



Ambulance Routing Optimization Problem with Patient Triage Priority

Sofoclis P. Zambirinis¹ · Maria N. Anastasiadou¹ · Efthymoulos Kyriacou² · Charalambos Chrysostomou¹ · Michalis Mavrovouniotis¹

Received: 2 February 2026 / Revised: 23 July 2026 / Accepted: 4 August 2026
© The Author(s), under exclusive licence to ITS Japan 2026

Abstract

Emergency medical services (EMS) face critical challenges in efficiently allocating and routing ambulances to minimize response times, particularly in scenarios where high-risk patients must be prioritized under limited resources. This study investigates an ambulance routing problem (ARP) with patient triage priority, where the objective is to jointly assign ambulances to patients and determine their routes so as to minimize total travel time, defined as the sum of the ambulance-to-patient and patient-to-hospital travel times. A mixed-integer linear programming (MILP) model is proposed that integrates assignment and routing decisions, while capturing realistic EMS operational constraints such as patient prioritization and heterogeneous travel times. The proposed MILP is formulated as a multi-criteria optimization problem that prioritizes the transportation service of red-code patients while minimizing total travel time. To address the computational complexity of solving large-scale ARP instances, a metaheuristic based on Tabu Search (TS) is developed, incorporating a set of problem-specific local search operators. To evaluate the effectiveness of the MILP model and TS algorithm, a new set of benchmark instances is generated for the ARP with triage-based patient priority. Computational experiments demonstrate that the MILP model solves small-scale instances (up to 40 patients) optimally in less than one minute, but requires up to 32 minutes to solve large-scale instances (up to 150 patients) and more than 5 hours for instances up to 250 patients. The proposed TS algorithm finds optimal solutions for 20 of the 39 benchmark instances and achieves an average optimality gap below 0.7% in the remaining 19 instances, while requiring less than 42 seconds, on average, to solve the largest instances.

Keywords Ambulance routing problem · Emergency medical service · Mixed integer linear programming · Tabu search

1 Introduction

The efficient operation of Emergency Medical Services (EMS) is essential for public health, as improving patient flow can save lives and enhance patient outcomes [6, 16]. When an emergency call is received, dispatchers must rapidly determine which ambulance should be dispatched and which hospital to transport the patient to, with the objective of minimizing the time until definitive care is

provided [1, 20]. These decisions are further complicated by multiple operational constraints (e.g., high-priority emergency incidents). Typically, EMS systems follow simple dispatching rules such as assigning the closest available ambulance, but such myopic strategies can often be suboptimal [11]. Figure 1 illustrates the typical EMS workflow in four steps: 1) the call center receives an emergency call, 2) the dispatcher assigns an ambulance, 3) the ambulance travels to the patient, and 4) the patient is transported to the nearest hospital. For the third step, operations research provides powerful optimization tools for addressing the ambulance routing problem (ARP), with the potential to significantly improve the response time performance [19].

Despite the extensive literature on the ARP, most existing models rely on weighted objective functions to represent patient priority, which may allow lower-priority patients to be served before higher-priority patients [9, 34], while

✉ Michalis Mavrovouniotis
michalis.mavrovouniotis@eratosthenes.org.cy

¹ ERATOSTHENES Centre of Excellence, Cyprus University of Technology, Limassol, Cyprus

² Department of Electrical, Computer Engineering and Informatics, Cyprus University of Technology, Archiepiskopou Kyprianou 30, Limassol 3036, Cyprus

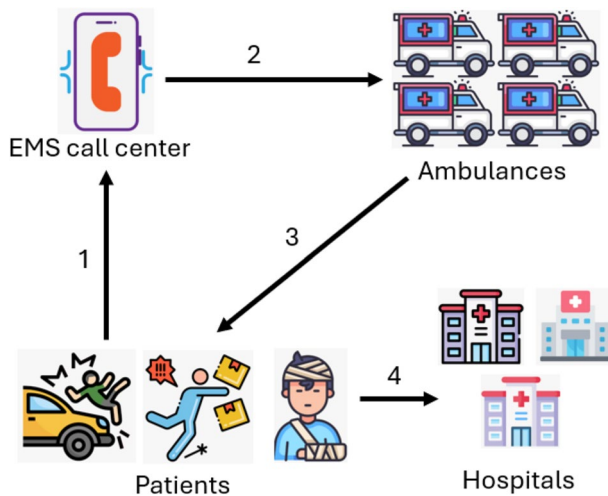


Fig. 1 Four steps in the ambulance-routing process

other models do not explicitly enforce patient triage-based prioritization [17]. Furthermore, several studies treat the assignment and routing decisions as separate optimization tasks [2]. This study investigates the ARP under triage-based patient prioritization [2, 9]. The basic ARP can be viewed as a variant of the vehicle routing problem with pickup and delivery [5], which is itself an $\mathcal{NP}$ -hard problem. Given a set of emergency incidents (i.e., patient location and severity) and a fleet of ambulances (i.e., positioned at base stations or strategic locations), the objective is to efficiently allocate ambulances to patients to minimize total travel time while satisfying critical operational constraints. In the investigated ARP with triage-based patient priority, patients are categorized into red-code patients who require immediate transportation to a hospital and green-code patients whose conditions are less urgent and can tolerate longer waiting times. A hard constraint of the particular ARP is that, at any stage, no green-code patient may be scheduled for transportation while an unassigned red-code patient is still awaiting service. Although other ARP formulations assign different weights to patient groups to reflect their urgency, they may still produce solutions in which green-code patients are served while red-code patients remain unserved [9, 17, 34]. Such an outcome is not permitted in the proposed ARP formulation.

