation model can be found in Appendix A.

The Vienna case involves the establishment of mobile testing services during the pandemic, with Klinik Penzing hospital serving as the central laboratory capable of conducting 70 tests per day, while three depots in different areas accommodate mobile testers with a capacity of 10 tests each. Mobile testers operate within an 8-hour daily time frame, conducting tests at patient locations. Service time per patient, including kit preparation and sample collection, is estimated at 11 minutes based on Callahan et al. [9]. The travel time between locations is calculated using Open Street Map data. Overall, the scenario outlines a mobile testing system with a central laboratory and depots, adhering to specific time and capacity constraints to facilitate widespread testing across Vienna.

Patients are categorized into the following age intervals: 1–18, 18–40, 40–50, 50–60, 60–70, 70–80, and 80+. Using the k-means algorithm, a sufficient number of clusters is estimated as 2. The first group has a weight of 0.703, and the second group has a weight of 0.297.

The hazard ratios for individual patients were calculated using the methodology outlined by Jahn et al. [23], which assessed hazard ratios based on specific disease background as Table 3. A hazard ratio of 1 signifies no risk, with kidney disease patients having a hazard ratio of 2, diabetes patients at 1.95, heart disease at 1.17, respiratory diseases at 1.63, liver disease at 1.75, and cancer at 1.72. If a patient has multiple coexisting conditions, the respective hazard ratio coefficients are multiplied. For instance, a male patient with both cancer and hypertension (patient number 1) has a hazard ratio of 4.7859. Patients with a hazard ratio of 1 are considered “standard”, while those with a hazard ratio above 1 are deemed “risky”. The table specifies whether the patient belongs to cluster 1 or 2. The hazard ratio for standard patients in each period is divided by the period number, while for risky patients, it is multiplied by their period number.

Table 4 presents instances of contact tracing information for 20 patients, including recent contacts, ages, and their disease backgrounds. This information is used to calculate the weight of a priority group and the score of each patient.

Table 3
Hazard ratio for patients.

Criteria	Hazard ratio
Cancer	1.72
Liver disease	1.75
Respiratory diseases	1.63
Heart disease	1.17
Diabetes	1.95
Kidney disease	2

Table 4
Contact tracing information.

Time	Recent Contact	Age	Diabetes	Kidney	Heart	respiratory	Liver	Cancer	Hypertension
2020–02–06T20–35–48	123	73	false	false	false	false	TRUE	false	true
2020–02–06T17–52–55	37	80	false	true	true	false	false	false	false
2020–02–06T10–14–00	41	48	false	false	false	false	false	false	false
2020–02–06T23–42–30	34	53	false	false	false	false	false	false	false
2020–02–06T06–03–19	25	84	false	false	true	false	false	false	true
2020–02–06T15–33–43	24	49	false	false	false	false	false	false	false
2020–02–06T09–51–45	54	21	false	false	false	true	false	false	false
2020–02–06T18–46–07	54	65	false	false	false	false	false	false	true
2020–02–06T19–19–45	172	50	false	false	false	false	false	false	false
2020–02–06T19–09–05	20	79	false	false	false	false	false	false	true
2020–02–06T23–53–47	140	30	false	false	true	false	false	false	true
2020–02–06T06–48–20	277	38	false	false	false	false	false	false	false
2020–02–06T06–04–54	158	20	false	false	false	false	false	false	false
2020–02–06T16–32–46	25	86	false	false	false	true	false	true	false
2020–02–06T23–53–33	176	72	false	false	false	false	false	false	false
2020–02–06T12–41–03	309	30	false	false	false	false	false	false	true
2020–02–06T08–26–36	47	64	false	false	false	true	false	false	false
2020–02–06T08–31–51	15	46	false	false	false	true	false	false	true
2020–02–06T07–06–17	35	51	true	false	false	false	false	false	false
2020–02–06T22–32–33	76	27	false	false	false	false	false	false	false

6.2.2. *Managerial insights*

In this subsection, the outputs of the mathematical model for the real case study in Vienna are analyzed. These analyses include comparisons between scenarios with and without priority to check the effect of equity, as well as between quarantine and non-quarantine scenarios. Additionally, analyses based on location, allocation, and routing are provided. An exemplary output and analysis of a solution is provided in Appendix B.

Fig. 8 illustrates the **comparison of average waiting times,** measured in days, over a span of four weeks, both with and without-priority scenarios. In the without-priority scenario, we consider a coefficient of 1 for the score of each patient. Firstly, both risky and standard patients experience longer waiting times during the fourth week compared to the preceding weeks in the with-priority scenario. Conversely, the waiting times for both groups are notably lower during the second week. **The significant waiting times experienced by both risky and standard patients during the fourth week suggest potential bottlenecks or inefficiencies in the system's capacity to handle patient loads.** This could indicate a need for more flexible scheduling strategies to better accommodate fluctuations in demand and optimize resource allocation based on patient priorities. In a scenario without priority, the waiting time for risky patients increases compared to the scenario with priority, while it decreases for standard patients.

Fig. 9 illustrates a **comparison of the average number of unserved patients** over a four-week period, both with and without priority scenarios. In the with-priority scenario, the average number of unserved risky patients remains consistently lower than that of standard patients across all weeks. Consequently, **the model prioritizes serving risky patients with higher priority than standard patients to ensure equity.** For instance, in the first week, the average number of unserved risky patients are 0.033, 0.047, 0.057, and 0.072, while for standard patients, the figures are 0.277, 0.254, 0.305, and 0.324, respectively. However, in the scenario without priority, the average number of unserved risky patients drastically increases. Additionally, in weeks 1 and 3, the average number of unserved risky patients exceeds that of standard patients.

Fig. 10 illustrates the **average waiting times for Vienna's 23 districts** based on the day of testing. The waiting times vary across districts due to different factors affecting demand and geographic location. Districts 1, 7, and 8 experience higher demand for testing kits compared to other districts, resulting in longer average waiting times. This increased demand is reflected in the data. Districts 22 and 23 exhibit longer waiting times due to their considerable distance from both the depot and laboratory facilities. The logistical challenges posed by these distances contribute to delays in testing availability. Conversely, District 14 experiences the shortest waiting time among all districts, attributed to its lower demand for testing services. Districts 15, 16, 17, and 18 demonstrate a similar trend in waiting times, with average waiting times of 0.231 and 0.174, respectively. **These districts benefit from their strategic locations situated between depots and laboratories, resulting in relatively shorter waiting times compared to others.**

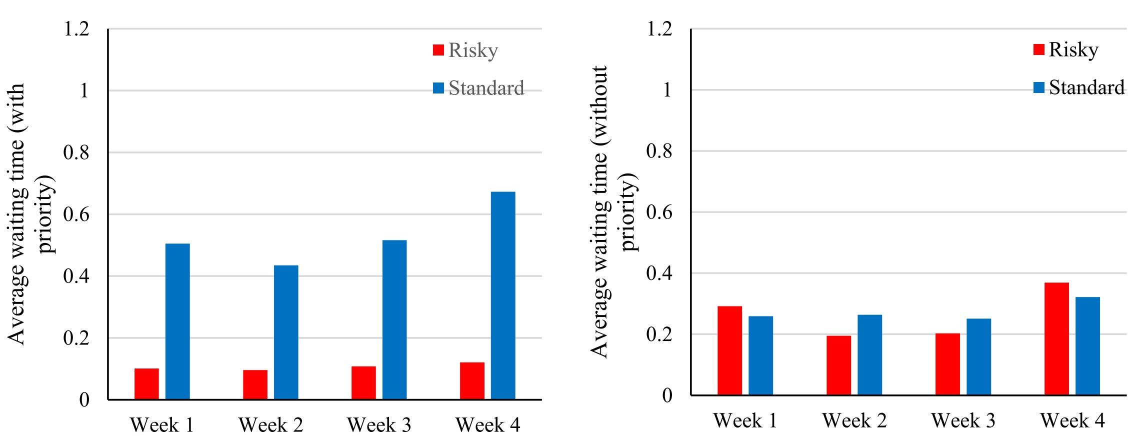

Fig. 8. Comparison of weekly average waiting times with and without priority scenarios.

