



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

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



Towards an efficient collection and transport of COVID-19 diagnostic specimens using genetic-based algorithms



Takwa Tlili *, Hela Masri, Saoussen Krichen

LARODEC Laboratory, Institut Supérieur de Gestion de Tunis, Université de Tunis, 41 Rue de la liberté, Le Bardo 2000, Tunisia

ARTICLE INFO

Article history:

Received 19 November 2020
Received in revised form 1 November 2021
Accepted 27 November 2021
Available online 9 December 2021

Keywords:

Team orienteering problem
Memetic algorithm
Hybrid genetic algorithm
COVID-19 specimen transport
Ambulance routing

ABSTRACT

The speed by which the COVID-19 pandemic spread throughout the world makes the emergency services unprepared to answer all the patients' requests. The Tunisian ministry of health established a protocol planning the sample collection from the patients at their location. A triage score is first assigned to each patient according to the symptoms he is showing, and his health conditions. Then, given the limited number of the available ambulances in each area, the location of the patients and the capacity of the nearby hospitals for receiving the testing samples, an ambulance scheduling and routing plan needs to be established so that specimens can be transferred to hospitals in short time. In this paper, we propose to model this problem as a Multi-Origin–Destination Team Orienteering Problem (MODTOP). The objective is to find the optimal one day tour plan for the available ambulances that maximizes the collected scores of visited patients while respecting duration and capacity constraints. To solve this NP-hard problem, two highly effective approaches are proposed which are Hybrid Genetic Algorithm (HGA) and Memetic Algorithm (MA). The HGA combines (i) a k-means construction method for initial population generation and (ii) a one point crossover operator for solution recombination. The MA is an improvement of HGA that integrates an effective local search based on three different neighborhood structures. Computational experiments, supported by a statistical analysis on benchmark data sets, illustrate the efficiency of the proposed approaches. HGA and MA reached the best known solutions in 54.7% and 73.5% of instances, respectively. Likewise, MA reached a relative error of 0.0675% and performed better than four existing approaches. Real-case instances derived from the city of Tunis were also solved and compared with the results of an exact solver Cplex to validate the effectiveness of our algorithm.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

With the emergence of the COVID-19 pandemic, many health-care facilities found themselves overwhelmed by the number of patients. The hospitals were not an exception as they were struggling to deal with the outbreak of the fast-moving pandemic spread. Several logistic management problems were raised due to limited resources and unusual time pressure. We herein cite some recent researches carried to tackle routing problems related to the pandemic. For instance, Pacheco et al. [1] studied the problem of vehicle routing for the urgent delivery of face shields during the COVID-19 pandemic in Spain. The problem was modeled as a pick up and delivery Vehicle Routing Problem (VRP) variant, where different drivers volunteers to pick up face shields from makers, deliver material to face shield makers and deliver face shields to demand points. A heuristic based on a multi-start insertion

algorithm was implemented. Recently, Singgih [2] considered the problem of deployment of mobile laboratories that are equipped with the testing capability to handle the over-demand situation in Indonesia during COVID-19 pandemic. The author presented a heuristic method to define the optimal location of a single mobile laboratory. Zhang et al. [3] studied the problem of transport of high-risk individuals being transferred for medical isolation in epidemic areas in China where the number of available quarantine vehicles is limited. The problem was solved using a water wave optimization metaheuristic. Chen et al. [4] designed a multi-vehicle multi-trip routing problem to model the contactless food distribution for closed gated communities.

In this paper, we consider another major issue encountered during the pandemic: how to manage the logistics associated with the collection of patients' specimens at their places. We should first highlight the fact that at-home testing of suspect COVID-19 cases could ease pressure on hospitals and emergency services and prevent the spread of infection. Hence, the transportation of specimens in a reliable and efficient manner is essential for effective patient care, allowing faster diagnosis and patient treatment.

* Corresponding author.

E-mail addresses: takwa.tlili@isamm.uma.tn (T. Tlili), masri_hela@yahoo.fr (H. Masri), saoussen.krichen@isg.rnu.tn (S. Krichen).

Exploring the Tunisian case, at the early stage of the pandemic, a special hotline was deployed to answer the requests of patients. The first step of the protocol consists in receiving the patient calls notifying the facility that they are seeking care due to COVID-19 symptoms (e.g. fever, cough, fatigue, headache, shortness of breath, loss of smell or taste, sore throat or other). The severity of COVID-19 symptoms can range from very mild to severe. For instance, people who are older have a higher risk of enduring severe conditions from COVID-19, and the risk increases with age. People who have existing chronic medical conditions also may have a higher risk of serious illness. Another important information, is about the social circle of people the patient met lately, if they had a close contact with someone who has COVID-19 or if they have traveled to/from hot-spots areas. According to the different information, a first triage of the demands is done to assign scores to the different requests. Given the limited size of the ambulances fleet available to carry the specimens collection task, the logistic management was a challenging problem.

In this paper, we propose a solution approach to find an efficient routing plan for the set of ambulances collecting the patients' specimens at their locations. The objective is to respond to the maximum number of requests by maximizing the collected patients' scores. Each ambulance route starts from a given depot in the target testing area, and finishes its one day tour at a hospital having a COVID-19 laboratory. As the testing capacities for each hospital are limited by the number of testing kit supplies and working hours of the health officers, the number of collected specimens at one hospital should not exceed this capacity. The problem is modeled as an integer linear problem, derived from team orienteering vehicle routing problem (TOP) [5,6].

The TOP was first proposed by Chao et al. in 1996 [5]. It can be represented by a directed complete graph where start and end points are specified along with other customers. Each customer has an associated score. The goal is to determine a fixed number of routes, limited in length, that visit some locations and maximize the sum of the collected scores. Furthermore, each customer can be served at most once. The TOP was used to model different routing problems in rescue and emergency situations [7–10]. Given the computational challenge of TOP (NP-hard), heuristic and metaheuristic algorithms are very suitable for finding near-optimal solutions for large sized instances that cannot be solved exactly in an acceptable computation time [5].

Our problem differs from the well-known TOP through the assumptions that the final depot has a determined capacity and also by the fact that the ambulances are located in different starting points. The considered problem can be also defined as an open VRP [11] given that the itinerary of an ambulance is not a closed circuit. We also assume that the generated route ends at one of the hospitals having capacity restrictions. An illustrative example of ambulance routing in the city of Tunis is presented in Fig. 1.

As a solution approach, we propose two variants of genetic algorithms.

- A Hybrid Genetic Algorithm (HGA) that combines k-means method with regular evolutionary operators. The initial population is generated based on a cluster-first route-second approach which starts by grouping the patients into a set of clusters based on their locations. The number of clusters is equal to the number of available ambulances. Then, in order to get the routes, a scheduling of the patients is performed for each cluster without violating the route duration and hospital capacity constraints. A one point crossover operator is used for solution recombination, and a inversion and swap operators are used for the mutation task.



Fig. 1. Illustrative example of ambulance routing problem in Tunis city.

- A Memetic Algorithm (MA) that integrates a Local Search (LS) procedure into the HGA to strengthen the exploitation process and enhance its performance. The main concept of LS technique is to improve interactively the solution using local modifications. The proposed version of MA is based on three effective neighborhood structures which are designed to expand the search space and accelerate the convergence of HGA. In summary, the key point is that both genetic operators and LS neighborhood structures are carefully selected in order to run jointly when producing the solution.

In the literature, GAs have for long been recognized as powerful optimization tools for complex routing problems [12]. Also, numerous local search heuristics [13] showed very promising results on solving different TOP extensions [14,15]. MA offers a framework to combine the exploration power of GA and the exploitation effectiveness of local search. Such combination out-comes a robust metaheuristic that has demonstrated a significant success to handle several NP-hard problems [16]. Motivated by these facts, we propose to implement an HGA and an MA to tackle the TOP.

In order to validate our proposed algorithms, we experiment it using a TSP-based benchmark proposed by Fischetti et al. [17]. Computational experiments include a comparison of our algorithms with five existing methods from the literature as well as a real-word case study. HGA and MA reached the best known solutions in 54.7% and 73.5% of instances, respectively. In terms of the number of average relative percentage deviation, MA produced challenging results by achieving 0.0675% and performed better than four state-of-the-art approaches.

To make more rigorous comparisons, statistical tests have been conducted and proved the competitiveness of our approaches versus the state-of-the-art methods.

The experiments on a real case data set show that the MA improves consistently the exploration of the search space as it produces high quality solutions compared to the HGA. We should note that this enhanced algorithm performance in solving the MODTOP is to the detriment of the required running time.

The remainder of this paper is structured as follows. Section 2 reviews the related work. Section 3 presents the problem description and the proposed mathematical model. Section 4 details the proposed approach and its implementation. Section 5 describes the computational results and comparisons with state-of-the-art approaches. Finally, conclusions and future research directions are provided in Section 6.

2. Related work

With the fast spread of COVID-19 pandemic, several concerns have been increased regarding the performance of healthcare systems to handle the permanent emergencies. Trying to cope with the pandemic situation, different studies considered the problems relative to healthcare rescue, ambulance routing and emergency states. These problems are based on the timely delivery of services/rescue or needed supplies to different locations. Abadi et al. [18] proposed a model of the hybrid salp swarm algorithm and genetic algorithm to solve nurses' scheduling and designation problem during COVID-19 pandemic. However the most important problem in the healthcare optimization context is the ambulances management, allocation and routing especially in COVID-19 context [19]. Cerna et al. [20] evoked the COVID-19 impact on ambulances' turnaround time and proposed a two-stage machine learning methodology to solve the problem in a given time and hospital. Kumar and Susan [21] proposed a novel fuzzy time series forecasting with the particle swarm optimization to handle the emergency ambulance dispatch problem.

Hence, one key question in emergency logistics operations is how to efficiently assign the work to the available ambulances and how to optimize the followed routes notably when we are talking about the transportation of coronavirus specimens. Zahedi et al. [22] minimized the maximum ambulances response and the total critical response time utilizing Internet of Things (IoT). Different vehicle routing variants were used to tackle these problems. An optimal transportation planning based on supply chain theory was evoked by Mosallanezhad et al. [23] using a multi-objective model. An excellent literature review was presented by Tassone et al. [24] describing the recent developments of the Ambulance Routing Problem (ARP) and Ambulance Location Problem (ALP) which are derived variants of the VRP and Maximum Covering Problem (MCP).

Özdamar et al. [25] studied the emergency logistics planning problem in natural disasters. A new model was developed to generate the plan of optimal mixed pick up and delivery schedules for vehicles within the considered planning time horizon as well as the optimal quantities and types of loads picked up and delivered on these routes. Wohlgemuth et al. [26] considered the same problem in a dynamic context to solve vehicle routing problem with anticipation in disaster relief. Campbell et al. [27] studies the routing of vehicles carrying critical supplies in large disasters, given that the timely arrival to patient is critical, the authors introduced two new objective functions: one that minimizes the maximum arrival time and one that minimizes the average arrival time. Different heuristics based on insertion and local search techniques were used. Tlili et al. [11] focused in a similar problem handling emergency requests during disaster situations, where an ambulatory service is provided to patient transportation. A multi-depot VRP model was proposed and a genetic algorithm is implemented to solve a real case study.