A mixed integer linear programming (MILP) formulation is proposed to solve the ARP with patient triage priority. The proposed MILP jointly optimizes the ambulance-to-patient and patient-to-hospital assignments across the ambulance fleet. The model ensures that each ambulance transports no more than one patient at a time, in accordance with standard EMS protocols. Although the MILP provides an exact optimization framework for the ARP and produces optimal solutions for small instances – making it valuable for

evaluating heuristic methods and quantifying the benefit of optimization – it becomes computationally prohibitive for large-scale instances. To efficiently solve larger problem instances, a hybrid metaheuristic based on Tabu search (TS) [10, 13] and large neighborhood search (LNS) [27, 28] is proposed for the ARP.

In EMS operations, rapid decision-making is critical; therefore, generating high-quality routing decisions within a short time is essential. The proposed TS constructs an initial feasible solution, then iteratively improves it by exploring neighboring solutions using seven problem-specific search operators designed for the ARP. To avoid cycling and getting trapped in local optima, the proposed TS uses a tabu list that temporarily forbids reversing recent moves, and employs aspiration criteria to allow certain tabu moves if they produce significantly improved solutions [14]. To further enhance search diversification, LNS is applied to partially destroy and repair the current solution.

The main contributions of this study are threefold. First, we formulate a novel MILP allocation model for ambulance routing with single-patient assignments, incorporating realistic EMS-specific operational considerations. The proposed model jointly assigns patients to ambulances and ambulances to hospitals while prioritizing urgent patients over non-urgent patients through differentiated weighting of travel times to and from each patient category. Although the literature contains numerous studies addressing problems that appear to be similar, to the best of our knowledge, none of them considers the problem under the same assumptions, objectives, and prioritization strategy presented in this study. Second, we develop a specialized hybrid adaptive TS (HATS) algorithm that produces high-quality solutions for medium- and large-scale instances within practical computational times, particularly when the MILP model cannot provide a solution within an acceptable time limit. Third, a new benchmark dataset for the ARP with triage-based patient priority is generated to evaluate the effectiveness of the MILP model and the efficiency of the HATS algorithm.

The remainder of this paper is organized as follows. Section 2 reviews the literature on ARP and highlights the research gap addressed by this study. Section 3 defines the proposed problem and presents the MILP formulation with all parameters, decision variables, and constraints. Section 4 describes the proposed HATS algorithm in detail, including solution representation, neighborhood structure with the LNS component, and the diversification mechanism. Section 5 compares the performance of the MILP and HATS approaches on the proposed benchmark instances, and discusses the obtained results. Section 6 concludes the paper and outlines directions for future research.

2 Related Work

The literature recognizes ambulance route optimization as a key opportunity for improving EMS performance [4]. Three main variants of ARP have been studied in the literature: (i) the basic ARP, in which an ambulance picks up a patient and returns to its starting location [29]; (ii) the capacitated ARP, where hospitals and ambulances have limited capacities [33]; and (iii) the ARP with time windows, in which ambulances must arrive within predefined time intervals [24]. Several studies classify patients according to their severity and assign weighted priorities accordingly [25, 29, 34]. Furthermore, some ARP models assume a homogeneous ambulance fleet, in which patients receive first aid by medical personnel before being transported to the hospital [15]. In contrast, other ARP models consider a heterogeneous fleet of ambulances equipped with special equipment and staffed by trained personnel to transport patients requiring different levels of care [31].

The objective functions of most existing ARP models aim to improve service quality for patients. Common objective functions include minimizing routing costs [32], average response time [23, 30], average waiting time [8, 26], and the latest service completion time [29]. Several mathematical optimization models have been proposed to solve the ARP to optimality. However, these approaches often suffer from scalability limitations due to the inherent complexity of the ARP, making them unsuitable for efficiently solving large-scale instances. Consequently, heuristic and metaheuristic approaches have been proposed to efficiently obtain near-optimal, and in some cases optimal, solutions for large-scale ARP instances.

Talarico et al. [29] proposed two mathematical formulations that generate ambulance route plans by minimizing the latest service completion time among patients awaiting assistance. Krauss [18] proposed a branch-and-price algorithm for a static ambulance routing problem in an urban EMS environment, with the objective of minimizing weighted patient waiting times while accounting for ambulance diversion and reassignment. Rabbani et al. [25] investigated an ambulance routing model with multiple patients and a limited ambulance fleet. They proposed a MILP formulation to minimize the latest service completion time while prioritizing severely injured patients (i.e., red-code patients). In addition, they developed two metaheuristic solution approaches (i.e., genetic algorithm and particle swarm optimization) to generate solutions efficiently.

Tlili et al. [32] proposed a genetic algorithm to determine the optimal ambulance routes while minimizing operational costs. Similarly, a particle swarm optimization algorithm was proposed to minimize the total response time and the

patient waiting time [33]. Two TS algorithms were proposed in [34] to address the ARP with time windows and were compared with a mathematical model, demonstrating their computational efficiency. Aringhieri et al. [3] investigated the problem of determining optimal patient transportation routes while incorporating fairness criteria into the optimization process. They proposed a hybrid algorithm that combines machine learning with neighborhood search techniques to efficiently solve large-scale instances.

3 Ambulance Routing Problem with Patient Triage Priority

3.1 Problem Description

The objective of the proposed ARP is to determine the optimal routes for a fleet of d ambulances to provide emergency assistance to a set of n patients. Each ambulance transports at most one patient at a time to a hospital. Priority is given to red-code patients, who require immediate transportation to a hospital, over green-code patients, whose conditions are less urgent and can tolerate longer waiting times. The notation used for the ARP with patient triage priority is summarized in Table 1. Let K denote the set of d available ambulances that are ready to respond to emergency incidents. Let I_H be the set of h hospitals to which patients may be transported. Let $I_C = I_R \cup I_G$ denote the set of n patients requiring transportation to a hospital, partitioned into the set of red-code patients, I_R and the set of green-code patients, I_G . Let I_S denote the set of starting locations for each ambulance $k \in K$. Furthermore, $I = I_C \cup I_H \cup I_S$ denotes the set of all nodes, and let $A = \{(i, j) \mid i \in I_S, j \in I_C\} \cup \{(i, j) \mid i \in I_C, j \in I_H\}$ denote the set of feasible directed arcs. The travel time required for an ambulance to travel from node i to node j is denoted by $t_{ij} \geq 0$. Each patient $i \in I_C$ is associated with a service time S_i .¹ We assume that: (1) the capacity of hospitals is enough to accommodate all patients; (2) each ambulance transports one patient at a time; and (3) the ambulance must finish its route at a hospital.