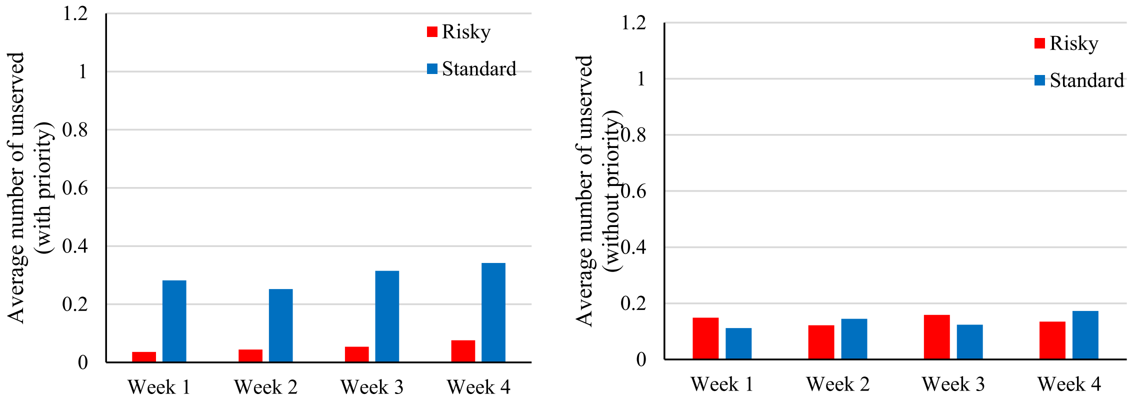

Fig. 9. Comparison of weekly unserved patients with and without priority scenarios.

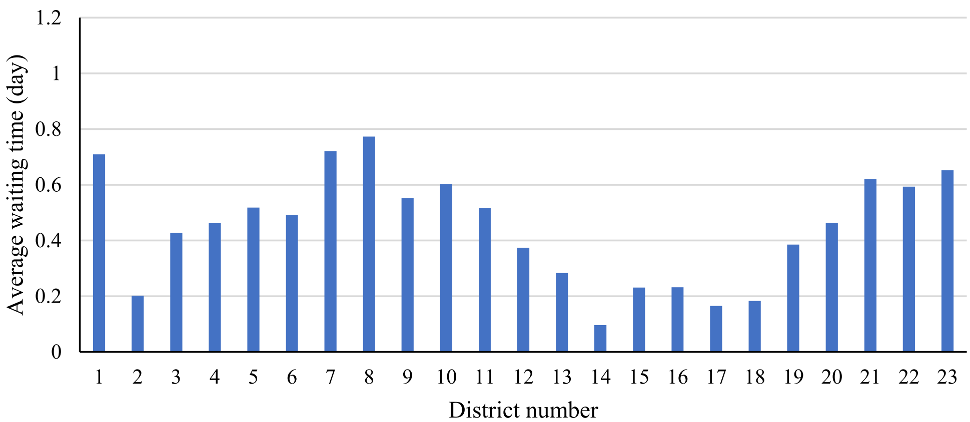

Fig. 10. Average waiting time based on Vienna districts.

6.2.3. Comparison of quarantine and non-quarantine scenarios

The data sets used in this work were generated using two different epidemic scenarios. The first scenario (“quarantine scenario”) is parameterized to the first SARS-CoV-2 wave in Vienna (Austria) during spring 2020 (2020–02–01 to 2020–06–01). Around mid-March, with approximately 250 daily new confirmed cases, the city implemented a strict lockdown. With $\alpha = 2{\cdot}10^{-5}, \beta = 5{\cdot}10^{-8}$, there were between 100 and 250 additional contacts per day from non-infected agents. The second scenario (“non quarantine”) covers the same timeframe but involves no interventions to curb disease spread. Consequently, the wave peaked at around 10,000 daily new confirmed cases in May due to natural immunity.

In Fig. 11, every curve represents one simulation run with the stochastic model, where blue indicates the quarantine scenario and red represents the no-quarantine scenario. The left and right plots display the same curves but use different y-limits. The plotted outcome shows the daily new confirmed COVID-19 cases in Vienna.

Fig. 12 shows the comparison of **average waiting times in quarantine scenarios and without quarantine.** As can be seen, the volume of demand in non-quarantine conditions is much higher than in quarantine conditions, and therefore the average waiting time in quarantine conditions is much lower than in non-quarantine conditions.

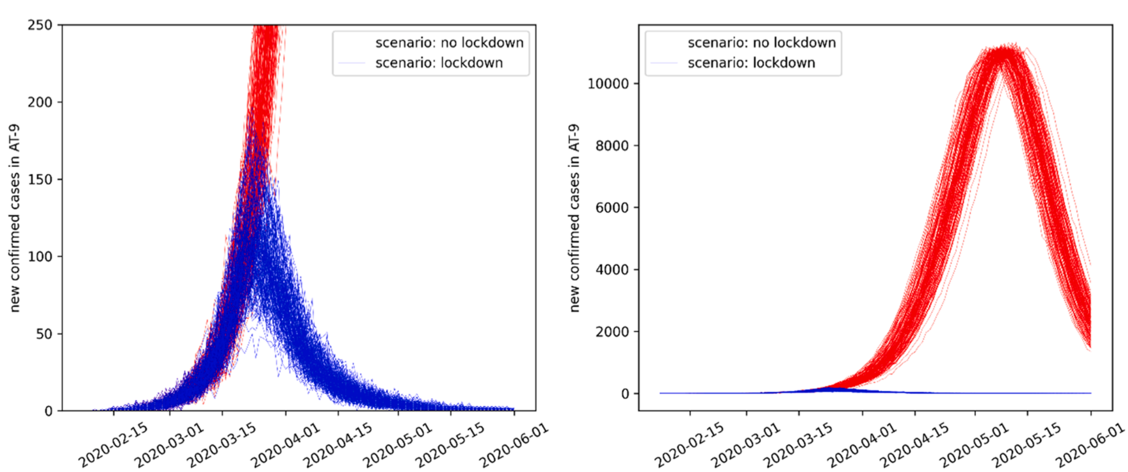

Fig. 11. Simulation run with the "Quarantine" scenario and "non-quarantine" scenario.

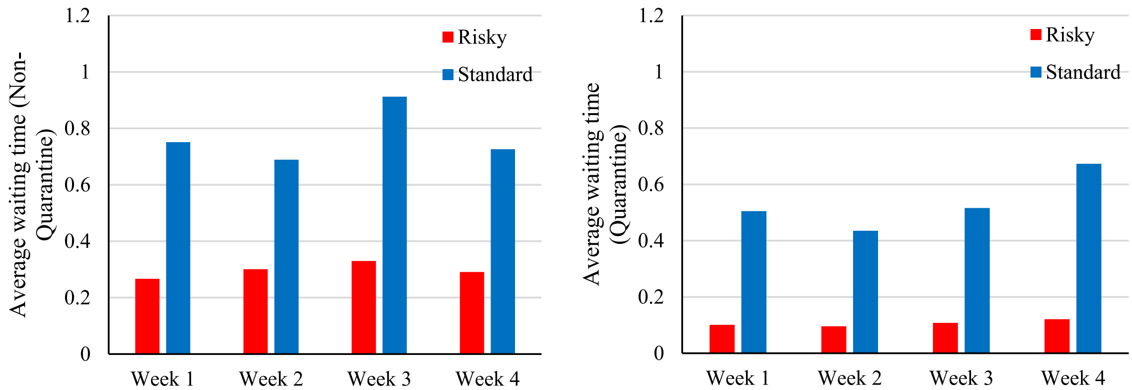

Fig. 12. Comparison of weekly average waiting times with and without quarantine scenarios.