In all the previous cited references the considered objective functions are related to either time or cost of the generated routes with the assumptions that all the requests (i.e. of pick up/service or delivery) are satisfied. In the other side, some routing models applied to emergency and rescue assume that only a subset of requests will be covered and the objective is to maximize the number of served requests or the number of collected rewards of the visited points. These assumptions fit with TOP model.

On its broad context, the TOP proposed by Chao et al. [5] is a routing problem where the goal is to determine the path for each team member, in order to maximize the total collected score by the team given a limited time span. The TOP is NP-hard [5]. Only few researches focused on using exact methods [28,29] to solve the TOP. Such methods become highly time-consuming as

the problem instances increase in size. Therefore, the main body of literature dealing with TOP is dominated by heuristic and metaheuristics methods [7,9,10].

Dang et al. [7] proposed an efficient heuristic approach for TOP based on an interval graph model and an inspired particle swarm optimization. Chen et al. [8] used the TOP to model the problem of optimal team deployment in urban search and rescue in post-disaster circumstances. The problem is formulated as a multistage stochastic program. The solution consists in identifying the teams tours maximizing the total expected number of people that can be saved. The authors proposed a dynamic solution framework to handle the continuous flow of stochastic information. Baffo et al. [9] presented an orienteering-based approach to manage emergency situation, where the purpose is to collect the highest number of people from several origins and bring them into a unique destination, using a limited number of capacitated vehicles and while respecting a time limit. The problem was modeled as multi-origins capacitated TOP and solved by the ant colony metaheuristic. Recently, Saeedvand et al. [10] studied the problem of TOP with time window for disaster rescue using robots. A multi-objective formulation is considered and an efficient solution combining multi-objective evolutionary algorithms with learning algorithms is presented. The authors presented a hybrid adaptive large neighborhood search to solve this problem.

Recently, various local search metaheuristics showed very promising results on solving different variants of the TOP problem. Hammami et al. [13] proposed a hybrid adaptive large neighborhood search to solve the TOP. Their solution approach combines the exploration power of local search procedures and an optimization stage using a set packing problem to improve the solutions. Orlis et al. [15] introduced a new variant called TOP with overlaps, where each node can be serviced via a set of service points. An exact branch-and-cut-and-price and a large neighborhood search was developed to solve the problem. Bayliss et al. [30] developed a learnheuristic solution approach that integrates metaheuristics and machine learning for solving the team orienteering problem with aerial drone. Amarouche et al. [14] studied the TOP with time windows where they proposed a neighborhood search method based on (1) splitting algorithms with the alternation between two different search spaces, a large tour search space and an inner route search, and (2) the use of a long term memory mechanism to keep the elite solutions.

3. Problem description and mathematical model

The Tunisian ministry of health established a protocol to be followed in order to schedule the visits for suspected COVID-19 patients requesting an at-home test. The Fig. 2 describes the adopted process. As a first step, the call center answers the requests of patients suspected to be COVID-19 positive. An electronic information form is filled for each patient to get the following information:

- The symptoms (e.g. fever, cough, fatigue, headache, shortness of breath, loss of smell or taste, sore throat or other)
- The age and health condition of the patient.
- Whether he was in a close contact of someone who is diagnosed with COVID-19
- Whether he traveled lately to a hot-spot area

During the triage phase, a score is assigned automatically to each patient. For those having a score greater than a certain threshold, they will be scheduled in a waiting list, to be visited by an ambulance at their place to take a PCR (Polymerase Chain Reaction) test. Given the limited size of the available ambulances fleet, the logistic management should be done efficiently to answer the maximum number of demands. Our model describes

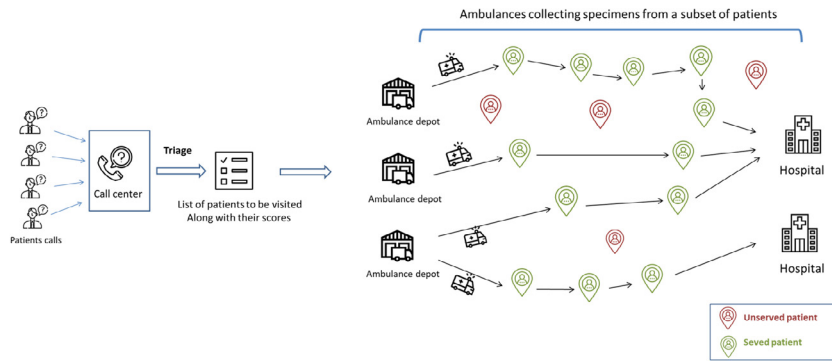


Fig. 2. The protocol followed to answer COVID-19 patients' requests for an at-home test.

the optimal routing plan of ambulances fleet collecting the specimens from the patients. We should point out that by the end of the tours, the collected tests will be delivered to a set of hospitals having limited capacities (e.g. the capacity depends on the number of testing kit supplies and working hours of the health officers). Hence, another decision along the routing is to assign each vehicle to one of the hospitals while respecting the capacity constraints. We propose to model this problem as a Multi-Origin–Destination Team Orienteering Problem (MODTOP) by extending the original TOP formulation [5,6]. The objective is to design a set of ambulances' routes that maximizes the total collected scores, while satisfying the hospitals capacities and the maximum duration constraints for each route.

The problem can be described by a complete directed graph $G = (N, A)$, where $N = \{0, 1, 2, \dots, n\}$ is the node set and $A = \{(i, j) : i, j \in N\}$ is the arc set. The routes of the ambulances start from one of the depots and finish the tour at one of the hospitals. Let O denoted the set of ambulances depots where each depot contains v_o vehicles. Let H denotes the set of hospitals where each hospital has a limited capacity c_h . Furthermore, a non-negative travel time t_{ij} is associated with each arc $(i, j) \in A$. The total time taken to visit the points on each of the paths cannot exceed the specified limit T_{max} . The set of patients is denoted by N_c . Each patient $i \in N_c$ has a predefined profit p_i (i.e. score) and a service time s_i .

Due to time constraint and hospitals' capacity restrictions, it is not possible to serve all patients. The objective is to find the subset of served patients along with the corresponding visiting sequences such that the total collected profit is maximized. We should note that each customer is visited at most once by only one ambulance.

Used parameters as well as the decision variables are described as follows.

Notation:

Parameters:

N	Set of nodes $N = \{O \cup H \cup N_c\}$.
N_c	Set of patients
O	Set of depots
H	Set of hospitals
A	$\{(i, j) : i, j \in N\}$ the arcs set
K	Set of ambulances
c_i	Capacity of hospital i
p_i	Profit or score of patient i
v_o	Number of ambulances in depot o
t_{ij}	Travel time from i to j

s_i	Service time for patient i
T_{max}	Maximum total travel time for an ambulance
<i>Decision variables:</i>	
x_{ijk}	$\begin{cases} 1 & \text{if the arc } (i, j) \text{ is traversed by ambulance } k \\ 0 & \text{otherwise} \end{cases}$
y_{ik}	$\begin{cases} 1 & \text{if patient } i \text{ is served by ambulance } k \\ 0 & \text{otherwise} \end{cases}$

$$\text{Max} \quad \sum_{i \in N_c} \sum_{k \in K} p_i y_{ik} \tag{1}$$

Route construction constraints

$$\sum_{k \in K} \sum_{j \in N_c} x_{ojk} = v_o \quad o \in O \tag{2}$$

$$\sum_{j \in N} x_{ijk} = \sum_{j \in N} x_{jik} \quad i \in N_c, k \in K \tag{3}$$

$$\sum_{h \in H} y_{hk} = 1 \quad k \in K \tag{4}$$

$$\sum_{i \in N} \sum_{k \in K} x_{hik} = 0 \quad h \in H \tag{5}$$

Capacity constraints

$$\sum_{k \in K} y_{hk} \sum_{i \in N_c} y_{ik} \leq c_h \quad h \in H \tag{6}$$

Time constraints

$$\sum_{i \in N} \sum_{j \in N} t_{ij} x_{ijk} + \sum_{i \in N_c} s_i y_{ik} \leq T_{max} \quad k \in K \tag{7}$$

$$\text{Patient visiting constraints} \quad \sum_{k \in K} y_{ik} \leq 1 \quad i \in N_c \tag{8}$$

$$\sum_{j \in N} x_{ijk} = y_{ik} \quad i \in N_c \cup H, k \in K \tag{9}$$

Decision variables type constraints

$$y_{ik}, x_{ijk} \in \{0, 1\} \quad i, j \in N, k \in K \tag{10}$$

The objective function (1) maximizes the total collected scores. The set of constraints (2)–(3)–(4)–(5) ensures that each ambulance starts from one depot and arrives at one of the hospitals as open vehicle tour.

- Constraints (2) ensure that v_o vehicles leave each depot o .
- Constraints (3) are the flow conservation constraints.
- Constraints (4) ensure that each ambulance visits exactly one hospital.
- Constraints (5) ensure that the hospitals are the end-points of the ambulances' routes.

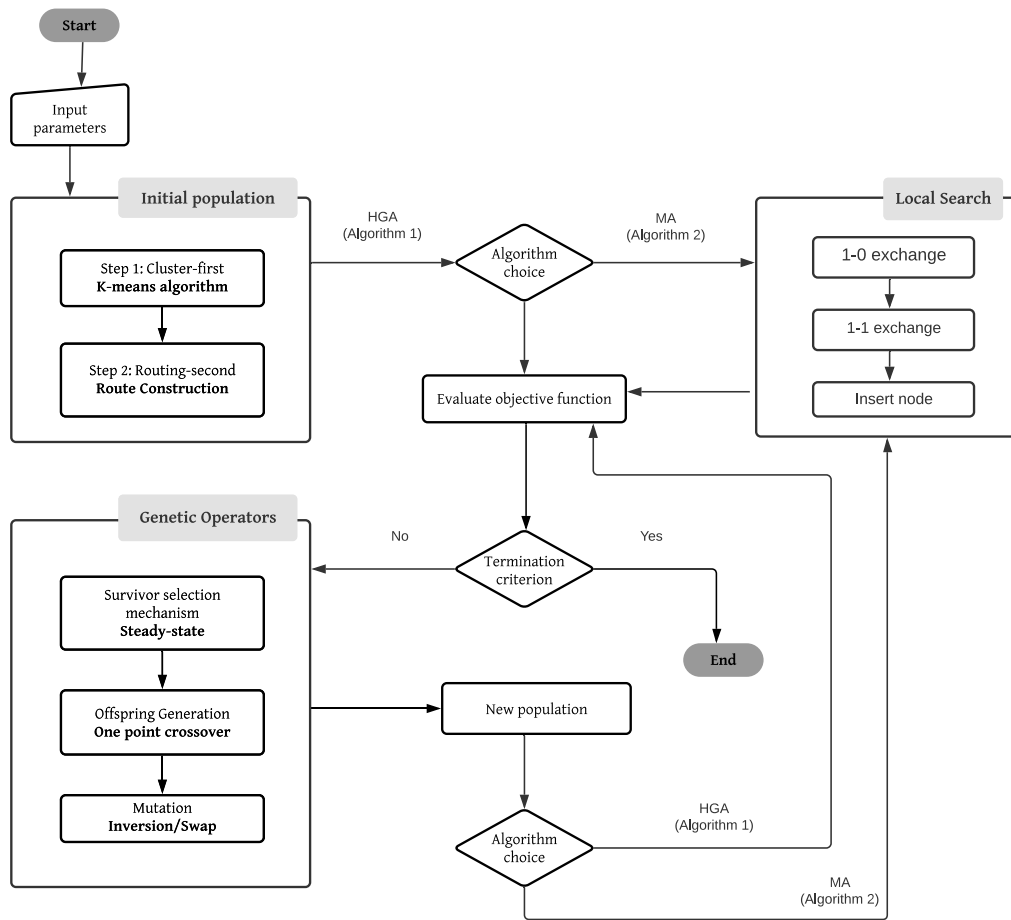


Fig. 3. Solution framework flowchart.