3.2 Mathematical Formulation

Using the notation presented in Table 1, the MILP model for the ARP with patient triage priority is formulated as follows. Specifically, the model uses a binary decision variable

¹ For simplicity, in our experiments we assume that $S_i = 0$. In practice, the call center can assign a pre-defined estimated service time to each incident, according to the type of incident, by looking at available tables of estimated service times per incident type that should be prepared by analyzing historical data.

Table 1 Notation used to model the ARP with patient triage priority

Sets and indices:	
$K = \{1, 2, \dots, d\}$	Set of d ambulances.
$I_R = \{1, 2, \dots, r\}$	Set of r red-code patients.
$I_G = \{r + 1, r + 2, \dots, n\}$	Set of $n - r$ green-code patients.
$I_C = \{1, 2, \dots, n\}$	Set of n patients ($I_C = I_R \cup I_G$).
$I_H = \{n + 1, n + 2, \dots, n + h\}$	Set of h hospitals.
$I_S = \{n + h + 1, n + h + 2, \dots, n + h + d\}$	Set of d ambulance starting positions (one per ambulance).
$I = I_C \cup I_H \cup I_S$	Set of all nodes.
$A = \{(i, j) \mid i \in I_S, j \in I_C\} \cup \{(i, j) \mid i \in I_C, j \in I_H\}$	Set of feasible directed arcs.
Parameters:	
$p_1^R, p_2^R, p_3^R > 0$	Priority weights for red-code patients.
$p_1^G, p_2^G, p_3^G > 0$	Priority weights for green-code patients.
t_{ij}	Travel time from node i to node j .
S_i	Service time of patient i .
$\omega_1, \omega_2, \omega_3, \omega_4 > 0$	Weights for the four components of the multi-criteria function.
Decision variables:	
$x_{ijk} \in \{0, 1\}$	Binary variable equal to 1 if ambulance k travels directly from node i to node j , and 0 otherwise.
$w_{ik} \in [0, 0.1M]$	Arrival time of ambulance k at node i .
$y_{ik} \in [0, \infty)$	The number of times that ambulance k enters node i , for $i \in I, k \in K$. Defined as a continuous variable, but it will take binary values at any feasible solution. Therefore, y_{ik} equals 1 if ambulance k enters node i , and 0 otherwise.
$z_{ik} \in [0, \infty)$	The number of times that ambulance k exits node i , for $i \in I, k \in K$. Defined as a continuous variable, but it will take binary values at any feasible solution. Therefore, z_{ik} equals 1 if ambulance k exits node i , and 0 otherwise.
$Y_i \in [0, \infty)$	The total number of ambulances that enter node i , for $i \in I$. Defined as a continuous variable, it will take integer values at any feasible solution.
$Z_i \in [0, \infty)$	The total number of ambulances that exit node i , for $i \in I$. Defined as a continuous variable, it will take integer values at any feasible solution.

x_{ijk} , which is equal to the value of 1 if ambulance k travels directly from node i to node j and the value of 0 otherwise. It also employs a continuous variable w_{ik} to represent the arrival time of ambulance k at node i . In addition, the model employs the auxiliary variables y_{ik} and z_{ik} , which are equal to 1 if ambulance k enters and exits node i , respectively, and 0 otherwise. Finally, let Y_i denote the total number of ambulances entering node i (i.e., $Y_i = \sum_{k \in K} y_{ik}$). Similarly, let $Z_i = \sum_{k \in K} z_{ik}$ denote the total number of ambulances leaving node i . Note that for patient nodes $i \in I_C$, the variables Y_i and Z_i can only take the values 0 or 1 in any feasible solution (i.e., each patient is visited by at most one ambulance), whereas for hospital nodes $i \in I_H$, Y_i can be as large as d (more than one ambulance can go to the same hospital). The variables Y_i , Z_i , y_{ik} , and z_{ik} are defined as sums of the binary variable x_{ijk} . This is because these expressions take integer values whenever the underlying binary variables are

integer-valued (i.e., whenever a feasible solution is obtained), and therefore they may be declared as continuous variables without changing the feasible set or the optimal solution. This relaxation removes unnecessary integrality constraints, thereby improving the performance of the solver.

The optimization problem is formulated as a maximization of a weighted sum of four components, f_1 , f_2 , f_3 , and f_4 , each reflecting a particular aspect of performance. The first component f_1 represents the weighted number of patients assigned to ambulances, defined as follows:

$$f_1 = \sum_{i \in I_R} p_1^R Y_i + \sum_{i \in I_G} p_1^G Y_i \quad (1)$$

where p_1^R and p_1^G are the reward weights of assigning red-code and green-code patients, respectively, contributing to f_1 . By setting $p_1^R \gg p_1^G$ the model strongly prioritizes

red-code patients over green-code patients. The second component f_2 represents the negative weighted travel time from ambulance starting locations to patients, defined as follows:

$$f_2 = -p_2^R \sum_{k \in K} \sum_{\substack{i \in I_S: \\ (i,j) \in A}} \sum_{j \in I_R} t_{ij} x_{ijk} - p_2^G \sum_{k \in K} \sum_{\substack{i \in I_S: \\ (i,j) \in A}} \sum_{j \in I_G} t_{ij} x_{ijk} \quad (2)$$

where p_2^R and p_2^G are the penalty weights associated with travel times when traveling from an ambulance starting location to a red-code and green-code patient, respectively, contributing to f_2 . By setting $p_2^R \gg p_2^G$ the model emphasizes responding more quickly to red-code patients. The third component f_3 represents the negative weighted travel time from patients to hospitals, defined as follows:

$$f_3 = -p_3^R \sum_{k \in K} \sum_{\substack{i \in I_R: \\ (i,j) \in A}} \sum_{j \in I_H} t_{ij} x_{ijk} - p_3^G \sum_{k \in K} \sum_{\substack{i \in I_G: \\ (i,j) \in A}} \sum_{j \in I_H} t_{ij} x_{ijk} \quad (3)$$

where p_3^R and p_3^G are the penalty weights associated with the travel time from red-code and green-code patient, respectively, to a hospital, contributing to f_3 . By setting $p_3^R \gg p_3^G$ the model emphasizes transporting red-code patients to hospitals faster. Note that the combination of f_2 and f_3 minimizes the total ambulance travel time (that is, if we set $p_2^R = p_2^G = p_3^R = p_3^G > 0$ and $\omega_2 = \omega_3 > 0$). Finally, the fourth component f_4 represents the negative sum of arrival times at hospitals, defined as follows:

$$f_4 = - \sum_{k \in K} \sum_{i \in I_H} w_{ik} \quad (4)$$

which encourages ambulances to complete their assigned routes as early as possible. This component also takes into account the estimated service times S_i at patients, which is not taken into account when minimizing travel times.

For each objective component f_1, f_2, f_3 , and f_4 is associated with a weight $\omega_1, \omega_2, \omega_3$, and ω_4 , respectively, to construct a single aggregate objective. By choosing $\omega_1 \gg \omega_2 \gg \omega_3 \gg \omega_4$, we impose a lexicographic priority: the first component f_1 (number of patients served, weighted by their priority) is maximized first and foremost; the subsequent terms then act as tie-breakers in descending order of importance. For example, we can choose $\omega_1 = 10^8$, $\omega_2 = 10^2$, $\omega_3 = 1$, $\omega_4 = 10^{-6}$ and $p_1^R = 10$, $p_1^G = 1$, $p_2^R = 5$, $p_2^G = 1$, $p_3^R = 5$, $p_3^G = 1$, which effectively enforces maximizing the number of patients served (especially red-code patients) as the primary goal, then minimizing travel time from start to patient, then minimizing travel time from patient to hospital, and finally minimizing the sum of arrival times at hospitals.

The objective of the proposed MILP is defined as follows:

$$\max \sum_{i=1}^4 \omega_i f_i, \quad (5)$$

subject to:

$$y_{ik} = \sum_{j \in I: (j,i) \in A} x_{ijk}, \quad \forall i \in I, k \in K, \quad (6)$$

$$z_{ik} = \sum_{j \in I: (i,j) \in A} x_{ijk}, \quad \forall i \in I, k \in K, \quad (7)$$

$$Y_i = \sum_{k \in K} y_{ik}, \quad \forall i \in I, \quad (8)$$

$$Z_i = \sum_{k \in K} z_{ik}, \quad \forall i \in I, \quad (9)$$

$$y_{ik} = z_{ik}, \quad \forall i \in I_C, k \in K, \quad (10)$$

$$z_{ik} \leq 0, \quad \forall i \in I_S \setminus \{n+h+k\}, k \in K, \quad (11)$$

$$Z_i \leq 1, \quad \forall i \in I_S \cup I_C, \quad (12)$$

$$Z_i = Y_i, \quad \forall i \in I_C, \quad (13)$$

$$Y_i = 0, \quad \forall i \in I_S, \quad (14)$$

$$Z_i = 0, \quad \forall i \in I_H, \quad (15)$$

$$Y_i \leq d, \quad \forall i \in I_H, \quad (16)$$

$$y_{ik}, z_{ik} \leq 1, \quad \forall i \in I, k \in K, \quad (17)$$

$$w_{ik} + S_i + t_{ij} - w_{jk} \leq (1 - x_{ijk})M, \quad \forall (i,j) \in A, k \in K, \quad (18)$$

$$w_{ik} \leq M y_{ik}, \quad \forall i \in I_C \cup I_H, k \in K, \quad (19)$$

$$w_{ik} = 0, \quad \forall i \in I_S, k \in K, \quad (20)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall (i,j) \in A, k \in K, \quad (21)$$

$$y_{ik}, z_{ik} \in [0, \infty), \quad \forall i \in I, k \in K, \quad (22)$$

$$w_{ik} \in [0, 0.1M], \quad \forall i \in I, k \in K, \quad (23)$$

$$Y_i, Z_i \in [0, \infty), \quad \forall i \in I, \quad (24)$$

where constraints Eqs. 6 and 7 define y_{ik} and z_{ik} in terms of the x variables, as the number of times that ambulance k enters and exits node i , respectively, and in combination with constraint Eq. 17, the following are true: y_{ik} is 1 if and only if ambulance k enters node i along some incoming arc (i.e., k travels from some node j to i), and z_{ik} is 1 if and only if ambulance k exits node i along some outgoing arc. Constraint Eq. 8 provides the number of ambulances entering node i , whereas constraint Eq. 9 provides the number of ambulances exiting node i . Constraint Eq. 10 ensures that the same ambulance enters and exits each patient node, constraint Eq. 11 ensures that only ambulance k is allowed to exit node $n + h + k$, constraint Eq. 12 ensures that no more than one ambulance can depart from any starting node or patient node, and together with constraint Eq. 11, they ensure that at most one ambulance can leave the starting node and at most one ambulance can leave any given patient node, preventing in this way two different ambulances from serving the same patient. Constraint Eq. 13 ensures that the number of ambulances entering and exiting each patient node is the same, and together with constraint Eq. 12, they preserve the flow conservation at each patient node: if an ambulance enters a patient node i (meaning $Y_i = 1$), it must also exit that node ($Z_i = 1$), and vice versa (if a patient is not visited, nothing enters or exits that node). Constraints Eqs. 12 and 13 imply that $Y_i \leq 1$ for all patient nodes, which guarantees that each patient node is entered by at most one ambulance (no patient is served twice). Constraints Eqs. 14 and 15 prevent any ambulance from entering a start node or exiting a hospital node, respectively. Constraint Eq. 16 allows multiple ambulances to enter the same hospital node. Constraint Eq. 17 ensures that each ambulance may visit (enter and exit) each node at most once. Constraint Eq. 18 represents time-continuity as follows: if ambulance k travels from node i to node j (i.e., $x_{ijk} = 1$), then it ensures that the arrival time w_{jk} at node j is at least $w_{ik} + S_i + t_{ij}$. Here, S_i is the service time at node i (which is nonzero if i is a patient node, representing the on-scene time to stabilize and load the patient), and t_{ij} is the travel time from i to j . Note that the use of the M term renders the constraint inactive (satisfied trivially) whenever $x_{ijk} = 0$. Constraint Eq. 19 ensures that the arrival time w_{ik} is set to zero whenever ambulance k does not actually enter node i . Constraint Eq. 20 initializes all ambulances' start times: it sets $w_{ik} = 0$ at time zero for each ambulance k and for all starting locations $i \in I_S$. Finally, constraints Eqs. 21–24 specify the domains of the decision variables.