Quarantine significantly reduces waiting times by limiting the influx of demand, as depicted in Fig. 12, showcasing the vital role of quarantine in streamlining processes and minimizing delays. The strong contrast in average waiting times between quarantine and non-quarantine scenarios underscores the **importance of implementing quarantine measures to effectively manage and optimize resource allocation.**

6.3. Sensitivity analysis

In this subsection, sensitivity analysis is performed on important parameters of the model, including the number of mobile testers, laboratory capacity, number of laboratories, and demand. Also, the effects of changing these parameters on the decision variables and objective functions are examined.

Fig. 13 depicts the **sensitivity analysis of the number of mobile testers on average waiting time.** In this analysis, the number of mobile testers varies, decreasing and increasing by 20 % and 40 %. A total of 30 mobile testers are considered as a base case. As evident, **increasing the number of mobile testers results in a reduction of the average waiting time.** For instance, during the first week, the average waiting time for the benchmark mode for risky and standard patients is 0.101 and 0.505, respectively. With a decrease of 20 % and 40 %, these values rise to 0.079, 0.417 and 0.071, 0.199. Likewise, an increase of 20 % and 40 % leads to a reduction of 0.017, 0.128, and 0.045, 0.338. The key observation is that increasing the number of mobile testers consistently leads to a reduction in average waiting time. Conversely, decreasing the number of mobile testers results in an increase in average waiting time. **This suggests that the availability of mobile testers has a substantial impact on the overall efficiency of the testing and delivery system.** The impact on both risky and standard patients implies that a sufficient number of mobile testers is crucial not only for addressing risky patients promptly but also for maintaining efficient service for standard cases.

Fig. 14 illustrates the sensitivity analysis of the **number of mobile testers on the average number of unserved patients.** Consequently, the number of mobile testers decreases and increases by 20 % and 40 % each week compared to the standard mode. As observed, with the increase in the number of mobile testers, the average number of unserved patients decreases, and the slope of the graph is lower for risky patients than for standard patients. **This indicates that the reduction in the number of mobile testers has a greater impact on standard patients.** The analysis demonstrates the importance of optimal resource allocation, particularly in allocating mobile testers.

Fig. 15 depicts the effects of **altering laboratory capacity** on the average waiting time, measured in days over a four-week period. Both increases and decreases in capacity are implemented at rates of 20 % and 40 %. It is important to note that the standard capacity for laboratories is set at 70 people per day. **Changes in laboratory capacity significantly influence the average waiting time for patients.** Given the priority assigned to risky patients, alterations in capacity have a more pronounced impact on standard patients. Consequently, the graph slope is steeper for standard patients compared to risky patients. The significant impact of capacity changes in laboratory operations is evident in the substantial influence on average waiting times for patients. This influence is further exacerbated by the prioritization of risky patients, which results in a more pronounced impact on standard patients when laboratory capacity fluctuates, as indicated by the steeper slope in the graph

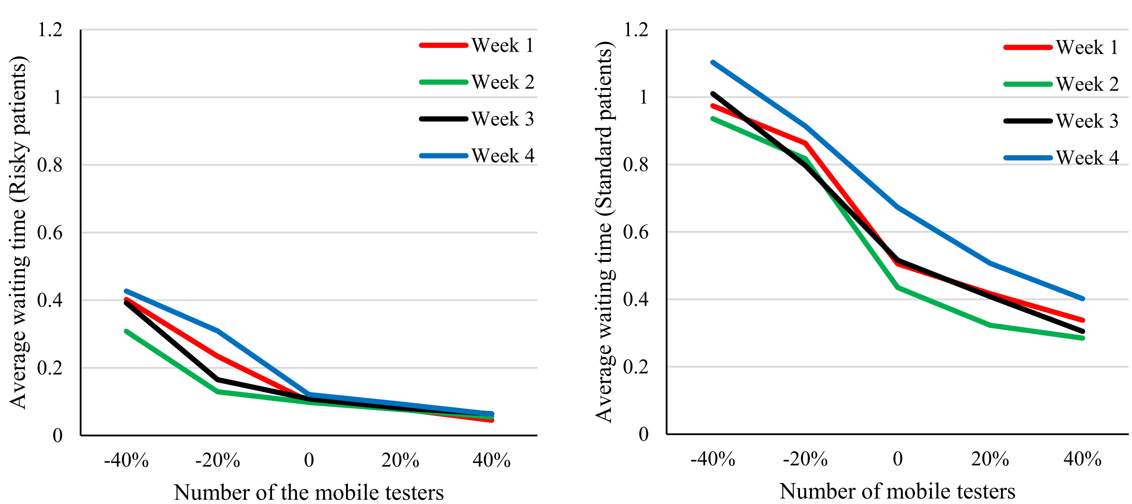

Fig. 13. The impact of mobile tester numbers on the average waiting time.

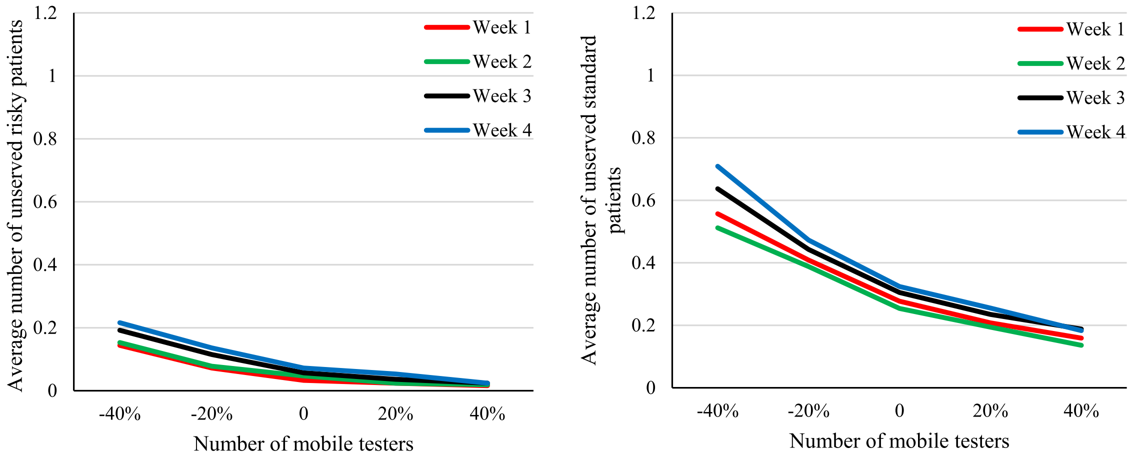

Fig. 14. The impact of mobile testers on the average number of unserved patients.

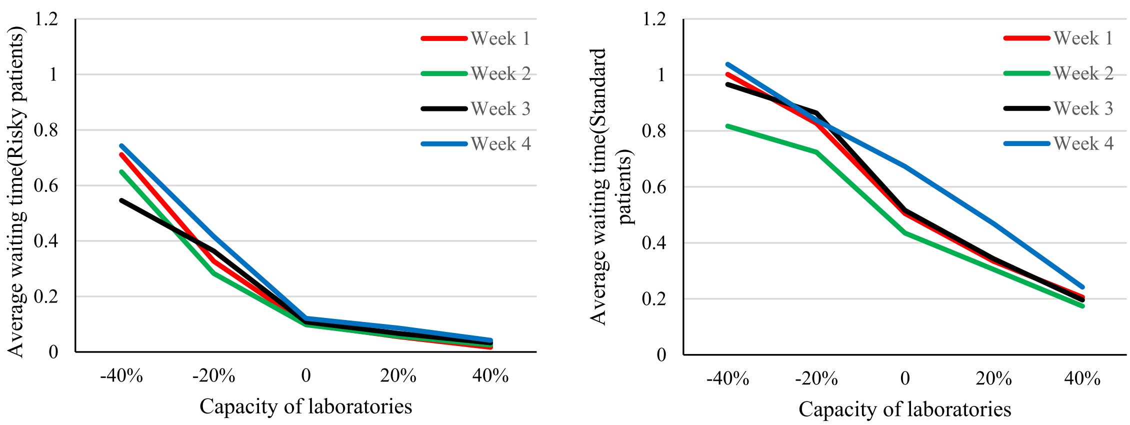

Fig. 15. The impact of laboratory capacity on the average waiting time.

for standard patients compared to risky patients. These findings underscore the challenges posed by resource constraints, with limited laboratory capacity making the system more susceptible to variations and struggling to meet demand, particularly for standard patients. **Consequently, the prioritization of risky patients creates a more challenging situation for standard patients when laboratory capacity is altered,** emphasizing the need for strategic resource allocation, flexibility in capacity planning, and investment in technology and automation to mitigate these challenges and optimize patient care delivery.