- Constraints (6) state that for each hospital h , the total number of collected specimens is lower than the capacity of the hospital c_h . Constraints (7) checks that the route duration of each ambulance does not exceed the T_{max} . Constraints (8) and (9) are related to patient visiting assumptions.
- Constraints (8) ensure that each patient is visited at most once by one of the ambulances.
- Constraints (9) set values of the decision variable y_{ik} . A patient i is considered to be visited by the ambulance k if this vehicle crossed one of the arcs (i, j) where $j \in N$.
- Constraints (10) impose binary restrictions to x and y decision variables.

4. Solution approaches

Due to its NP-hardness [31], the MODTOP cannot be solved optimally within a reasonable computational time particularly for large-scale instances. Therefore, we designed two genetic based algorithms which are (1) Hybrid Genetic Algorithm (HGA) and (2) Memetic Algorithm (MA). The general structure of the proposed approaches is based on the combination of k-means clustering and genetic operators. The k-means clustering process was proved to be efficient in simplifying logistic networks and improving the routing solution [32,33]. Furthermore, in order to prevent the HGA from being trapped into local optimum, we developed the MA that incorporates a local search (LS) procedure to improve the solution in local search scope.

For both of HGA and MA, the first step is to generate an initial population based on a cluster-first route-second method that integrates K-means algorithm and route construction procedure (Section 4.3). Common GA operators are proposed as follows.

- Survivor selection mechanism adapting “Steady-state” technique (Section 4.4).
- Offspring generation using “Random one point crossover” (Section 4.5).
- Mutation operator applying “Inversion and swap” methods (Section 4.6).

Three different neighborhood structures have been designed in the local search for the proposed MA. The neighborhood operators named, *1-0 exchange*, *1-1 exchange* and *Insert node*, are executed in sequence to improve the solutions. More details for the LS are outlined in Section 4.7. The solution quality evaluation is based on the fitness function that seeks to maximize the total collected profit as explained in Section 4.1. The termination criterion is presented in Section 4.2.

The aforementioned framework is detailed in Fig. 3.

4.1. Solution representation and evaluation function

The chromosome is coded as a vector of K substrings where K is the number of ambulances. Each substring, describing the route followed by a vehicle $k \in K$, is composed of three parts:

- The depot index $o \in O$ of the vehicle k
- The vector containing the index of visited patients where $i \in N_C$.
- The index of the hospital $h \in H$ where the vehicle k will deliver the collected specimens.

In the proposed encoding, $K - 1$ zeros are employed to separate the set of routes. Given VP the number of visited patients and

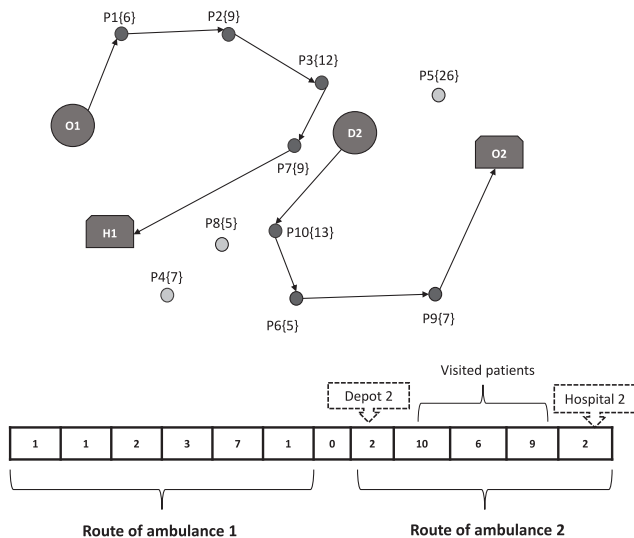


Fig. 4. Example of a solution representation.

UP the number of unvisited patients. The total length of the chromosome is equal to $VP + (K - 1) + (K \times 2)$, where $(K \times 2)$ is the number of depots and hospitals in the chromosome.

The quality of the solution s is evaluated using a fitness function defined as $f(s) = TS(s)$, where $TS(s)$ is the total score of s .

An example of the solution for COVID-19 context is shown in Fig. 4. Given 10 patients to be visited by 2 ambulances based on 2 medical depots. The 10 diagnostic specimens are transported to 2 different hospitals. The visiting sequence of route 1 from $o1$ to $h1$ is $1 - 2 - 3 - 7$, and the visiting sequence of route 2 from $o2$ to $h2$ is $10 - 6 - 9$.

4.2. Termination criterion

In genetic algorithms, three kinds of termination criteria are usually employed, CPU time limit, number of fitness function evaluations, and maximum number of generations. The CPU time depends on the computer specifications which are not detailed enough in many studies. The best fitness value is unpredictable and the convergence is uncertain. Thus, the maximum number of generations is the most adequate stopping criterion for the proposed approach.

4.3. Initial population

The initial population can be viewed as a “cluster-first route-second” heuristic.

- **Cluster-first step:** K-means algorithm is used to assign a given set of sub-patients $P \in N_c$ to clusters [34]. It is about generating a set of N clusters, named C , where $C = n_1, n_2, \dots, N$. Each cluster (n_i) involves a set of patients and the number of clusters is equal to the number of routes (i.e. ambulances). At the end of this step, k-means generates a set of k clusters containing the patients to visit. Each feasible cluster should satisfy the constraints described in the problem formulation of Section 3. K-means method is detailed in algorithm 1.
- **Route-second step:** In the resulting clusters from step 1, the patients in each cluster are sorted randomly. Each obtained route contains a number of nearby patients with random order. To generate an initial population, a set of generated

routes needs to be feasible. Based on the encoding previously described in Section 4.1, each route R is adjusted as follows.

- Add a depot number o as a first element of R .
- Add the patients sequentially in a random order to R .
- Add a random hospital number h at the end of R while checking its capacity constraint.
- If the route duration exceeds T_{max} then remove the patients with the lowest scores until the duration constraint is satisfied.

The aforementioned steps construct a set of feasible routes to be inserted in one solution.

This procedure is iterated multiple times by varying the subset of patients P , given as input to the k-means, until the population size is reached. The initial subset $P \in N_c$ is randomly generated containing from 80% up to 100% of all patients.

Algorithm 1 K-means algorithm

Require: P : A set of nodes (patients)

Ensure: C : A set of K clusters

- 1: Choose randomly p patients from P as the initial centroids
 - 2: $e = 0$;
 - 3: **repeat**
 - 4: **for all** $c \in C$ **do**
 - 5: **for all** $j \in P$ **do**
 - 6: Calculate the Euclidean distance to each centroid;
 - 7: Assign the closest node to centroid to create the cluster ;
 - 8: Update cluster means with the smallest distance;
 - 9: Recompute the new cluster centroid;
 - 10: **end for**
 - 11: **end for**
 - 12: $e + +$;
 - 13: **until** No change in the centroid
-

4.4. Survivor selection mechanism

The survivor selection of the candidate solutions is an important step in genetic algorithms. The most promising chromosomes are included in the next generation and will be used as parents in the crossover operations. There are different selection techniques, e.g roulette-wheel, rank selection, tournament, elitism and steady-state [35].

In the proposed algorithm, a chromosome is more likely to be selected if its fitness function value f is high and the steady-state mechanism is adopted. The main idea of steady-state is that the candidate solutions are allowed in the current population to become a part of the new population. The steady-state selection process used in HGA and MA is detailed as follows.

Step 1: Identify the best solutions of the population.

Step 2: Remove nb bad chromosomes.

Step 3: The rest of the current population survives to the new generation without going through selection process.

4.5. Offspring generation

We apply a random one-point crossover, one random combination point is selected from both parents' chromosomes. The chromosomal section next to the chosen point are swapped with each other, giving birth to two new offsprings. As the generated offspring may contains redundant visit to a same patient by

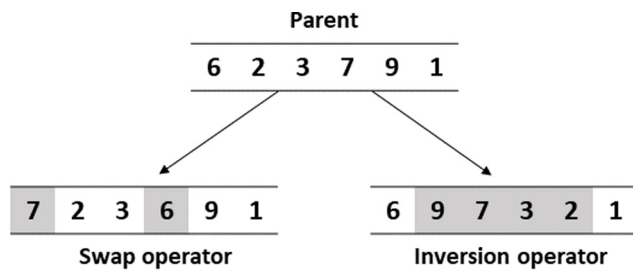


Fig. 5. Mutation operators used by the proposed algorithms.

different ambulances, a repair phase is necessary to make the solution feasible. we detail in what follows the applied algorithm

Algorithm 2 Offspring generation

```

Require: two parent solutions  $p1, p2$ 
Ensure: New offspring  $s1, s2$ 
1:  $s1, s2 \leftarrow \text{randomonepointcrossover}$ ;
2: for all  $s \in \{s1, s2\}$  do
3:    $s \leftarrow \text{remove\_redundant\_patients}(s)$ ;
4:   for all  $x \in \text{Sorted\_list\_of\_unvisited\_patient}(s)$  do
5:      $s \leftarrow \text{best\_insertion}(s, x)$ ;
6:   end for
7: end for
    
```

As an improve to the produced offsprings, a new insertion heuristic is introduced. Insertion heuristics have proven to be popular methods for solving a variety of vehicle routing and scheduling problems [36]. The *best_insertion*(s, x) contains three steps

- Compute the geographical center of all routes (i.e. genes)
- Assign x to the nearest route that has not exceeded the capacity constraint
- If the generated route exceed T_{Max} , mark x as unvisited.

4.6. Mutation

In order to maintain the population diversity, a mutation operator is performed after the crossover previously described. The mutation process applied in the proposed algorithm integrates two different types which are (1) inversion operator and (2) swap sequence operator. The inversion is about switching the order of a couple of nodes randomly. The swap consists in exchanging two nodes that are randomly selected from the path [37]. Fig. 5 presents the difference between the two operators.

4.7. Local search

Local search (LS) is a classical method used for solving combinatorial optimization problems. In extant studies, LS proved to be an effective method for generating high-quality solutions to routing problems [38]. The LS basic idea is to improve interactively an initial solution using local modifications until finding the high-quality solution. The LS approach (1) operates various techniques to specify a move type then (2) constructs a neighborhood of the current solution. One of the key features when implementing a LS procedure is the choice of the neighborhood operators. The neighborhood is the set of solutions $N(s_i)$ that can be reached after modifying some components of an initial solution s_i .

Three neighborhoods are implemented in MA described below.