There are some obvious variations of the above formulation, which may be preferred by different practitioners,

according to different policies. From this point onward, we will refer to *MILP-1* as the formulation defined by Eqs. 5–24, with Eq. 5 as the objective function, corresponding to the weighted sum of the four component objectives f_1 – f_4 introduced earlier by Eqs. 1–4.

To express the weighted multi-criteria values in Eq. 5 in terms of total travel time values (that are more meaningful to practitioners or decision-makers), a simpler objective function that is a linear combination of the total travel time and total hospital arrival time is defined as follows:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} t_{ij} x_{ijk} + \epsilon \sum_{k \in K} \sum_{i \in I_H} w_{ik} \quad (25)$$

subject to the constraints defined in Eqs. 6–24, and two additional constraints:

$$\sum_{i \in I_R} Y_i \geq \min\{r, d\} \quad (26)$$

$$\sum_{i \in I_C} Y_i \geq \min\{n, d\} \quad (27)$$

where the first term (i.e., $\sum_{k \in K} \sum_{(i,j) \in A} t_{ij} x_{ijk}$) of the objective function in Eq. 25 is the total travel time of all ambulances (from start to patient to hospital for each used ambulance), whereas the second term (i.e., $\sum_{k \in K} \sum_{i \in I_H} w_{ik}$) is the sum of all arrival times at hospitals (i.e., the times at which each ambulance finishes its route at a hospital), ϵ is a small positive constant (i.e., $\epsilon = 10^{-6}$) ensuring that the minimization of the total travel time is the dominant objective, and minimizing the sum of arrival times acts as a secondary objective. In other words, the lexicographic hierarchy of the multi-criteria objective in Eq. 5 is replicated in Eq. 25 by this weighted sum with a small constant ϵ . Furthermore, the additional constraints Eqs. 26 and 27 ensure, respectively, the priority for all red-code patients and that each available ambulance will serve one patient, whenever sufficient ambulances are present. We will refer to *MILP-2* as the formulation with Eq. 25 as the objective function, subject to constraints Eqs. 6–24 and 26–27.

Furthermore, since minimizing travel time is more meaningful to most decision makers, all the experiments reported in Sect. 5 that use the MILP formulation were performed using *MILP-2*. In the corresponding tables of results, we will report the values of only the first and dominant component objective of Eq. 25, i.e., $\sum_{k \in K} \sum_{(i,j) \in A} t_{ij} x_{ijk}$, which is the total travel time of all ambulances, and refrain from reporting the second component objective, for the following reasons. Firstly,

the first term is the dominant term, and with $\epsilon = 10^{-6}$, even rounding the value of Eq. 25 with both terms to 2 decimal places would match the value of the first term, as given in the tables. Secondly, the second term acts as a tie-breaker in cases of two solutions with identical values in the first term, and is essentially used for technical purposes, so that when combined with constraints Eqs. 18 and 19, the choice of w_{ik} values is such that the idle waiting times at all stages are minimized (to zero).

Finally, a third formulation (*MILP-3*) can be obtained by using

$$\max \sum_{i=2}^4 \omega_i f_i, \quad (28)$$

as the objective function, subject to constraints Eqs. 6–24 and 26–27. This third formulation avoids the use of f_1 , as given by Eq. 1, while still giving different priorities on choosing faster routes for red-code versus green-code patients and for different route segments.

It should be emphasized that both the number of patients n and the number of red-code patients r may be greater than, equal to, or less than the number of available ambulances, d . In other words, all three MILP variations are feasible no matter whether or not $d < r$, and no matter whether or not $d < n$. If $d \geq n$, all patients will be served in the initial optimization planning (that is, by running one of the MILPs once). If $r \leq d < n$, all red-code patients will be served in the first-stage optimization plan (first run of one of the MILPs). However, some green-code patients will remain unserved. In this case, re-optimization is needed to schedule these patients, which can be achieved by rerunning the model, after removing all patients planned to be served in the first stage, while changing ambulances' starting positions to the hospital nodes to which they are scheduled to transfer the patients in the previous stage. If $d < r$, some red-code patients and all green-code patients will remain unserved.

In any case, if $d < n$, some patients will remain unserved after the first stage of optimization. We can then apply multi-stage optimization with $\lceil \frac{n}{d} \rceil$ stages, by removing patients served in previous stages and adjusting ambulance starting positions, until all patients are assigned, as discussed above.

Note that, in all of our experiments reported in Sect. 5, for simplicity, we present the results of only the first stage of optimization, corresponding to one run of the *MILP-2* (and similarly for the metaheuristic). If $d \geq n$, this optimizes the whole system, whereas if $d < n$, results correspond to the

first optimization stage only. Note that, for our purposes, this is enough to compare the performance of the algorithm, since further stages of one instance simply correspond to a separate instance of smaller size.