Fig. 16 illustrates the effect of changes in **laboratory capacity on the number of unserved patients.** Both increasing and decreasing its capacity significantly influence the number of unserved patients, causing considerable fluctuations. Additionally, the graph indicates that the slope for risky patients is lower than for standard patients. **This implies that standard patients are more affected by changes in**

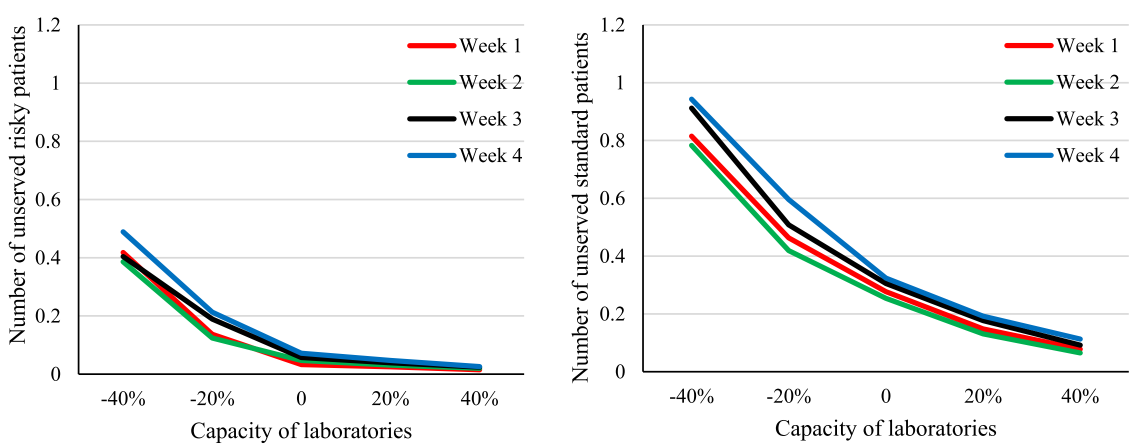

Fig. 16. The impact of laboratory capacity on the number of unserved patients.

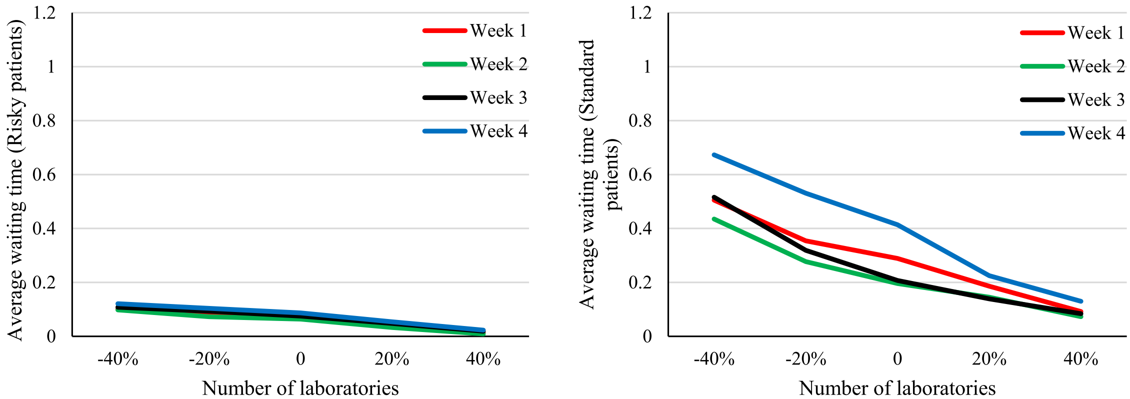

Fig. 17. The impact of number of laboratories on the average waiting time.

capacity compared to risky patients and suggests that the system is more sensitive to changes in laboratory capacity when resources are limited.

Fig. 17 shows the effect of **increasing and decreasing the number of laboratories on the average waiting time.** The number of labs in standard conditions was initially 1, and it was tested here with 5, 10, 20, and 30. As it is known, the increase in the number of laboratories every four weeks reduces the average waiting time for both risk and standard patients. This indicates that investing in expanding infrastructure (in this case, laboratories) can have a direct impact on improving service delivery and patient satisfaction. This is despite the fact that the impact of these changes on standard patients is much higher than on risky patients.

Finally, Fig. 18 shows the effect of **changes in demand on the average waiting time and average number of unserved patients.** As demonstrated, an increase in demand leads to higher average waiting times and average numbers of unserved patients for both risk and standard patients. Additionally, the rate of increase in average waiting time and average number of unserved patients is greater for risk patients compared to standard patients.

Finally, the managerial insights from outputs are as follows: Firstly, prioritizing risky patients over standard patients leads to lower waiting times for the risky patients but can result in longer waiting times for the standard patients, especially during periods of high demand. This highlights the need for equitable and dynamic scheduling strategies to balance priorities effectively. Additionally, quarantine measures significantly reduce waiting times by limiting demand influx, underscoring the importance of proactive measures to manage resource allocation effectively. Sensitivity analysis demonstrates the critical role of effective resource planning and allocation, particularly regarding the number of mobile testers and laboratory capacity. Adjusting these resources based on demand fluctuations can help optimize service delivery and reduce waiting times. Investing in expanding infrastructure, such as laboratories, can directly improve service delivery and patient satisfaction by reducing waiting times, although the impact may vary between risky and standard patients. By incorporating these insights into strategic planning processes, healthcare systems can enhance efficiency, optimize resource allocation, and ultimately improve patient care delivery.

7. Conclusion

In this paper, a bi-objective mathematical model for routing and equitable distribution of testing kits for contagious diseases has been presented. Due to limited resources and the fact that it is not possible to serve all patients, a multi-period TOP modeling has been proposed. The main goal of this research is to maximize the reward obtained from patient visits and minimize the number of unserved patients. Therefore, clustering of geographical areas based on the age of patients and their contact rate as well as prioritization of patients based on their level of risk was performed to ensure equity in testing. Numerical examples have been used to validate the model, which has been solved using the ECP. Indicators such as MID and SM demonstrate the quality of the solutions. The comparison of solution times for the proposed algorithms shows that the MOVNS algorithm outperforms both ECM and NSGA-II, making it the most efficient choice for solving the case study. Therefore, the MOVNS approach has been used to solve the model for a case study involving real data from the COVID-19 pandemic in the City of Vienna, Austria.