Neighborhoods

In the proposed LS, three neighborhood structures have been developed to explore the solution space. An illustrative example is illustrated in Fig. 6 to better explain the developed operators. In the illustrative example, the initial routes are:

$$R_1 = \{D \rightarrow x_1 \rightarrow x_2 \rightarrow x_4 \rightarrow x_5 \rightarrow x_6 \rightarrow H_1\} \text{ and } R_2 = \{D \rightarrow x_7 \rightarrow x_8 \rightarrow x_9 \rightarrow x_3 \rightarrow H_2\}.$$

(a) **1-0 exchange:** Relocates a patient from its current position to another by replacing three solution arcs. As explained in case (a), the patient x_4 is relocated after deleting the arcs (x_2, x_4) and (x_4, x_5) from R_1 and (x_8, x_9) from R_2 . Three new arcs are created which are (x_2, x_5) , (x_8, x_4) and (x_4, x_9) . After the 1-0 exchange operator, the new routes are:

$$R_1 = \{D \rightarrow x_1 \rightarrow x_2 \rightarrow x_5 \rightarrow x_6 \rightarrow H_1\} \text{ and } R_2 = \{D \rightarrow x_7 \rightarrow x_8 \rightarrow x_4 \rightarrow x_9 \rightarrow x_3 \rightarrow H_2\}.$$

(b) **1-1 exchange:** Swaps the positions of two patients, x_4 from R_1 and x_9 from R_2 , by removing four arcs: (x_2, x_4) , (x_4, x_5) , (x_8, x_9) , and (x_9, x_3) , then creating four new ones. After the 1-1 exchange operator, the new routes are:

$$R_1 = \{D \rightarrow x_1 \rightarrow x_2 \rightarrow x_9 \rightarrow x_5 \rightarrow x_6 \rightarrow H_1\} \text{ and } R_2 = \{D \rightarrow x_7 \rightarrow x_8 \rightarrow x_4 \rightarrow x_3 \rightarrow H_2\}.$$

(c) **Insert node:** Inserts a new patient in a route where the location consumes the least travel time. A detailed explanation of the insert operator is introduced in [6].

As shown in case (c), the unvisited patient x_{11} is supposed (1) to be the least time consuming compared to patient x_{10} and (2) its insertion does not exceed the maximum travel time. A single arc is deleted (x_6, H_1) and two other arcs are created (x_6, x_{11}) and (x_{11}, H_1) . After the insert node operator, the new routes are:

$$R_1 = \{D \rightarrow x_1 \rightarrow x_2 \rightarrow x_4 \rightarrow x_5 \rightarrow x_6 \rightarrow x_{11} \rightarrow H_1\} \text{ and } R_2 = \{D \rightarrow x_7 \rightarrow x_8 \rightarrow x_9 \rightarrow x_3 \rightarrow H_2\}.$$

The LS process follows the scheme of Algorithm 3. It is about carrying on consecutively the neighborhood structures previously presented.

Algorithm 3 Local search algorithm

```

Require: Initial solution: S
Ensure: Updated S
1:  $S_1 \leftarrow$  First Improvement on S using 1-0 exchange operator;
2:  $S_2 \leftarrow$  First Improvement on  $S_1$  using 1-1 exchange operator;
3:  $S_3 \leftarrow$  First Improvement on  $S_2$  using Insert node operator;
4: if  $f(S_3) > f(S_1)$  then
5:    $S \leftarrow S_3$ ;
6: else
7:   GO to Line 1;
8: end if
    
```

Given an initial solution S , we sequentially improve this solution by choosing the first incumbent neighbor improving S with respect to 1-0 exchange operator and so on until computing a first local optimum S_1 that cannot be improved no more. In the same way, given the solution S_1 as the starting point for the next local search using 1-1 exchange operator until finding a solution S_2 . Finally, we apply the insert node operator to improve S_2 and find the new solution S_3 . This process is repeated until a local optimum of the three structures of neighborhood is reached.

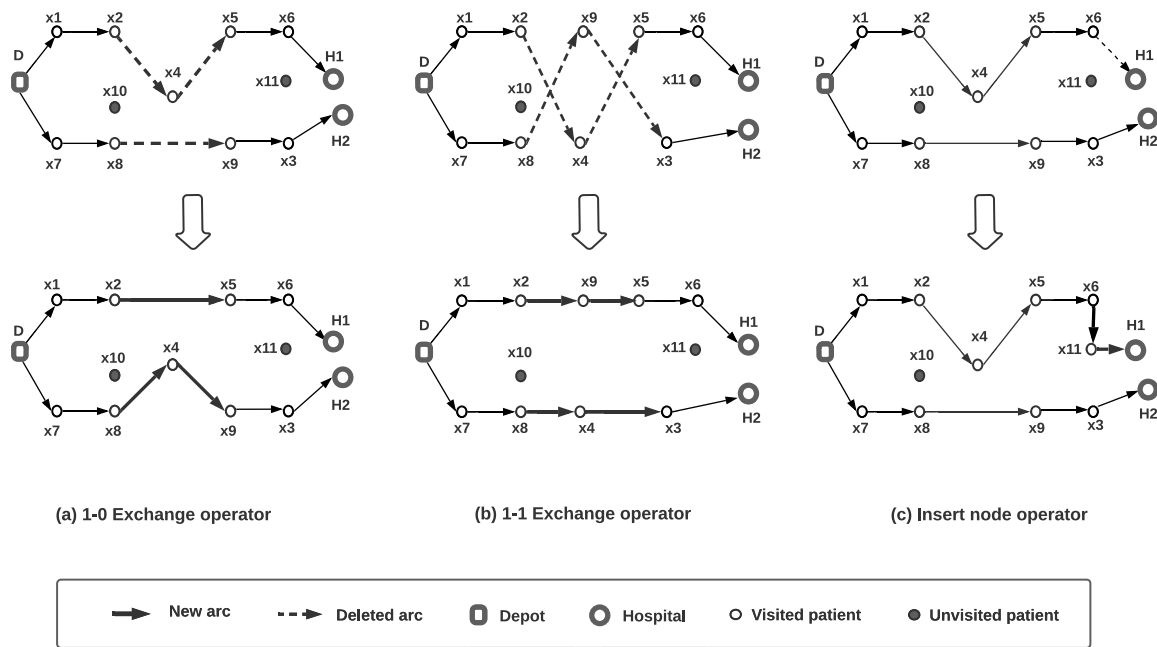


Fig. 6. Local search operators used in the memetic algorithm; Given two routes R_1 and R_2 with nine visited patients $\{x_1, \dots, x_9\}$, two unvisited patients $\{x_{10}$ and $x_{11}\}$, one depot D , and two hospitals H_1 and H_2 .

5. Computational experiments

The proposed algorithms described in the previous section were coded in Java. The testing environment is a computer equipped with an Intel Core i7-7500U processor running at 2.9 GHz and with 8 GB of RAM. In this section, we detail the numerical results carried on to demonstrate the effectiveness of HGA and MA. For each instance, the proposed algorithms are run for 10 times. It is a two-pronged experimentation which provides a comparison against the existing heuristics in the literature as well as a real-word case study.

1. We compare the experimental values obtained by our approaches to those resulting from four other state-of-the-art algorithms which are 2-PIA, GRASP-SR, GRASP-PR, EA4OP and ALNS, depicted in Section 5.3. For adapting the benchmark to MODTOP model, we relax the capacity constraints by assuming that the destination points have infinite capacities. Furthermore, as the standard TOP model includes a single origin and destination nodes, we suppose that this node is duplicated to represent both depots and hospitals.
2. We provide a comparison of HGA and MA executed under Cplex version 12.6.2 with a time limit of three hours. For this reason, we generate a set of instances, inspired from Fischetti benchmark, to evaluate the performance of our approaches on a real-word case (Section 5.4).

5.1. Benchmark description

For performance assessment, we carried out 135 benchmark instances grouped into three classes, called 'generations'. Each generation contains 45 problems with up to 400 vertices. The corresponding T_{max} value for each instance is computed as 50% of the shortest Hamiltonian cycle length [39]. The tested benchmark is a TSP-based data set adapted by Fischetti et al. [17] to handle TOP. It is available at the following link <http://www.mech.kuleuven.be/en/cib/op>. The generations are based on the profit p_i detailed in the following rules.

Table 1
Metaheuristics parameter tuning.

Parameter	Considered values	Final value
MA		
Population size	100, 200, 300	300
Number of generations	100, 200, 300	300
Crossover probability	0.7, 0.8	0.7
Mutation probability	0.05, 0.1	0.05
HGA		
Population size	100, 200, 300	300
Number of generations	100, 200, 300	300
Crossover probability	0.7, 0.8	0.8
Mutation probability	0.05, 0.1	0.05

For all vertices $i \in V$:

- Generation 1: $p_i := 1$;
- Generation 2: $p_i := 1 + (7141 \times i + 73) \bmod(100)$;
- Generation 3: $p_i := 1 + (\frac{99 * d_{ij}}{d_{max}})$;

where V is the set of vertices, d_{ij} is the distance from vertex 1 to vertex i and d_{max} is the distance from vertex 1 to its farthest vertex.

5.2. Parameters settings

The performance of designed metaheuristics depends inherently of its parameter settings. To tune the different parameters, we apply an automatic procedure called F-Race to determine the best configuration. The F-Race is an offline automatic statistical procedure proposed by [40] to enable configuration of parameterized algorithms. The tuning was performed on a random selection of 10 large instances from the benchmark and the real case instances, with the total time limit set to 600 s. To evaluate the performance of each parameter, we test each considered value from Table 1 while fixing the other parameters to their final values. For each algorithm, the total number of configurations in this preliminary experiment is equal to $3 \times 3 \times 2 \times 2 = 36$. We report in Table 1, the set of candidate configurations and the final retained values.

Table 2
Solution approaches details.

Name	Description	Processor	Reference
2-PIA	Two-parameter iterative algorithm	Intel Xeon, 1.9 GHz	[41]
GARSP-PR	GRASP with path relinking	Intel Xeon, 1.9 GHz	[42]
GARSP-SR	GRASP with segment remove	Intel Core i7 with, 3.4 GHz	[43]
EA4OP	Evolutionary algorithm	Intel Xeon, 1.9 GHz	[44]
ALNS	Adaptive large neighborhood search	Quad-core Intel Xeon E5, 2.2 GHz	[45]
HGA	Hybrid genetic algorithm	Intel Core i7-7500U, 2.5 GHz	This paper
MA	Memetic algorithm	Intel Core i7-7500U, 2.5 GHz	This paper

5.3. Comparison with existing algorithms

In order to investigate the proposed algorithms' performance in solving MODTOP model, we outline and statistically compare the obtained results with state-of-the-art methods. For the sake of completeness, we also report the processor details of the computing environments used to test the five solution methods (Table 2).

The CPU time depends on a variety of factors, such as hardware, compiler and programming language used to develop the different approaches. Therefore, we did not directly compare the executing time efficiency of the algorithms.

Tables 3–5 summarize the test results of our proposed approaches and those of the state-of-the-art algorithms for all benchmark instances. The first three columns describe the instances and report, the name, T_{max} value and the best known solution (BKS), respectively. For each algorithm, columns 'Best' report the best found solutions and columns 'PRE' report the percentages relative error (PRE) calculated as follows.

$$PRE = 100 \times (BKS - Best)/BKS.$$

- The bold values mean the best-so-far results found by the algorithms.
- * means BKS achieved.
- - means not available value.