Although the focus of this paper is on the first stage of optimization, for the sake of completeness, in Sect. 5.5 we present an example which requires three stages of optimization.

4 Hybrid Adaptive Tabu Search Metaheuristic

In an ideal setting, an exact MILP solver, such as the one proposed in Sect. 3.2, identifies the provably optimal assignment of ambulances to patients and the corresponding routing decisions. However, as the problem size increases (e.g., due to a larger number of patients, ambulances, or hospitals), the MILP approach becomes computationally expensive, often requiring excessive computation time to obtain an optimal solution.

To efficiently solve the ARP with patient triage, a hybrid metaheuristic combining TS [12, 14] and the LNS [27, 28], is proposed. In LNS, a solution is partially destroyed using destroy operators and subsequently reconstructed using repair operators. This enables multiple neighborhood structures to be explored within the same search process in an adaptive manner, where the performance of each neighborhood is monitored and its selection probability is dynamically adjusted accordingly. TS is a powerful metaheuristic that guides local search operators toward promising regions of the search space containing high-quality solutions. Starting from an initial solution, TS explores its neighborhood and applies the best move at each iteration. A memory structure, known as the tabu list, is used to prevent the search from cycling back to recently visited solutions [10, 13]. Specifically, recently performed moves are stored in the tabu list, and they remain prohibited for a predefined number of iterations, referred to as the tabu tenure. The tabu status of a move is overridden if the aspiration criterion is satisfied.

The proposed HATS algorithm consists of the following main features: (a) seven problem-specific local search operators; (b) adaptive neighborhood control; (c) reactive control of the tabu tenure; and (d) diversification mechanism based on the Shaw destroy-repair operator [28]. Algorithm 1 outlines the overall HATS algorithm for the ARP.

```

1: Initialize tabu list  $TL$ 
2: Initialize tabu tenure  $tt$ 
3: Let  $Q^*$  be the best-so-far solution
4: Let  $f(Q)$  be the objective value of solution  $Q$ 
5: Let  $i$  be the iteration counter
6: Let  $s$  be the stagnation counter
7:  $Q \leftarrow \text{GenerateInitialSolution}$ 
8:  $Q^* \leftarrow Q$ 
9:  $i \leftarrow 0, s \leftarrow 0$ 
10: while (termination condition not met) do
11:   Compute tabu tenure bounds  $tt_{min}$  and  $tt_{max}$ 
12:    $S \leftarrow \text{generate neighborhood structure}$ 
13:   for each move  $m \in S$  do
14:     if  $m$  not tabu or aspiration criterion is met then
15:       score  $\leftarrow$  update score of  $m$ 
16:       if score  $\geq$  max score then
17:         max score  $\leftarrow$  score,  $Q^m \leftarrow$  Apply  $m$  on  $Q$ 
18:       end if
19:     end if
20:   end for
21:    $Q \leftarrow Q^m$ 
22:   Update  $TL$  with move made by  $m$ 
23:   Adjust the tabu tenure  $tt_{min} \leq tt \leq tt_{max}$ 
24:   if ( $i \bmod D_R == 0$ ) then
25:      $Q \leftarrow \text{ApplyDestroyRepairOperator}$ 
26:   end if
27:   if ( $s \geq S_L$ ) then
28:     random  $\leftarrow \mathcal{U}(0, 1)$ 
29:     if (random  $\geq R_P$ ) then
30:        $Q \leftarrow \text{GenerateRandomSolution}$ 
31:     else
32:        $Q \leftarrow \text{GeneratePerturbedSolution}(Q^*)$ 
33:     end if
34:      $s \leftarrow 0$ 
35:      $TL \leftarrow \emptyset$ 
36:   end if
37:   if ( $f(Q)$  is better  $f(Q^*)$ ) then
38:      $Q^* \leftarrow Q$ 
39:      $s \leftarrow 0$ 
40:   else
41:      $s \leftarrow s + 1$ 
42:   end if
43:    $i \leftarrow i + 1$ 
44: end while
45: OUTPUT:  $Q^*$  %ARP Solution

```

4.1 Objective Function

For the proposed HATS implementation, Eq. 5 corresponding to the objective function of *MILP-1* is adopted using the following parameter values: $\omega_1 = 10^8$, $\omega_2 = \omega_3 = 1$, $\omega_4 = 10^{-6}$, $p_1^R = 10$, and $p_1^G = p_2^R = p_2^G = p_3^R = p_3^G = 1$. This choice of parameter values makes the HATS results directly comparable to those obtained using *MILP-2*, where Eq. 25 serves as the objective with $\epsilon = 10^{-6}$. With the selected parameter values, *MILP-1* and *MILP-2* are expected to produce identical solutions.

4.2 Generation of Initial Solution

Each candidate solution Q for the ARP with patient triage is represented as an ordered collection of d ambulance assignment triples, $Q = (Q_1, Q_2, \dots, Q_d)$, where d denotes the number of available ambulances. Specifically, for each ambulance k , the solution stores a triple $Q_k = (s_k, p_k, h_k)$, where $s_k \in I_S$ is the starting location, $p_k \in I_C \cup \{-1\}$ is the assigned patient, and $h_k \in I_H \cup \{-1\}$ is the assigned hospital. When $p_k = -1$, the ambulance is not assigned to any patient. Consequently, $h_k = -1$ indicating that no hospital is associated with an idle ambulance.

Consider an example with $d = 3$ ambulances: $Q = ((30, 4, 12), (31, 2, 18), (32, -1, -1))$. This solution corresponds to the following assignments: the first ambulance will start from node 30 to pick up patient 4 and transfer the patient to hospital 12, the second ambulance will start from node 31 to pick up patient 2 and transfer the patient to hospital 18, and the third ambulance will remain at its station at node 32 without any assignment.