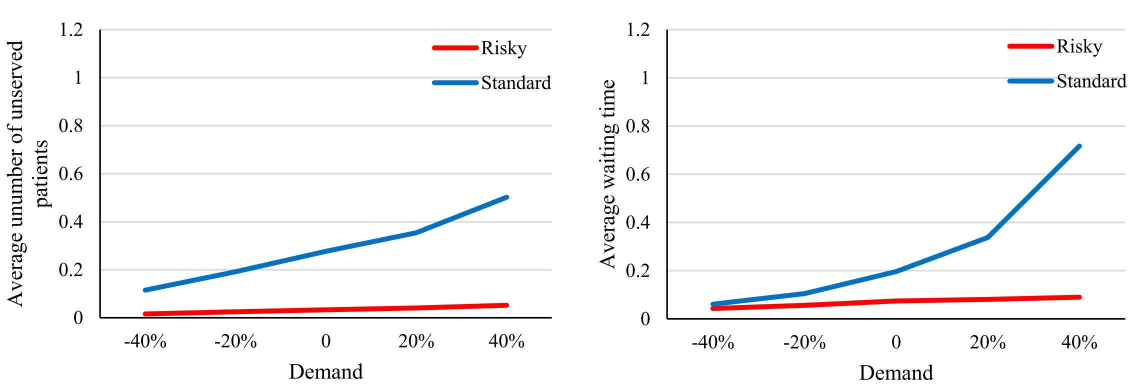

Fig. 18. The impact of demand changes on the average number of unserved patients and average waiting time.

The findings show that in the priority scenario, both risky and standard patients experience longer waiting times during the fourth week. Conversely, waiting times decrease for standard patients but increase for risky patients in the non-priority scenario, indicating a trade-off between prioritization and overall waiting times. Also, in the quarantine scenario, the average waiting time is notably lower compared to the non-quarantine scenario, attributed to reduced demand volume. This highlights the crucial role of quarantine in streamlining processes and optimizing resource allocation to minimize delays. The sensitivity analysis output indicates that increasing the number of mobile testers consistently reduces average waiting time, emphasizing the critical role of tester availability in system efficiency. Furthermore, changes in laboratory capacity significantly impact the number of unserved patients, with standard patients being more affected than risky patients, indicating higher sensitivity to capacity fluctuations in resource-limited scenarios.

In the realm of managing contagious disease testing, several future suggestions can be proposed. Firstly, considering correlation between demand nodes could prove beneficial. By identifying correlated nodes, unnecessary visits could be eliminated. Secondly, acknowledging the inherent uncertainty in such operations, incorporating stochastic service times and other forms of uncertainty through chance constraint-based models could improve the decision-making processes. Finally, considering a cooperative game among the components of the supply chain for distributing diagnostic kits to reduce response time and increase synergy between players could be advantageous.

CRediT authorship contribution statement

Peiman Ghasemi: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Jan Fabian Ehmke:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Martin Bicher:** Writing – review & editing, Validation, Software, Resources, Methodology, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

There are no conflicts of interest to declare regarding this article.

Acknowledgements

This work was partially supported by the FWF Austrian Science Fund (Peiman Ghasemi, Jan Fabian Ehmke, Martin Bicher): I 5908-G.

Appendix A

The simulation model consists of four modules: The *population module* can be seen as the basis of the whole model [6]. It generates the initial agent population using census data and handles the population dynamics (births, deaths, migration). The *contact module* generates contacts between agents based on contact locations (households, workplaces, schools, etc.). These are in turn used by the *disease module* to transmit the virus. Once transmission has taken place, the agent runs through a disease or treatment pathway that ends in recovery and time-limited immunisation. The fourth module is the *policy module*, which is used to simulate measures such as quarantine, contact tracing, school closures, lockdowns or vaccination programmes. One of the unique features of the model is its hybrid time update strategy: while interactions between the agents take place at predefined discrete time steps (daily), status updates from agents that are independent of others are processed on an event-based basis. The latter particularly refers to the disease and treatment pathway, and has already proven to be very valuable for the generation of synthetic data [33,46]. In the present work, the disease pathway, represented in DWH report[2], was divided at the "React on Disease" event. This event marks the point where an affected agent, presumably a person or organism, recognizes symptoms of a disease and decides to seek testing. Following this decision, the agent initiates a "Phonecall" event to contact the relevant authority responsible for conducting the test.

The distribution of the phone call events throughout the day is chosen in consultation with domain experts so that there are two peaks – one in the morning and one in the evening, which corresponds to the off-peak times before and after work. To create additional realism, a certain number of non-infected agents also make a call to the test authority in the model in addition to the actually infected model agents, e.g. because they are overly cautious or are experiencing symptoms due to another infection. The probability that an uninfected agent will generate a Phone call event per day is defined as $\alpha + \beta C$, where C corresponds to the number of active-confirmed cases at that time. With α an epidemic-independent background noise is modelled, with β an increased caution during the disease waves.

Appendix B

Exemplarily, the following figure illustrates the outputs of allocation and routing decisions. The color of each patient node represents the allocation of patients to a specific depot: red nodes are allocated to depot 1, black nodes to depot 2, and blue nodes to depot 3. Yellow nodes signify patients who are not served by the end of the period. Additionally, this figure depicts the two tours of each depot. The unserved patient locations, situated far from the depots, highlight the significance of depot locations in effectively serving patients. Moreover, the allocation trend reveals that the number of nodes allocated to depot 2 (black) exceeds those allocated to depot 1 and 3. The results show that the central regions, such as 1, 8, 7, and 9, bear the greatest load on the depots.

Fig. B1

[2] https://www.dwh.at/projects/covid-19/Covid19_Model-20230322.pdf

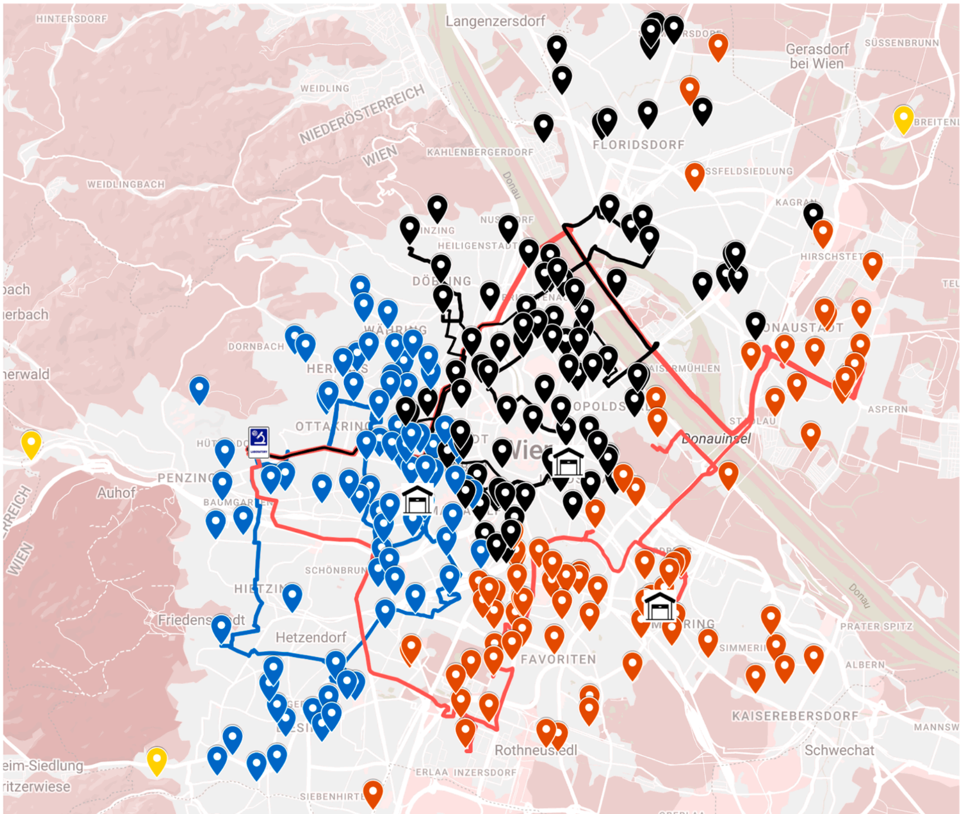

Fig. B1. Allocation and routing output map.