Table 3 presents the results for generation 1 instances. As shown, HGA reaches the BKS for 30 instances with an overall average equals to 0.25% which is better than 2PIA. MA find the BKS for 38 instances of generation 1 which is equivalent to 84.4%. With regard to the number of times that we reached the BKS in 45 instances, MA outperforms HGA (30/45 = 66.7%), 2-PIA (21/45 = 46.7%), GRASP-PR (32/45 = 71.1%) and EA4OP (30/45 = 66.7%). However, it was not the case compared to ALNS that identifies the BKS in 42/45 = 93.3% instances.

Table 4 summarizes the results of generation 2 and shows that, in terms of the average, ALNS ranks first among all the algorithms with 0.05%, MA is in the second rank with 0.06% followed by HGA with 0.16%. According to the same table, MA obtained the BKS in 33 instances which is the best result compared to all the other algorithms.

In Table 5, we eliminate GRASP-SR and ALNS from the comparison since there are 6 unvalued instances. MA outperforms all the rest of algorithms in terms of average (0.2%) and number of BKS found (24 instances).

While focusing on HGA and MA results, we can show that the integration of LS heuristic improved the solution quality of HGA. There is a decrease of the average by 0.17% in generation 1 and 0.1% in generation 2 and generation 3.

In Table 6, we provide the number of times the BKS is found (#Best) and the average relative percentage deviation (ARPD) over the total number of benchmark instances for the generations. Since there are unaddressed instances for GRASP-SR and ALNS algorithms, we excluded 18 instances (6 for each generation) from the data set in order to ensure a fair comparison. For GRASP-SR, there is no indication about instances Berlin52,

ts225 and a280 in all the generations. For ALNS, there are no available values for instances tps225, rat99 and rat195 in generation 3. Therefore, the missing instances are eliminated from the comparative study and only 117 instances are used. The ARPD is calculated as follows.

$$ARPD = \sum_{i=1}^{117} \left(\frac{BKS_i - Best_i}{BKS_i} \right) \times 100,$$

where $Best_i$ and BKS_i denote, respectively, the objective function value and the best known solution of problem instance i .

The average computational time (AVG Time) depends on a variety of factors, such as hardware, compiler and programming language used to develop the different algorithms (Table 2). Therefore, we did not deeply compare the run time efficiency of the algorithms.

Table 6 indicates that for the number of times the BKS is found, MA reached the best so far solutions in 86 instances which is 73.5% of the tested instances. For HGA, the #BKS is equal to 64 which is 54.7%. Consequently, MA and HGA outperform 2-PIA, GRASP-PR and EA4OP. According to the same table, for the average relative percentage deviation criterion, HGA calculated 0.20%, which is better than the results of EA4OP (0.21%), GRASP-PR (0.27%) and 2-PIA (0.44%). As it can be seen for MA, the ARPD achieves 0.0675% and performs better than all the other approaches except ALNS with a difference of 0.005% but in terms of time MA is speedier than ALNS with 147.32 s. More illustration for the results are depicted in Fig. 7.

From Table 6, we can obviously observe that 2-PIA, with the worst results in terms of #Best and ARPD, is the fastest algorithm with 1.62 s. For HGA and MA, the AVG time is equal to 2.28 and 3.95 s, respectively. However, ALNS that outperformed the others in #Best and ARPD, is ranked the last one with a huge difference for the computational performance (AVG Time = 151.27 s).

The AVG Time values are plotted in Fig. 8 that shows a comparison of (a) HGA and MA computational performance and (b) HGA and MA vs all the used algorithms.

We further compare statistically the algorithms outcomes in terms of percentage relative error (PRE) using a group of one-sided paired-samples T-tests. The T-test is able to carry out a comparison between the means of two algorithms in order to conclude whether it is significant or not. Since MA outperforms HGA in almost all experiments (as shown in Fig. 7), we choose to perform the t-test by comparing MA against the other methods at a confidence level $\alpha = 0.05$.

The statistical results reported in Table 7 shows that MA outperforms, in all generations, 2-PIA, GRASP-PR and EA4OP with a general p -value equals to 0.00, 0.00 and 0.03, respectively. Compared to HGA, the test is significant on generation 1 (p -value = 0.04) and generation 3 (p -value = 0.02) but it is not the case for generation 2 where p -value > 0.05. Regarding GRASP-SR and ALNS, the proposed MA shows a competitive results with p -value > 0.05.

Table 3
Comparison with existing algorithms for generation 1 instances.

Instance	HGA		MA		2-PIA		GRASP-SR		GRASP-PR		EA4OP		ALNS			
	Name	Tmax	BKS	Best	%PRE	Best	%PRE	Best	%PRE	Best	%PRE	Best	%PRE	Best	%PRE	
att48	5314	31	31	*	31	*	31	*	31	*	31	*	31	*		
gr48	2523	31	31	*	31	*	31	*	31	*	31	*	31	*		
hk48	5731	30	30	*	30	*	30	*	30	*	30	*	30	*		
eil51	213	29	29	*	29	*	29	*	29	*	29	*	29	*		
berlin52	3771	37	37	*	37	*	37	*	–	–	37	*	37	*		
brazil58	12689	46	46	*	46	*	46	*	46	*	46	*	46	*		
st70	338	43	43	*	43	*	43	*	43	*	43	*	43	*		
eil76	269	47	47	*	47	*	46	2.13	47	*	47	*	46	2.13		
pr76	54080	49	49	*	49	*	49	*	49	*	49	*	49	*		
gr96	27605	64	64	*	64	*	64	*	64	*	64	*	64	*		
rat99	606	52	51	1.92	51	*	51	1.92	52	*	52	*	52	*		
kroA100	10641	56	55	1.79	55	*	56	*	56	*	56	*	55	1.79		
kroB100	11071	58	58	*	58	*	58	*	58	*	58	*	57	1.72		
kroC100	10375	56	56	*	56	*	56	*	56	*	56	*	56	*		
kroD100	10647	59	59	*	59	*	59	*	59	*	59	*	58	1.69		
kroE100	10375	57	57	*	57	*	55	3.51	57	*	57	*	57	*		
rd100	3955	61	61	*	61	*	61	*	61	*	61	*	61	*		
eil101	315	64	63	1.56	64	*	63	1.56	64	*	64	*	64	*		
lin105	7190	66	65	1.52	65	*	66	*	66	*	66	*	66	*		
pr107	22152	54	52	3.70	53	1.85	54	*	54	*	54	*	54	*		
gr120	3471	75	75	*	75	*	74	1.33	75	*	75	*	74	1.33		
pr124	29515	75	75	*	75	*	75	*	75	*	75	*	75	*		
bier127	59141	103	101	1.94	103	*	103	*	103	*	103	*	103	*		
pr136	4386	71	71	*	71	*	69	2.82	71	*	70	1.40	71	*		
gr137	34927	81	81	*	81	*	81	*	81	*	81	*	78	3.70		
pr144	29269	77	77	*	77	*	73	5.19	77	*	77	*	77	*		
kroA150	13262	86	86	*	86	*	85	1.16	86	*	86	*	86	*		
kroB150	13065	87	86	1.15	86	1.15	86	1.15	87	*	86	1.15	86	1.15		
pr152	36841	77	70	9.09	77	*	76	1.30	77	*	77	*	77	*		
u159	21040	93	92	1.08	92	1.08	82	11.83	93	*	92	1.08	92	1.08		
rat195	1162	102	102	*	102	*	99	2.94	102	*	102	2.94	99	2.94		
d198	7890	123	123	*	123	*	120	2.44	123	*	123	0.81	123	*		
kroA200	14684	117	115	1.71	117	*	112	4.27	117	*	117	*	117	*		
kroB200	14719	119	119	*	119	*	117	1.68	119	*	119	0.84	119	*		
gr202	20080	145	147	*	147	*	140	3.45	145	*	147	*	145	*		
ts225	63322	124	123	1.60	124	*	124	*	124	*	124	*	124	*		
tsp225	1958	129	127	1.55	127	1.55	117	9.30	–	–	126	2.33	127	1.55		
pr226	40185	126	126	*	126	*	121	3.97	126	*	126	*	126	*		
gr229	67301	176	173	1.70	173	1.70	174	1.14	175	0.56	174	1.14	176	*		
gil262	1189	158	158	*	158	*	150	5.06	158	*	151	4.43	156	1.27		
pr264	24568	132	132	*	132	*	132	*	132	*	132	*	132	*		
a280	1290	147	143	2.72	143	2.72	133	9.52	–	–	143	2.72	143	2.72		
pr299	24096	162	162	*	162	*	154	4.94	162	*	158	2.47	160	1.23		
lin318	21015	205	205	*	205	*	194	5.37	205	*	200	2.44	202	1.46		
rd400	7641	239	237	0.84	237	0.84	218	8.79	235	1.67	225	5.86	234	2.09		
Average				0.25		0.08		0.72		0.02		0.22		0.21		0.05

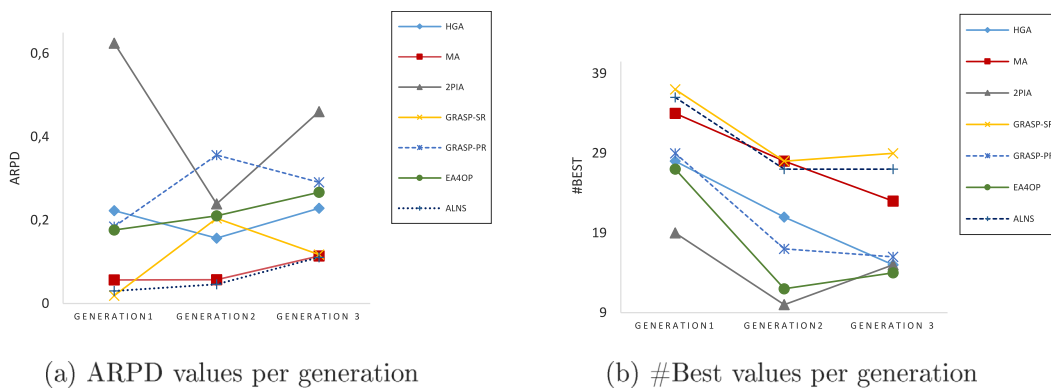


Fig. 7. Comparison of all the algorithms based on ARPD and #Best results.

5.4. Results for real-case instances

Since no benchmark instances are readily available for the studied problem, there is a need to generate the test instances

to evaluate HGA. We should point out that in the handled variant of TOP, we consider that: (1) the starting node (depot) is different from the ending node (hospital), so that the itinerary route is open, (2) each hospital can receive a maximum number of specimens, thus there is a hospital capacity constraint.

Table 4
Comparison with existing algorithms for generation 2 instances.