An initial feasible solution is generated using a greedy construction procedure. The solution is constructed incrementally by assigning one patient until all eligible patients (up to $\min\{n, d\}$ patients) have been assigned to ambulances. Initially, all the red-code patients (i.e., I_R) are assigned to the available ambulances (subject to ambulance availability), followed by the assignment of the green-code patients (i.e., I_G). Each patient is assigned to the closest available ambulance and subsequently to the nearest hospital, thereby minimizing the total travel time.

4.3 Neighborhood Structure

Starting from an initial feasible solution, the proposed HATS algorithm explores neighboring solutions generated using seven problem-specific local search operators. The feasibility of all neighboring solutions generated by the local search operators is strictly maintained. Specifically, a local search move is considered feasible if and only if it satisfies the following constraints: (a) no patient can be assigned to more than one ambulance; (b) each ambulance can carry at most one patient; and (c) each ambulance route must consist of a valid sequence (i.e., from a starting location to the patient and then to a hospital). Each feasible neighboring solution is subsequently evaluated, and the best candidate is selected to replace the current solution.

In every iteration of HATS, the following search operators are applied:

1. The `create_route` operator assigns a patient p_k to an idle ambulance k . For each idle ambulance, this move first considers any unassigned red-code patients; if none are available, it then considers unassigned green-code patients. It selects a patient-hospital pair that minimizes the total travel time, effectively dispatching ambulance k to the closest available patient and sending that patient to the nearest hospital h_k . Applying the operator transforms the triple $(s_k, -1, -1)$ into (s_k, p_k, h_k) , and results in the addition of arcs (s_k, p_k) and (p_k, h_k) .
2. The `remove_route` operator removes an existing route assignment from an ambulance, thereby making that ambulance idle. It does so by selecting an ambulance k , removing its current patient-hospital assignment (s_k, p_k, h_k) , marking patient p_k as unassigned, and allowing ambulance k to be assigned to a different patient. Applying the operator transforms the triple (s_k, p_k, h_k) into $(s_k, -1, -1)$, and results in the deletion of arcs (s_k, p_k) and (p_k, h_k) .
3. The `two_way_patient_swap` operator exchanges the patients assigned between two ambulances. Suppose that ambulance k is assigned to transfer patient p_k to the nearest hospital h_k and ambulance l is assigned to transfer patient p_l to the nearest hospital h_l . After the swap, ambulance k will serve patient p_l and ambulance l will serve patient p_k with both ambulances retaining their original starting location (i.e., s_k and s_l), and their closest hospitals (i.e., h_k and h_l). Applying the operator transforms the two triples (s_k, p_k, h_k) and (s_l, p_l, h_l) into (s_k, p_l, h_l) and (s_l, p_k, h_k) , respectively, and results in the deletion of arcs (s_k, p_k) and (s_l, p_l) ; and in the addition of arcs (s_k, p_l) and (s_l, p_k) .
4. The `three_way_patient_swap` operator selects three assigned patients p_k , p_l , and p_q , and performs a cyclic swap rotation in both directions. When applied in the counter-clockwise direction the rotation transforms the three triples (s_k, p_k, h_k) , (s_l, p_l, h_l) and (s_q, p_q, h_q) into (s_k, p_l, h_l) , (s_l, p_q, h_q) and (s_q, p_k, h_k) , respectively; and results in the deletion of arcs (s_k, p_k) , (s_l, p_l) , and (s_q, p_q) , and in the addition of arcs (s_k, p_l) , (s_l, p_q) , and (s_q, p_k) . Applying the operator in the clockwise direction the rotation transforms the three triples (s_k, p_k, h_k) , (s_l, p_l, h_l) and (s_q, p_q, h_q) into (s_k, p_q, h_q) , (s_l, p_k, h_k) and (s_q, p_l, h_l) , respectively; and results in the deletion of arcs (s_k, p_k) , (s_l, p_l) , and (s_q, p_q) , and in the addition of arcs (s_k, p_q) , (s_l, p_k) , and (s_q, p_l) . Note that the corresponding nearest hospitals for each patient are also swapped during the rotations as in the `two_way_patient_swap` operator. The rotation direction leading to the best result is selected.
5. The `ejection` operator selects two assigned patients p_k and p_l , and one unassigned patient p_q . A cyclic rotation is performed by replacing p_l with p_k and p_q with p_l . The corresponding closest hospitals for each patient are also swapped during the rotation. Applying the operator transforms the three triples (s_k, p_k, h_k) , (s_l, p_l, h_l) and $(s_q, -1, -1)$ into $(s_k, -1, -1)$, (s_l, p_k, h_k) and (s_q, p_l, h_l) , respectively; and results in the deletion of arcs (s_k, p_k) , (s_l, p_l) , and in the addition of arcs (s_l, p_k) and (s_q, p_l) .
6. The `swap_ambulances` operator exchanges both the assigned patients and hospitals between two ambulances. Suppose that ambulance k starting from s_k transfers patient p_k to hospital h_k , and ambulance l starting from s_l transfers patient p_l to hospital h_l . After the swap, the ambulance starting from s_k will be assigned to transfer patient p_l to hospital h_l , and the ambulance starting from s_l will be assigned to transfer patient p_k to hospital h_k . Applying the operator transforms the two triples (s_k, p_k, h_k) and (s_l, p_l, h_l) into (s_l, p_k, h_k) and (s_k, p_l, h_l) ; and results in the deletion of arcs (s_k, p_k) and (s_l, p_l) ; and in the addition of arcs (s_k, p_l) and (s_l, p_k) .
7. The `swap_assigned_with_unassigned` operator is a combination of the `create_route` and `remove_route` operators. Suppose ambulance k is assigned to transfer patient p_k to hospital h_k and that p_l is an unassigned patient. After the swap, the ambulance will remove the assignment of patient p_k (e.g., a green-code patient) and assign patient p_l (e.g., a red-code patient) to be transferred to the closest hospital h_l (i.e., $h_l = \arg \min_{h_l \in I_H} t_{p_l, h_l}$). Applying the operator transforms the triple (s_k, p_k, h_k) into (s_k, p_l, h_l) , and results in the deletion of arcs (s_k, p_k) and (p_k, h_k) ; and in the addition of arcs (s_k, p_l) and (p_l, h_l) .