Appendix C

The detailed information related to test instances is presented in Tables C1 and C2. Numerical experiments with problems of different scales (small, medium, and large) are presented in Table C1, while the test instance inputs are defined in Table C2 based on a uniform distribution function.

Table C1
Numerical experiment of problems in different scales.

Classification	Instances	N_c	o	v	l	t
Small	**S1**	3	1	1	1	1
	S2	4	1	1	2	1
	S3	6	1	2	1	1
	S4	7	2	2	2	2
	S5	8	1	2	3	2
	S6	10	2	3	1	2
	S7	12	2	3	3	2
	S8	13	2	3	3	1
	S9	14	2	3	1	2
	S10	16	1	3	2	2
Medium	**M1**	19	3	4	3	3
	M2	22	2	4	2	3
	M3	25	3	4	3	4
	M4	30	3	5	4	4
	M5	33	3	5	3	3
	M6	36	4	6	5	5
	M7	40	4	4	4	4
	M8	42	4	6	7	6
	M9	45	3	5	4	4
	M10	50	4	6	4	5
Large	**L1**	60	3	7	2	4
	L2	65	4	5	3	4
	L3	70	5	4	4	5
	L4	75	5	6	4	5
	L5	80	4	6	5	4
	L6	90	6	7	6	4
	L7	110	7	7	6	5

(continued on next page)

Table C1 (*continued*)

Classification	Instances	N_c	o	v	l	t
	L8	125	7	8	7	5
	L9	130	8	9	8	6
	L10	150	10	13	9	6

Table C2
Test instances inputs.

Scales	Instances	λ_r	C_l	p_{it}	k_o	t_{ij} (Min)	t'_{oi} (Min)	t''_{il} (Min)	s_i (Min)	T_{max} (Min)
Small	**S1**	u ~(0, 1)	u ~(3, 5)	u ~(1, 3)	u ~(1, 3)	u ~(5, 20)	u ~(5, 20)	u ~(5, 20)	u ~(5, 15)	u ~(30, 120)
	S2	u ~(0, 1)	u ~(4, 10)	u ~(1, 4)	u ~(2, 4)	u ~(5, 25)	u ~(5, 25)	u ~(5, 15)	u ~(5, 15)	u ~(30, 180)
	S3	u ~(0, 1)	u ~(5, 10)	u ~(1, 3)	u ~(2, 6)	u ~(5, 20)	u ~(5, 10)	u ~(5, 20)	u ~(5, 15)	u ~(30, 240)
	S4	u ~(0, 1)	u ~(6, 12)	u ~(1, 5)	u ~(3, 8)	u ~(5, 35)	u ~(5, 20)	u ~(5, 25)	u ~(5, 15)	u ~(60, 240)
	S5	u ~(0, 1)	u ~(5, 16)	u ~(1, 4)	u ~(3, 9)	u ~(5, 30)	u ~(5, 25)	u ~(5, 30)	u ~(5, 15)	u ~(120, 240)
	S6	u ~(0, 1)	u ~(5, 10)	u ~(1, 3)	u ~(2, 10)	u ~(5, 35)	u ~(5, 30)	u ~(5, 35)	u ~(5, 15)	u ~(300, 480)
	S7	u ~(0, 1)	u ~(6, 15)	u ~(1, 5)	u ~(4, 8)	u ~(5, 40)	u ~(5, 35)	u ~(5, 40)	u ~(5, 15)	u ~(180, 480)
	S8	u ~(0, 1)	u ~(2, 9)	u ~(1, 2)	u ~(5, 10)	u ~(5, 45)	u ~(5, 40)	u ~(5, 45)	u ~(5, 15)	u ~(120, 540)
	S9	u ~(0, 1)	u ~(4, 9)	u ~(1, 4)	u ~(5, 10)	u ~(5, 50)	u ~(5, 25)	u ~(5, 50)	u ~(5, 15)	u ~(120, 480)
	S10	u ~(0, 1)	u ~(5, 16)	u ~(1, 5)	u ~(6, 12)	u ~(5, 60)	u ~(5, 50)	u ~(5, 60)	u ~(5, 15)	u ~(180, 300)
Medium	**M1**	u ~(0, 1)	u ~(5, 15)	u ~(1, 5)	u ~(5, 8)	u ~(5, 10)	u ~(5, 55)	u ~(5, 10)	u ~(5, 15)	u ~(120, 480)
	M2	u ~(0, 1)	u ~(10, 15)	u ~(1, 4)	u ~(3, 6)	u ~(5, 15)	u ~(5, 60)	u ~(5, 15)	u ~(5, 15)	u ~(120, 540)
	M3	u ~(0, 1)	u ~(5, 15)	u ~(1, 6)	u ~(3, 7)	u ~(5, 20)	u ~(5, 65)	u ~(5, 20)	u ~(5, 15)	u ~(60, 600)
	M4	u ~(0, 1)	u ~(10, 20)	u ~(1, 4)	u ~(3, 10)	u ~(5, 25)	u ~(5, 70)	u ~(5, 25)	u ~(5, 15)	u ~(120, 540)
	M5	u ~(0, 1)	u ~(5, 20)	u ~(1, 6)	u ~(3, 10)	u ~(5, 30)	u ~(5, 50)	u ~(5, 30)	u ~(5, 15)	u ~(180, 600)
	M6	u ~(0, 1)	u ~(15, 20)	u ~(1, 5)	u ~(3, 9)	u ~(5, 35)	u ~(5, 60)	u ~(5, 35)	u ~(5, 15)	u ~(120, 480)
	M7	u ~(0, 1)	u ~(10, 25)	u ~(1, 7)	u ~(2, 12)	u ~(5, 40)	u ~(5, 70)	u ~(5, 40)	u ~(5, 15)	u ~(180, 540)
	M8	u ~(0, 1)	u ~(10, 20)	u ~(1, 6)	u ~(6, 13)	u ~(5, 45)	u ~(5, 40)	u ~(5, 45)	u ~(5, 15)	u ~(300, 600)
	M9	u ~(0, 1)	u ~(10, 30)	u ~(1, 7)	u ~(5, 10)	u ~(5, 50)	u ~(5, 50)	u ~(5, 50)	u ~(5, 15)	u ~(180, 540)
	M10	u ~(0, 1)	u ~(6, 25)	u ~(1, 8)	u ~(6, 11)	u ~(5, 60)	u ~(5, 60)	u ~(5, 60)	u ~(5, 15)	u ~(120, 600)
Large	**L1**	u ~(0, 1)	u ~(10, 30)	u ~(1, 9)	u ~(3, 9)	u ~(5, 10)	u ~(5, 60)	u ~(5, 10)	u ~(5, 15)	u ~(240, 600)
	L2	u ~(0, 1)	u ~(30, 40)	u ~(1, 5)	u ~(4, 16)	u ~(5, 15)	u ~(5, 65)	u ~(5, 15)	u ~(5, 15)	u ~(420, 540)
	L3	u ~(0, 1)	u ~(10, 25)	u ~(1, 7)	u ~(4, 15)	u ~(5, 20)	u ~(5, 70)	u ~(5, 20)	u ~(5, 15)	u ~(180, 600)
	L4	u ~(0, 1)	u ~(10, 30)	u ~(1, 6)	u ~(5, 14)	u ~(5, 25)	u ~(5, 50)	u ~(5, 25)	u ~(5, 15)	u ~(120, 540)
	L5	u ~(0, 1)	u ~(30, 40)	u ~(1, 8)	u ~(6, 15)	u ~(5, 30)	u ~(5, 40)	u ~(5, 30)	u ~(5, 15)	u ~(180, 600)
	L6	u ~(0, 1)	u ~(20, 25)	u ~(1, 7)	u ~(5, 10)	u ~(5, 35)	u ~(5, 30)	u ~(5, 35)	u ~(5, 15)	u ~(120, 480)
	L7	u ~(0, 1)	u ~(20, 30)	u ~(1, 6)	u ~(5, 10)	u ~(5, 40)	u ~(5, 20)	u ~(5, 40)	u ~(5, 15)	u ~(240, 540)
	L8	u ~(0, 1)	u ~(30, 35)	u ~(1, 7)	u ~(8, 10)	u ~(5, 45)	u ~(5, 45)	u ~(5, 45)	u ~(5, 15)	u ~(180, 600)
	L9	u ~(0, 1)	u ~(20, 35)	u ~(1, 9)	u ~(8, 12)	u ~(5, 50)	u ~(5, 50)	u ~(5, 50)	u ~(5, 15)	u ~(120, 540)
	L10	u ~(0, 1)	u ~(30, 40)	u ~(1, 7)	u ~(9, 20)	u ~(5, 60)	u ~(5, 60)	u ~(5, 60)	u ~(5, 15)	u ~(240, 600)