Instance Name	HGA		MA		2-PIA		GRASP-SR		GRASP-PR		EA4OP		ALNS			
	Tmax	BKS	Best	%PRE	Best	%PRE	Best	%PRE	Best	%PRE	Best	%PRE	Best	%PRE		
att48	5314	1717	1717	*	1717	*	1717	*	1717	*	1717	*	1717	*		
gr48	2523	1761	1761	*	1761	*	1750	0.62	1761	*	1761	*	1749	0.68		
hk48	5731	1614	1614	*	1614	*	1614	*	1614	*	1614	*	1614	*		
eil51	213	1674	1674	*	1674	*	1674	*	1674	*	1674	*	1668	0.36		
Berlin52	3771	1897	1897	*	1897	*	1897	*	-	-	1897	*	1897	*		
brazil58	12 698	2220	2218	0.09	2220	*	2220	*	2220	*	2220	*	2218	0.09		
st70	338	2286	2285	0.04	2286	*	2285	0.04	2286	*	2286	*	2285	0.04		
eil76	269	2550	2505	1.76	2550	*	2540	0.39	2550	*	2550	*	2550	*		
pr76	54 080	2708	2701	0.26	2708	*	2708	*	2708	*	2708	*	2708	*		
gr96	27 605	3396	3425	*	3396	*	3394	0.06	3425	*	3396	*	3394	0.06		
rat99	606	2944	2944	*	2944	*	2932	0.41	2944	*	2944	*	2944	*		
kroA100	10 641	3212	3181	0.97	3212	*	3212	*	3212	*	3212	*	3212	*		
kroB100	11 071	3241	3237	0.12	3239	0.06	3239	0.06	3241	*	3241	*	3238	0.09		
kroC100	10 375	2947	2924	0.78	2928	0.65	2947	*	2947	*	2909	1.29	2931	0.54		
kroD100	10 647	3307	3307	*	3307	*	3295	0.36	3307	*	3307	*	3307	*		
kroE100	11 034	3090	3090	*	3090	*	3090	*	3090	*	3082	0.26	3082	0.26		
rd100	3955	3359	3351	0.24	3359	*	3351	0.24	3359	*	3351	0.24	3359	*		
eil101	315	3655	3645	0.27	3655	*	3636	0.52	3665	*	3643	0.33	3655	*		
lin105	7190	3544	3544	*	3544	*	3536	0.23	3544	*	3544	*	3530	0.40		
pr107	22 152	2667	2660	0.26	2667	*	2667	*	2667	*	2667	*	2667	*		
gr120	3471	4371	4371	*	4371	*	4358	0.30	4371	*	4371	*	4356	0.34		
pr124	29 515	3917	3840	1.97	3917	*	3917	*	3917	*	3901	0.41	3899	0.46		
bier127	59 141	5383	5383	*	5383	*	5328	1.02	5379	0.07	5331	0.97	5381	0.04		
pr136	48 386	4309	4309	*	4309	*	4244	1.51	4309	*	4228	1.88	4309	*		
gr137	34 927	4286	4283	0.06	4286	*	4281	0.12	4286	*	4270	0.37	4099	4.36		
pr144	29 269	4003	4003	*	4003	*	3963	1.00	4003	*	4003	*	3965	0.95		
kroA150	13 262	4918	4913	0.10	4915	0.06	4913	0.10	4915	0.06	4842	1.55	4902	0.33		
kroB150	13 065	4869	4869	*	4869	*	4853	0.33	4869	*	4853	0.33	4869	*		
pr152	36 841	4279	4275	0.09	4275	0.09	4269	0.23	4279	*	4227	1.22	4245	0.79		
u159	21 040	4960	4960	*	4960	*	4938	0.44	4960	*	4889	1.43	4941	0.38		
rat195	1162	5791	5790	0.02	5790	0.02	5666	2.16	5786	0.86	5612	3.09	5703	1.52		
d198	7890	6670	6670	*	6670	*	6622	0.72	6669	0.015	6625	0.67	6660	0.15		
kroA200	14 684	6547	6547	*	6547	*	6461	1.31	6544	0.046	6279	4.09	6534	0.20		
kroB200	14 719	6419	6409	0.16	6409	0.16	6328	1.42	6404	0.234	6282	2.13	6278	2.20		
gr202	20 080	7789	7848	*	7848	*	7703	1.10	7789	*	7659	1.67	7789	*		
ts225	63 322	6834	6784	0.73	6808	0.38	6749	1.24	6808	0.38	6743	1.33	6819	0.22		
tsp225	1958	6987	6936	0.73	6936	0.73	6936	0.73	-	-	6818	2.42	6936	0.73		
pr226	40 185	6662	6615	*	6615	*	6646	0.24	6662	*	6621	0.62	6658	0.06		
gr229	67 301	9177	9187	*	9187	*	9111	0.72	9151	0.28	9046	1.43	9174	0.03		
gli262	1189	8321	8100	2.66	8212	1.3	8100	2.66	8286	0.42	7907	4.98	8175	1.75		
pr264	24 568	6654	6654	*	6654	*	6244	6.16	6406	3.73	6654	*	6173	7.23		
a280	1290	8428	8222	2.44	8350	0.9	8269	1.89	-	-	8021	4.83	8304	1.47		
pr299	24 096	9182	8689	5.37	9013	1.84	9060	1.33	9165	0.19	8846	3.66	9112	0.76		
lin318	21 015	10 923	10 900	*	10 900	*	10 724	1.82	9165	16.09	10 424	4.57	10 866	0.52		
rd400	7641	13 652	13 255	3.14	13 309	2.51	13 255	2.91	13 274	2.77	12 617	7.58	13 442	1.54		
Average				0.16		0.06		0.25		-		0.4		0.21		0.05

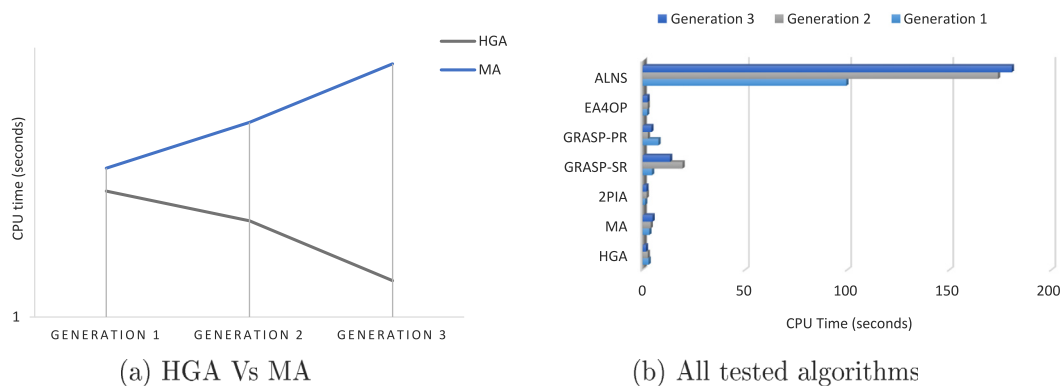


Fig. 8. Comparison of the average computational time per generation.

A real data set of 30 instances has been generated assuming that there are 6 hospitals and 6 depots. Each hospital has a maximum capacity of specimen tests (Table 8) and in each depot, there are a prefixed number of ambulances (Table 9).

Inspired from Fischetti benchmark, the patients scores are randomly chosen from these values {5, 15, 25, 35}. The maximum total travel time in a single day is $T_{max} = 600$ (min).

Given the aforementioned data set, the problem instances were solved using Cplex 12.6.2 with default settings. We got to

Table 5
Comparison with existing algorithms for generation 3 instances.

Instance Name	HGA		MA		2-PIA		GRASP-SR		GRASP-PR		EA4OP		ALNS	
	Tmax	BKS	Best	%PRE	Best	%PRE	Best	%PRE	Best	%PRE	Best	%PRE	Best	%PRE
att48	5314	1049	1049	*	1049	*	1049	*	1049	*	1049	*	1049	*
gr48	2523	1480	1480	*	1480	*	1480	*	1480	*	1480	*	1480	*
hk48	5731	1764	1764	*	1764	*	1764	*	1764	*	1764	*	1764	*
eil51	213	1399	1399	*	1399	*	1399	*	1399	*	1398	0.07	1399	*
berlin52	3771	1036	1036	*	1036	*	1036	*	–	–	1036	*	1034	0.19
brazil58	12 698	1702	1702	*	1702	*	1702	*	1702	*	1702	*	1702	*
st70	338	2108	2108	*	2108	*	2108	*	2108	*	2108	*	2108	*
eil76	269	2467	2467	*	2467	*	2461	0.24	2467	*	2462	0.20	2467	*
pr76	54080	2430	2430	*	2430	*	2430	*	2430	*	2430	*	2430	*
gr96	27 605	3170	3166	0.13	3166	0.13	3170	*	3170	*	3153	0.54	3166	0.13
rat99	606	2908	2886	0.76	2892	0.55	2896	0.41	2908	*	2881	0.93	2886	0.76
kroA100	10 641	3211	3211	*	3211	*	3211	*	3211	*	3180	0.97	3211	*
kroB100	11 071	2804	2785	0.68	2804	*	2804	*	2804	*	2785	0.68	2804	*
kroC100	10 375	3155	3155	*	3155	*	3155	*	3155	*	3149	0.19	3155	*
kroD100	10 647	3167	3141	0.82	3155	0.38	3123	1.39	3167	*	3167	*	3141	0.82
kroE100	11 034	3049	2049	*	3049	*	3027	0.72	3049	*	3049	*	3049	*
rd100	3955	2926	2923	0.10	2926	*	2924	0.07	2926	*	2924	0.07	2923	0.10
eil101	315	3345	3345	*	3345	*	3333	0.36	3345	*	3322	0.69	3345	*
lin105	7190	2986	2973	0.44	2986	*	2986	*	2986	*	2973	0.44	2986	*
pr107	22 152	1877	1802	4.00	1854	1.22	1877	*	1877	*	1877	*	1802	4.00
gr120	3471	3779	3748	0.82	3751	0.74	3736	1.14	3779	*	3745	0.90	3748	0.82
pr124	29 515	3557	3557	*	3455	2.87	3517	1.12	3557	*	3549	0.22	3455	2.87
bier127	59 141	2365	2361	0.17	2361	0.17	2356	0.38	2356	0.38	2332	1.40	2361	0.17
pr136	48 386	4390	4390	*	4390	*	4390	*	4390	*	4380	0.23	4390	*
gr137	34 927	3954	4954	*	4954	*	3928	0.66	3979	*	3926	0.71	3954	*
pr144	29 269	3745	3700	1.20	3710	0.93	3633	2.99	3741	1.79	3745	*	3700	1.20
kroA150	13 262	5039	5019	0.40	5030	0.17	5037	0.04	5039	*	5018	0.42	5019	0.40
kroB150	13 065	5314	5314	*	5314	*	5267	0.88	5314	*	5272	0.79	5314	*
pr152	36 841	3905	3902	0.08	3905	*	3557	8.91	3905	*	3905	*	3902	0.08
u159	21 040	5272	5272	*	5272	*	5272	*	5272	*	5272	*	5272	*
rat195	1162	6195	6143	0.83	6152	0.7	6174	0.34	6191	0.06	6086	1.76	6139	0.90
d198	7890	6320	6292	0.44	6320	*	5985	5.30	6163	2.48	6162	2.50	6290	0.47
kroA200	14 684	6123	6119	0.06	6110	0.21	6048	1.22	6123	*	6084	0.64	6114	0.15
kroB200	14 719	6266	6213	0.85	6211	0.88	6251	0.24	6266	*	6190	1.21	6213	0.85
gr202	20 080	8616	8600	0.18	8560	0.65	8111	5.86	8469	1.89	8419	2.29	8605	0.13
ts225	63 322	7575	7490	1.12	7575	*	7149	5.62	–	–	7510	0.86	7575	*
tsp225	1958	7740	7575	*	7575	*	7353	5.00	7575	*	7565	2.26	7488	3.26
pr226	40 185	6993	6923	1.00	6977	0.23	6652	4.88	6912	1.16	6964	0.41	6908	1.22
gr229	67 301	6328	6311	0.27	6299	0.46	6190	2.18	6235	1.76	6205	1.94	6297	0.49
gil262	1189	9246	9178	0.73	9220	0.28	8915	3.58	9128	1.28	8922	3.50	9094	1.64
pr264	24 568	8137	7754	4.70	8137	1.89	7820	3.90	8137	*	7959	2.19	8068	0.85
a280	1290	9774	8702	10.97	8724	10.74	8719	10.79	–	–	9426	3.56	8684	11.15
pr299	24 096	10 343	9959	3.71	10 201	1.37	10 305	0.37	10 277	0.78	10 033	3.00	9959	3.71
lin318	21 015	10 368	10 273	1.05	10 330	0.37	9909	4.43	10 275	1.03	9758	5.88	10 273	0.92
rd400	7641	13 223	13 088	1.07	13 106	0.89	12 828	2.99	13 070	1.20	12 678	4.12	13 088	1.02
Average				0.3		0.17		0.6		–		0.32		0.29

Table 6
Number of best solutions found and the average relative percentage deviation for each generation.