Also, for the test instances, we consider 9 laboratories and 10 depots. Their coordinates (longitude and latitude) are as follows:
Laboratories:

(48.09560, 16.25698), (48.17613, 16.28058), (48.19928, 16.26544), (48.24822, 16.34915), (48.22083, 16.46458), (48.22302, 16.36429), (48.18018, 16.34967), (48.18270, 16.40906), (48.19506, 16.34795)

Depots:

(48.17722, 16.41415), (48.21423, 16.37445), (48.20315, 16.34597), (48.19988, 16.37220), (48.18865, 16.32995), (48.24959, 16.41844), (48.25584, 16.43712), (48.21587, 16.33444), (48.19247, 16.34709), (48.21253, 16.35301)

Appendix D

Table D1 demonstrates the payoff of single-objective versus multi-objective optimization using the MOVNS algorithm. The model was solved for 30 cases with different scales to compare single-objective optimization with multi-objective optimization. For single-objective optimization, the first or second objective function was removed (Best OBJ1 column/Best OBJ2 columns), respectively. In the ‘Best Solution of Multi-OBJ’ column, the best solution for each Pareto front in each case was selected. For the first objective function, the maximum number of Pareto solutions was selected, and for the second objective function, the minimum number of Pareto solutions was selected. As can be seen, the comparison between the single-objective model and the multi-objective model demonstrates that the Pareto solutions are between the global optimum of the single-objective function; however, in some cases, they are equal, for example, in instances 1 to 3 and 5.

Table D1
Payoff of single objective versus multi objective optimization using MOVNS algorithm.

Instances	Single OBJ		Best solution of Multi OBJ
	Best OBJ1	Best OBJ2	
S1	2.13	1	(1.73,1)
S2	2.54	1	(1.54,1)
S3	3.92	2	(2.65,2)
S4	5.35	2	(5.15,3)
S5	5.18	3	(4.94,3)
S6	6.07	4	(5,83,5)
S7	12.34	6	(10.79,8)
S8	9.25	4	(8.25,7)
S9	17.74	6	(17.06,9)
S10	25.09	5	(22.25,8)
M1	22.10	8	(18.64,10)
M2	38.42	11	(34.56,15)
M3	48.83	16	(43.01,19)
M4	41.26	21	(40.35,26)
M5	46.77	18	(41.94,22)
M6	63.32	20	(57.16,24)
M7	86.07	23	(80.90,29)
M8	55.68	21	(51.64,23)
M9	92.90	26	(84.25,32)
M10	118.04	30	(112.78,33)
L1	82.67	36	(74.55,41)
L2	102.43	29	(95.17,32)
L3	107.85	34	(98.96,39)
L4	98.11	38	(91.85,44)
L5	130.07	44	(119.89,46)
L6	153.56	50	(138.05,55)
L7	164.99	51	(158.74,53)
L8	238.21	58	(223.83,65)
L9	188.06	69	(176.96,74)
L10	274.79	63	(262.30,69)

Fig. D1 represents the Pareto frontier of the problem. We present 20 random Pareto points (10 small cases and 10 medium cases) to illustrate the non-dominance of the frontier and the conflict between objective functions. As can be seen, as the first objective function improves, the second objective function deteriorates. Also, except for the first three Pareto solutions where the MOVNS solution and the epsilon constraint solution overlap, the epsilon constraint solution was better than the MOVNS solution in all other cases. Moreover, MOVNS performs better than NSGA-II in terms of Pareto solutions, as it is closer to the epsilon constraint solution.

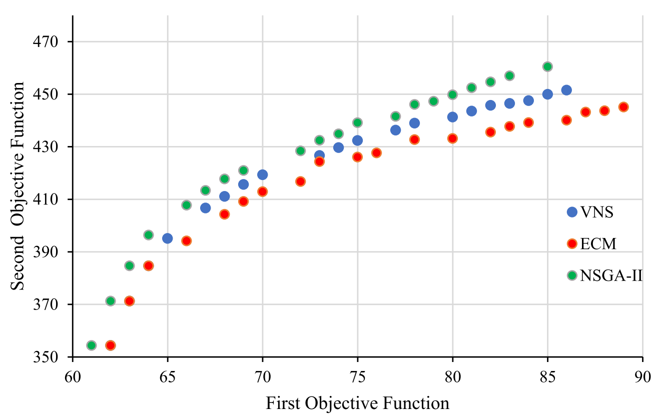

Fig. D1. The Pareto solutions.

Appendix E

Linearization process of Constraint (14):

$$z_{lvt} \leq M \times \sum_{i \in N_c} x''_{ilvt} \qquad l \in L, v \in V, t \in T \tag{e1}$$

$$z_{lvt} \leq \sum_{i \in N_c} y_{ivt} \qquad l \in L, v \in V, t \in T \tag{e2}$$

$$-z_{lvt} + \sum_{i \in N_c} y_{ivt} + M \times \sum_{i \in N_c} x''_{ilvt} \leq M \qquad l \in L, v \in V, t \in T \tag{e3}$$

Data availability

Data will be made available on request.

References

[1] Arbabian ME, Rikhtehgar Berenji H. Inventory systems with uncertain supplier capacity: an application to covid-19 testing. Oper Manage Res 2023;16(1):324–44.
[2] Archetti C, Hertz A, Speranza MG. Metaheuristics for the team orienteering problem. J Heurist 2007;13:49–76.
[3] Aringhieri R, Bigharaz S, Druetto A, Duma D, Grosso A, Guastalla A. The daily swab test collection problem. Ann Oper Res 2024;335(3):1449–70.
[4] Bektaş T, Gouveia L. Requiem for the Miller–Tucker–Zemlin subtour elimination constraints? Eur J Oper Res 2014;236(3):820–32.
[5] Bicher M, Rippinger C, Urach C, Brunmeir D, Siebert U, Popper N. Evaluation of contact-tracing policies against the spread of Sars-Cov-2 in Austria: an agent-based simulation. Med Decis Mak 2021;41(8):1017–32.
[6] Bicher M, Urach C, Popper N. Gepoc ABM: a generic agent-based population model for Austria. In: 2018 Winter Simulation Conference (WSC). IEEE; 2018. p. 2656–67.
[7] Bish DR, Bish EK, El Hajj H. Disease bundling or specimen bundling? Cost-and capacity-efficient strategies for multidisease testing with genetic assays. Manufac Serv Oper Manage 2024;26(1):95–116.
[8] Brimberg J, Salhi S, Todosijević R, Urošević D. Variable neighborhood search: the power of change and simplicity. Comput Oper Res 2023;155:106221.
[9] Callahan C, Lee RA, Lee GR, Zulauf K, Kirby JE, Arnaout R. Nasal swab performance by collection timing, procedure, and method of transport for patients with SARS-CoV-2. J Clin Microbiol 2021;59(9):10–1128.
[10] Chao I, Golden B, Wasil E. The team orienteering problem. Eur J Oper Res 1996;88 (3):464–74.
[11] Chen W, Kumcu GÇ, Melamed B, Baveja A. Managing resource allocation for the recruitment stocking problem. Omega (Westport) 2023;120:102912.
[12] Colajanni G, Daniele P, Sciacca D. Reagents and swab tests during the COVID-19 Pandemic: an optimized supply chain management with UAVs. Oper Res Perspec 2022;9:100257.
[13] Collette Y, Siarry P. Three new metrics to measure the convergence of metaheuristics towards the Pareto frontier and the aesthetic of a set of solutions in biobjective optimization. Comput Oper Res 2005;32(4):773–92.
[14] Deb K, Pratap A, Agarwal S, Meyarivan TAMT. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97.
[15] Duarte A, Pantrigo JJ, Pardo EG, Mladenovic N. Multi-objective variable neighborhood search: an application to combinatorial optimization problems. J Glob Optimiz 2015;63:515–36.
[16] Ehrgott M, Ruzika S. Improved ε-constraint method for multiobjective programming. J Optim Theory Appl 2008;138(3):375–96.
[17] Foy BH, Wahl B, Mehta K, Shet A, Menon GI, Britto C. Comparing COVID-19 vaccine allocation strategies in India: a mathematical modelling study. Int J Infec Dis 2021;103:431–8.
[18] Hanafi S, Mansini R, Zanotti R. The multi-visit team orienteering problem with precedence constraints. Eur J Oper Res 2020;282(2):515–29.
[19] Hansen P, Mladenović N, Todosijević R, Hanafi S. Variable neighborhood search: basics and variants. EURO J Comput Optimiz 2017;5(3):423–54.
[20] Geiger, M.J. (2008). Randomised variable neighbourhood search for multi objective optimisation. arXiv preprint arXiv:0809.0271.
[21] Ghasemi P, Khalili-Damghani K, Hafezalkotob A, Raissi S. Uncertain multi-objective multi-commodity multi-period multi-vehicle location-allocation model for earthquake evacuation planning. Appl Math Comput 2019;350:105–32.
[22] Hosseini-Motlagh SM, Samani MRG, Farokhnejad P. Designing a testing kit supply network for suspected COVID-19 cases under mixed uncertainty approach. Appl Soft Comput 2021;111:107696.
[23] Jahn B, Sroczynski G, Bicher M, Rippinger C, Mühlberger N, Santamaria J, Schmid D. Targeted COVID-19 vaccination (TAV-COVID) considering limited vaccination capacities-an agent-based modeling evaluation. Vaccines (Basel) 2021; 9(5):434. 2021.
[24] Jin H, Thomas BW. Team orienteering with uncertain rewards and service times with an application to phlebotomist intrahospital routing. Networks 2019;73(4): 453–65.
[25] Liang YC, Chuang CY. Variable neighborhood search for multi-objective resource allocation problems. Robot Comput Integr Manuf 2013;29(3):73–8.
[26] Lin S. Computer solutions of the traveling salesman problem. Bell Syst Techn J 1965;44(10):2245–69.
[27] Lin SW, Vincent FY. A simulated annealing heuristic for the team orienteering problem with time windows. Eur J Oper Res 2012;217(1):94–107.
[28] Luo Z, Cheang B, Lim A, Zhu W. An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem. Eur J Oper Res 2013;229(3):673–82.
[29] Mladenović N, Hansen P. Variable neighborhood search. Comput Oper Res 1997; 24(11):1097–100.
[30] Navaei A, Taleizadeh AA, Goodarzian F. Designing a new sustainable test kit supply chain network utilizing Internet of Things. Eng Appl Artif Intell 2023;124:106585.
[31] Ozdemir I, Dursunoglu CF, Kara BY, Dora M. Logistics of temporary testing centers for coronavirus disease. Transp Res Part C: Emerg Technol 2022:103954.
[32] Popa A, Genger JW, Nicholson MD, Penz T, Schmid D, Aberle SW, Bergthaler A. Genomic epidemiology of superspreading events in Austria reveals mutational dynamics and transmission properties of SARS-CoV-2. Sci Transl Med 2020;12 (573):eabe2555.
[33] Popper, N., Zechmeister, M., Brunmeir, D., Rippinger, C., Weibrecht, N., Urach, C., ... & Rauber, A. (2020). Synthetic reproduction and augmentation of COVID-19 case reporting data by agent-based simulation. *medRxiv*, 2020-11.
[34] Rao IJ, Brandeau ML. Optimal allocation of limited vaccine to control an infectious disease: simple analytical conditions. Math Biosci 2021;337:108621.
[35] Roozbeh I, Hearne JW, Pahlevani D. A solution approach to the orienteering problem with time windows and synchronisation constraints. Heliyon 2020;6(6).
[36] Santini A. Optimising the assignment of swabs and reagent for PCR testing during a viral epidemic. Omega (Westport) 2021;102:102341.
[37] Shahnejat-Bushehri S, Kermani A, Arslan O, Cordeau JF, Jans R. A vehicle routing problem with time windows and workload balancing for COVID-19 testers: a case study. Ifac-Papersonline 2022;55(10):2920–5.
[38] Singgih IK. Mobile laboratory routing problem for Covid-19 testing considering limited capacities of hospitals. In: 2020 3rd international conference on mechanical, electronics, computer, and industrial technology (MECnIT). IEEE; 2020. p. 80–3.
[39] Taillard É, Badeau P, Gendreau M, Guertin F, Potvin JY. A tabu search heuristic for the vehicle routing problem with soft time windows. Transp Sci 1997;31(2): 170–86.
[40] Taguchi, G. (1986). Introduction to quality engineering: designing quality into products and processes.
[41] Thul L, Powell W. Stochastic optimization for vaccine and testing kit allocation for the COVID-19 pandemic. Eur J Oper Res 2023;304(1):325–38.
[42] Tlili T, Masri H, Krichen S. Towards an efficient collection and transport of COVID-19 diagnostic specimens using genetic-based algorithms. Appl Soft Comput 2022; 116:108264.
[43] Urrutia-Zambrana A, Tirado G, Mateos A. Variable neighborhood search to solve the generalized orienteering problem. Int Trans Oper Res 2021;28(1):142–67.
[44] Vahdani B, Mohammadi M, Thevenin S, Gendreau M, Dolgui A, Meyer P. Fair-split distribution of multi-dose vaccines with prioritized age groups and dynamic demand: the case study of COVID-19. Eur J Oper Res 2023;310(3):1249–72.
[45] Verdun CM, Fuchs T, Harar P, Elbrächter D, Fischer DS, Berner J, Krahmer F. Group testing for SARS-CoV-2 allows for up to 10-fold efficiency increase across realistic scenarios and testing strategies. Front Publ Health 2021;9:583377.
[46] Wolfinger D, Gansterer M, Doerner KF, Popper N. A large neighbourhood search metaheuristic for the contagious disease testing problem. Eur J Oper Res 2023;304 (1):169–82.
[47] Wu Q, He M, Hao JK, Lu Y. An effective hybrid evolutionary algorithm for the clustered orienteering problem. Eur J Oper Res 2024;313(2):418–34.
[48] Yücel E, Salman FS, Bozkaya B, Gökalp C. A data-driven optimization framework for routing mobile medical facilities. Ann Oper Res 2020;291:1077–102.
[49] Zhang G, Jia N, Zhu N, Adulyasak Y, Ma S. Robust drone selective routing in humanitarian transportation network assessment. Eur J Oper Res 2023;305(1): 400–28.
[50] Zitzler E. Evolutionary algorithms for multiobjective optimization: Methods and applications, 63. Ithaca: Shaker; 1999.
[51] Zitzler E, Thiele L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In: International conference on parallel problem solving from nature. Springer Berlin Heidelberg; 1998. p. 292–301.
[52] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 1999;3(4): 257–71.