	HGA	MA	2PIA	GRASP-SR	GRASP-PR	E4OP	ALNS
#Best							
Generation 1	28	34	19	37	29	27	36
Generation 2	21	28	10	28	17	12	27
Generation 3	15	24	15	29	16	14	27
All	64	86	44	94	62	53	90
ARPD							
Generation 1	0.2229	0.0565	0.6247	0.019	0.1847	0.1764	0.0301
Generation 2	0.1568	0.057	0.2390	0.2043	0.3562	0.2103	0.0461
Generation 3	0.2285	0.089	0.4602	0.1175	0.2909	0.2668	0.1109
All	0.2027	0.0675	0.4413	0.1136	0.2772	0.2178	0.0623
AVG Time (s)							
Generation 1	2.87	3.21	1.14	4.53	7.66	2.12	99.51
Generation 2	2.43	3.89	1.92	19.45	2.48	2.38	173.77
Generation 3	1.54	4.76	1.8	13.35	4.18	2.31	180.55
All	2.28	3.95	1.62	12.44	4.88	2.27	151.27

solve small instances with a time limit of 10 800 s. Table 10 presents the comparison of our algorithms results (Best) with the

exact solution values generated by Cplex (F_{cplex}). For each algorithm, we calculate the average percentage deviation computed as: $\%Gap = 100 \times (F_{cplex} - Best) / F_{cplex}$.

The columns of Table 10 correspond to, the instance name, the number of patients (#patients), the best solution reached (Best), the average deviation (%Gap) and the average computational time (Time), respectively.

- In the small-scaled instances in which the number of patients is less than 25, Cplex was able to find optimal solutions within three hours. For the rest of instances, Cplex either terminates with an out of memory error or it is stopped as we have set the maximum resolution time, while the HGA and MA continue to generate solutions within a very limited CPU time. This fact underlines the usefulness of metaheuristics in solving such NP-hard problems.
- The computational difficulty increases significantly with problem size. For instance, Cplex takes more than 2 hours to solve the problem instance *Cov*₁₀. To handle the same instance, HGA finds a near optimal solution in only 7.8 s, with a %Gap that amounts to only 0.5%. Although, MA succeeds to reach the best solution in 10.9 s. Moreover, the CPU time for the MA and HGA grows polynomially with

Table 7

Analytical results of the paired-samples t-tests with respect to PRE for three generation instances. Bold values mean p -value ≤ 0.05 .

MA vs	HGA	2PIA	GRASP SR	GRASP-PR	EA4OP	ALNS
Paired difference PRE						
Generation 1	-0.4530	-1,7044	0,1125	-0,3846	-0,3595	0,0794
Generation 2	-0.2992	-0,5459	-0,4419	-0,8977	-0,4600	0,0327
Generation 3	-0.4174	-1,1125	-0,0843	-0,6046	-0,3523	-0,0646
t-value						
Generation 1	-1.6854	-3.8951	1.2857	-1.7757	-2.2278	0.8613
Generation 2	-1.5296	-2.6883	-1.0224	-3.1243	-1.9690	0.3391
Generation 3	-2.0871	-3.2110	-0.6427	-2.6300	-2.0512	-0.2566
p-value						
Generation 1	0.0479	0.00010	0.1012	0.0398	0.0144	0.1958
Generation 2	0.0651	0.0044	0.1549	0.0012	0.0262	0.3677
Generation 3	0.0201	0.00097	0.2611	0.00516	0.0218	0.3990

Table 8

Specimens tests per Hospital.

Hospital	Capacity
O1. Hospital Charles Nicolle	10
O2. Hospital Abderrahmen Mami	25
O3. Hospital Mongi Slim	32
O4. Hospital Régional de Ben Arous	15
O5. Hospital Régional de Khéreddine	20

Table 9

Ambulances per Depot.

Depot	Nb ambulances
D1	3
D2	2
D3	3
O1	2
O4	3

Table 10

Results and comparison with optimal solutions.

Instance	#patients	Cplex		HGA			MA		
		Best	Time	Best	%Gap	Time	Best	%Gap	Time
Cov ₁	5	65	74.76	65	0	0.8	65	0	2.77
Cov ₂	8	67	180.33	67	0	0.9	67	0	2.98
Cov ₃	10	126	320.66	126	0	1.2	126	0	2.4
Cov ₄	12	113	709.2	113	0	2.3	113	0	4.6
Cov ₅	16	304	1200	304	0	5.1	304	0	7.1
Cov ₆	18	312	6000	312	0	3.4	312	0	5.6
Cov ₇	20	470	8021	470	0	2.3	470	0	2.32
Cov ₈	22	622	7332	622	0	4.3	622	0	5.6
Cov ₉	24	1174	7820	1174	0	8.0	1174	0	10.2
Cov ₁₀	25	1200	9200	1174	0.5	7.8	1200	0	10.9
Cov ₁₁	52	-	-	1562	-	6.3	1724	-	11.9
Cov ₁₂	57	-	-	1720	-	7.6	1890	-	9.4
Cov ₁₃	66	-	-	2112	-	5.8	2370	-	6.5
Cov ₁₄	72	-	-	1800	-	4.6	2015	-	8.9
Cov ₁₅	79	-	-	1978	-	5.7	2440	-	9.2
Cov ₁₆	83	-	-	2490	-	7.7	2766	-	11.3
Cov ₁₇	87	-	-	2610	-	6.3	2722	-	10.8
Cov ₁₈	90	-	-	2790	-	2.5	3054	-	5.3
Cov ₁₉	97	-	-	3300	-	4.5	3753	-	6.7
Cov ₂₀	101	-	-	3060	-	3.4	3299	-	4.4
Cov ₂₁	121	-	-	4114	-	8.7	4522	-	11.9
Cov ₂₂	156	-	-	5148	-	7.4	5223	-	11.3
Cov ₂₃	178	-	-	5874	-	10.3	5978	-	14.2
Cov ₂₄	180	-	-	5940	-	9.0	6319	-	11.7
Cov ₂₅	212	-	-	6360	-	7.8	7343	-	13.3
Cov ₂₆	223	-	-	7130	-	10.7	7400	-	12.6
Cov ₂₇	245	-	-	8085	-	9.7	8309	-	11.9
Cov ₂₈	257	-	-	7190	-	8.9	7514	-	11.5
Cov ₂₉	260	-	-	7280	-	10.4	7503	-	14.6
Cov ₃₀	300	-	-	9000	-	9.7	9410	-	13.7
Averages						7.02			8.85

the instance size, while the computational requirements of Cplex seem to increase exponentially.

- Among all instances solved by Cplex, HGA finds the optimal solutions in 9 instances out of 10. In fact, the average gap among all instances is equal to 0.5% which is considerably interesting. MA attains all the optimal values generated by Cplex with a general AVG time equals to 8.85 s which is a worst result compared to HGA that solved the instances in 7.02 s. In terms of solution values, MA outperforms HGA in all the tested data set.

6. Conclusions and future works

COVID-19 related issues become a top priority for researchers worldwide, notably in combinatorial optimization field. In this paper, we studied the collection and transport of COVID-19 specimens. The problem can be described as follows. Given (1) a set of suspected patients requesting COVID-19 tests at home, each one of them is associated with a priority and (2) a set of ambulances located in different hospitals. The objective is to select the subset of urgent patients to be visited in priority as well as to determine the order and the optimal itinerary to collect the COVID-19 specimens test. Some restrictions are imposed in our model, such as the hospital capacity and the daily working time of the ambulance driver. We modeled the collection and transportation of COVID-19 specimens as a new variant of the team orienteering problem, named multi-origin-destination team orienteering problem. Given the problem complexity (NP-hard) a hybrid genetic algorithm combining the k-means along with the evolutionary operators is proposed. A memetic algorithm considering a local search is also implemented to improve the convergence speed and fully exploit the solution space. Compared to the current state-of-the-art, the two algorithms are proved to be efficient as they matched in many cases the best reported results on different TOP benchmarks. Experiments on real-case benchmark data sets indicated that the both HGA and MA produce high quality solutions with reasonable computational requirements for small sized instances. The numerical results for large sized instances, supported by statistical tests, prove the efficiency of HGA in generating better approximation of the global optimum. However, MA is computationally less efficient than the HGA, this may be explained by the constructive nature of the considered neighborhoods.

As future works, we can consider the dynamic case where new requests are coming while serving the actual patient, also we can solve different objective functions in a multi-objective problem. Finally, we suggest to integrate our proposed approaches into a decision support system in order to assist dynamically the ambulance drivers of COVID-19 specimens to accomplish their works optimally.

CRediT authorship contribution statement

Takwa Tlili: Conceptualization, Writing methodology, Experimental study. **Hela Masri:** Problem formulation, Writing, Real-case study. **Saoussen Krichen:** Reviewing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] J. Pacheco, M. Laguna, Vehicle routing for the urgent delivery of face shields during the COVID-19 pandemic, *J. Heuristics* 26 (2020) 619–635, <http://dx.doi.org/10.1007/s10732-020-09456-8>.
- [2] I.K. Singgih, 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), 2020, pp. 80–83, <http://dx.doi.org/10.1109/MECnIT48290.2020.9166664>.
- [3] M.-X. Zhang, H.-F. Yan, J.-Y. Wu, Y.-J. Zheng, Quarantine vehicle scheduling for transferring high-risk individuals in epidemic areas, *Int. J. Environ. Res. Public Health* 17 (7) (2020) <http://dx.doi.org/10.3390/ijerph17072275>.
- [4] D. Chen, S. Pan, Q. Chen, J. Liu, Vehicle routing problem of contactless joint distribution service during COVID-19 pandemic, *Transp. Res. Interdiscip. Perspect.* 8 (2020) 100233, <http://dx.doi.org/10.1016/j.trip.2020.100233>.
- [5] I.-M. Chao, B.L. Golden, E.A. Wasil, The team orienteering problem, *European J. Oper. Res.* 88 (3) (1996) 464–474, [http://dx.doi.org/10.1016/0377-2217\(94\)00289-4](http://dx.doi.org/10.1016/0377-2217(94)00289-4).
- [6] P. Vansteenwegen, W. Souffriau, G.V. Berghe, D.V. Oudheusden, A guided local search metaheuristic for the team orienteering problem, *European J. Oper. Res.* 196 (1) (2009) 118–127, <http://dx.doi.org/10.1016/j.ejor.2008.02.037>.
- [7] D.-C. Dang, R.N. Guibadij, A. Moukrim, An effective PSO-inspired algorithm for the team orienteering problem, *European J. Oper. Res.* 229 (2) (2013) 332–344, <http://dx.doi.org/10.1016/j.ejor.2013.02.049>.
- [8] L. Chen, E. Miller-Hooks, Optimal team deployment in urban search and rescue, *Transp. Res. B* 46 (8) (2012) 984–999, <http://dx.doi.org/10.1016/j.trb.2012.03.004>.
- [9] I. Baffo, P. Carotenuto, S. Rondine, An orienteering-based approach to manage emergency situation, *Transp. Res. Procedia* 22 (2017) 297–304, <http://dx.doi.org/10.1016/j.trpro.2017.03.036>.
- [10] S. Saeedvand, H.S. Aghdasi, J. Baites, Novel hybrid algorithm for team orienteering problem with time windows for rescue applications, *Appl. Soft Comput.* 96 (2020) 106700, <http://dx.doi.org/10.1016/j.asoc.2020.106700>.
- [11] T. Tlili, S. Abidi, S. Krichen, A mathematical model for efficient emergency transportation in a disaster situation, *Am. J. Emerg. Med.* 36 (9) (2018) 1585–1590, <http://dx.doi.org/10.1016/j.ajem.2018.01.039>.
- [12] S. Karakatić, V. Podgorelec, A survey of genetic algorithms for solving multi depot vehicle routing problem, *Appl. Soft Comput.* 27 (2015) 519–532, <http://dx.doi.org/10.1016/j.asoc.2014.11.005>.
- [13] F. Hammami, M. Rekik, L.C. Coelho, A hybrid adaptive large neighborhood search heuristic for the team orienteering problem, *Comput. Oper. Res.* 123 (2020) 105034, <http://dx.doi.org/10.1016/j.cor.2020.105034>.
- [14] Y. Amarouche, R.N. Guibadij, E. Chaalal, A. Moukrim, Effective neighborhood search with optimal splitting and adaptive memory for the team orienteering problem with time windows, *Comput. Oper. Res.* 123 (2020) 105039, <http://dx.doi.org/10.1016/j.cor.2020.105039>.
- [15] C. Orlic, N. Bianchessi, R. Roberti, W. Dullaert, The team orienteering problem with overlaps: An application in cash logistics, *Transp. Sci.* 54 (2) (2020) 470–487, <http://dx.doi.org/10.1287/trsc.2019.0923>.
- [16] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: A literature review, *Swarm Evol. Comput.* 2 (2012) 1–14, <http://dx.doi.org/10.1016/j.swevo.2011.11.003>.
- [17] M. Fischetti, J.S. González, P. Toth, Solving the orienteering problem through branch-and-cut, in: *Solving the Orienteering Problem Through Branch-and-Cut*, Vol. 10, 1998, pp. 133–148, <http://dx.doi.org/10.1007/s10288-006-0009-1>.
- [18] M.Q.H. Abadi, S. Rahmati, A. Sharifi, M. Ahmadi, HSSAGA: Designation and scheduling of nurses for taking care of COVID-19 patients using novel method of Hybrid Salp Swarm Algorithm and Genetic Algorithm, *Appl. Soft Comput.* 108 (2021) 107449, <http://dx.doi.org/10.1016/j.asoc.2021.107449>.
- [19] E.A. Rashed, S. Koder, H. Shirakami, R. Kawaguchi, K. Watanabe, A. Hirata, Knowledge discovery from emergency ambulance dispatch during COVID-19: A case study of Nagoya City, Japan, *J. Biomed. Inform.* 117 (2021) 103743, <http://dx.doi.org/10.1016/j.jbi.2021.103743>.
- [20] S. Cerna, H.H. Arcolezi, C. Guyeux, G. Royer-Fey, C. Chevallier, Machine learning-based forecasting of firemen ambulances' turnaround time in hospitals, considering the COVID-19 impact, *Appl. Soft Comput.* 109 (2021) 107561, <http://dx.doi.org/10.1016/j.asoc.2021.107561>.
- [21] N. Kumar, S. Susan, Particle swarm optimization of partitions and fuzzy order for fuzzy time series forecasting of COVID-19, *Appl. Soft Comput.* 110 (2021) 107611, <http://dx.doi.org/10.1016/j.asoc.2021.107611>.
- [22] A. Zahedi, A. Salehi-Amiri, N.R. Smith, M. Hajiaghahi-Keshmeli, Utilizing IoT to design a relief supply chain network for the SARS-COV-2 pandemic, *Appl. Soft Comput.* 104 (2021) 107210, <http://dx.doi.org/10.1016/j.asoc.2021.107210>.
- [23] B. Mosallanezhad, V.K. Chouhan, M.M. Paydar, M. Hajiaghahi-Keshmeli, Disaster relief supply chain design for personal protection equipment during the COVID-19 pandemic, *Appl. Soft Comput.* 112 (2021) 107809, <http://dx.doi.org/10.1016/j.asoc.2021.107809>.
- [24] J. Tassone, S. Choudhury, A comprehensive survey on the ambulance routing and location problems, 2020, [arXiv:2001.05288](https://arxiv.org/abs/2001.05288).
- [25] L. Özdamar, E. Ekinici, B. Küçükyazici, Emergency logistics planning in natural disasters, *Ann. Oper. Res.* 129 (2004) 217–245, <http://dx.doi.org/10.1023/B:ANOR.0000030690.27939.39>.
- [26] S. Wohlgemuth, R. Oloruntopa, U. Clausen, Dynamic vehicle routing with anticipation in disaster relief, *Socio-Econ. Plan. Sci.* 46 (4) (2012) 261–271, <http://dx.doi.org/10.1016/j.seps.2012.06.001>.
- [27] A.M. Campbell, D. Vandenbussche, W. Herrmann, Routing for relief efforts, *Transp. Sci.* 42 (2) (2008) 127–145, <http://dx.doi.org/10.1287/trsc.1070.0209>.
- [28] M. Keshtkaran, K. Ziarati, A. Bettinelli, D. Vigo, Enhanced exact solution methods for the team orienteering problem, *Int. J. Prod. Res.* 54 (2) (2016) 591–601, <http://dx.doi.org/10.1080/00207543.2015.1058982>.
- [29] S. Boussier, D. Feillet, Gendreau, An exact algorithm for team orienteering problems, *4OR* 5 (2007) 211–230, <http://dx.doi.org/10.1007/s10288-006-0009-1>.
- [30] C. Bayliss, A.A. Juan, C.S. Currie, J. Panadero, A learnheuristic approach for the team orienteering problem with aerial drone motion constraints, *Appl. Soft Comput.* 92 (2020) 106280, <http://dx.doi.org/10.1016/j.asoc.2020.106280>.
- [31] B.L. Golden, L. Levy, R. Vohra, The orienteering problem, *Nav. Res. Logist.* 34 (3) (1987) 307–318, [http://dx.doi.org/10.1002/1520-6750\(198706\)34:3<307::AID-NAV3220340302>3.0.CO;2-D](http://dx.doi.org/10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D).
- [32] Y. Wang, Q. Li, X. Guan, M. Xu, Y. Liu, H. Wang, Two-echelon collaborative multi-depot multi-period vehicle routing problem, *Expert Syst. Appl.* 167 (2021) 114201.
- [33] J.G.C. Costa, Y. Mei, M. Zhang, Cluster-based hyper-heuristic for large-scale vehicle routing problem, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8, <http://dx.doi.org/10.1109/CEC48606.2020.9185831>.
- [34] B.B. Bhusare, S.M. Bansode, Centroids initialization for K-means clustering using improved pillar algorithm, *Int. J. Adv. Res. Comput. Eng. Technol.* 3 (4) (2014) 1317–1322.
- [35] S.L. Yadav, A. Sohal, Comparative study of different selection techniques in genetic algorithm, *Int. J. Eng. Sci. Math.* 6 (3) (2017) 174–180.
- [36] A.M. Campbell, M. Savelsbergh, Efficient insertion heuristics for vehicle routing and scheduling problems, *Transp. Sci.* 38 (3) (2004) 369–378, <http://dx.doi.org/10.1287/trsc.1030.0046>.
- [37] T.P. Hong, H.S. Wang, W.C. Chen, Simultaneously applying multiple mutation operators in genetic algorithms, *J. Heuristics* 6 (4) (2000) 439–455, <http://dx.doi.org/10.1023/A:1009642825198>.
- [38] X. Wang, S. Shao, J. Tang, Iterative local-search heuristic for weighted vehicle routing problem, *IEEE Trans. Intell. Transp. Syst.* (2020) 1–11, <http://dx.doi.org/10.1109/ITITS.2020.2983398>.
- [39] K. Ostrowski, J. Karbowska-Chilinska, J. Koszelew, P. Zabielski, Evolution-inspired local improvement algorithm solving orienteering problem, *Ann. Oper. Res.* 253 (1) (2017) 519–543, <http://dx.doi.org/10.1007/s10479-016-2278-1>.
- [40] M. Birattari, Tuning Metaheuristics - A Machine Learning Perspective, in: *Studies in Computational Intelligence*, vol. 197, Springer, 2009, <http://dx.doi.org/10.1007/978-3-642-00483-4>.
- [41] J. Silberholz, B. Golden, The effective application of a new approach to the generalized orienteering problem, *J. Heuristics* 16 (3) (2010) 393–415, <http://dx.doi.org/10.1007/s10732-009-9104-8>.
- [42] V. Campos, R. Marti, GRASP with path relinking for the orienteering problem, *J. Oper. Res. Soc.* 65 (12) (2014) 1800–1813, <http://dx.doi.org/10.1057/jors.2013.156>.
- [43] M. Keshtkaran, K. Ziarati, A novel GRASP solution approach for the Orienteering Problem, *J. Heuristics* 22 (2016) 699–726, <http://dx.doi.org/10.1007/s10732-016-9316-7>.
- [44] G. Kobeaga, M. Merino, J.A. Lozano, An efficient evolutionary algorithm for the orienteering problem, *Comput. Oper. Res.* 90 (2018) 42–59, <http://dx.doi.org/10.1016/j.cor.2017.09.003>.
- [45] A. Santini, An adaptive large neighbourhood search algorithm for the orienteering problem, *Expert Syst. Appl.* 123 (2019) 154–167, <http://dx.doi.org/10.1016/j.eswa.2018.12.050>.