The selection of neighborhood moves is dynamically weighted according to their historical performance. This is translated as follows: neighboring moves that frequently improve the solution have higher probabilities of being applied than the less effective moves.

4.4 Tabu Search

The tabu list TL used by HATS is maintained at the patient level. For each patient, HATS records the most recent iteration i_{last} at which the patient's assignment has changed (i.e., when an assigned patient moves from one ambulance to another or when its status changes). At each iteration i , a move defined from the neighborhood structure in Sect. 4.3 for a given patient is considered tabu if the following condition holds: $i - i_{last} \leq tt$ where tt denotes the tabu tenure.

Unlike the classic tabu tenure, which is fixed, HATS utilizes a reactive tabu tenure to effectively balance intensification and diversification [7]. A dynamic strategy is employed that adjusts the tabu tenure tt adaptively based on search behavior. The tt value is initialized as follows:

$$tt = \max(4, \lfloor T_F \cdot n \rfloor), \quad (29)$$

where T_F is a constant and n is the total number of patients. The minimum and maximum tabu tenures, tt_{min} and tt_{max} , respectively, are based on the number of currently assigned patients, and they are defined as follows:

$$tt_{min} = \max(2, \lfloor A_P/10 \rfloor), \quad (30)$$

$$tt_{max} = \max(8, \lfloor A_P/3 \rfloor), \quad (31)$$

where A_P is the number of assigned patients. When improvements occur frequently, the tenure gradually decreases toward tt_{min} to promote search intensification. Conversely, frequent repetitions of tabu moves increase the tenure toward tt_{max} , thereby encouraging diversification.

A standard aspiration criterion is applied to the TL as follows: a tabu move is adopted if and only if the best-so-far solution is improved.

4.5 Large Neighborhood Search

The Shaw destroy-repair operator [28] removes and reinserts a cluster of patients close to each other based on travel time. The procedure begins by randomly selecting one ambulance assignment, i.e., Q_k . Subsequently, a set of S_B ambulance assignments is generated. The patients in the assignments are geographically close, in terms of travel time, to the patient in the selected assignment Q_k , which is also include in the set.

The selected patients in the selected ambulance assignments are removed, resulting in a partially destroyed solution with unassigned patients. The repair procedure aims to reassign each patient to an ambulance only with moves that yield better solution quality. This destroy-repair operator serves as the LNS move, promoting diversification while maintaining solution feasibility. The Shaw destroy-repair operator is applied every D_R iterations.

4.6 Diversification Mechanism

To further encourage exploration, the algorithm incorporates occasional restarts whenever stagnation is detected. If no improvement is observed for S_L consecutive iterations,

a restart is triggered. During each restart, one of two diversification strategies is applied probabilistically. With probability $1 - R_P$ the best-so-far solution is partially perturbed by swapping assignments. Otherwise, with probability R_P , a completely random solution is generated from scratch, thereby restarting the search.

5 Experiments

5.1 Experimental Setup

The experiments were conducted on a system equipped with an Intel i5-1035G1 processor (1.00–1.19 GHz), 36 GB RAM, and a 64-bit Windows 10 operating system. The *MILP-2* formulation presented in Sect. 3.2 was solved using the non-commercial PuLP solver [21] (with Eq. 25 as the objective and $\epsilon = 10^{-6}$). The HATS termination condition was set to 500 iterations or when the best-so-far solution is not improved for 60 consecutive iterations in two cycles of iterations. The remaining parameters were set as follows: the tabu tenure factor in Eq. 29 is set to $T_F = 0.1$, the Shaw destroy-repair operator is applied every $D_R = 20$ iterations with a base set to $S_B = 8$, stagnation limit is set to $S_L = 60$, and the random restart probability is set to $R_P = 0.65$. Since HATS includes stochastic components, 30 independent runs were performed using different random seeds.

5.2 Generation of Benchmark Instances

To evaluate the performance of the proposed MILP and HATS algorithms, a set of benchmark instances was generated with varying numbers of red-code patients, green-code patients, hospitals, and ambulances. Each ARP instance was generated by randomly sampling the locations of hospitals, ambulance starting locations, and patient locations from a uniform distribution $\mathcal{U}^2(0, u_{bound})$, where u_{bound} defines the size of the area. The travel time t_{ij} corresponds to the Euclidean distance between nodes i and j . The number of patients, ambulances, and hospitals were systematically varied to investigate their effect on solver performance, resulting in 39 benchmark instances. The characteristics for the generated ARP instances are summarized in Table 2. All benchmark instances are publicly available in the following repository: https://github.com/Mavrovouniotis/arp_instances.

5.3 Comparative Analysis of MILP and HATS

Table 3 summarizes the optimal solutions obtained by *MILP-2* together with the mean (averaged over 30 runs) and

Table 2 Description of the generated benchmark instances

instance	n	r	$(n - r)$	h	d	u_{bound}
1	20	7	13	3	8	30
2	20	7	13	5	10	30
3	20	12	8	6	15	30
4	20	13	7	10	25	30
5	20	20	0	8	25	30
6	23	6	17	3	11	20
7	30	10	20	3	12	40
8	30	10	20	7	20	40
9	30	15	15	10	23	50
10	30	20	10	15	40	50
11	34	14	20	20	50	20
12	37	8	29	4	17	30
13	39	11	28	5	42	30
14	47	29	18	15	56	40
15	50	10	40	7	30	50
16	50	27	23	7	30	50
17	50	27	23	7	70	50
18	50	27	23	30	70	50
19	56	13	43	12	43	40
20	64	38	26	9	80	60
21	77	42	35	23	65	60
22	89	24	65	36	40	70
23	70	30	40	15	40	50
24	70	40	30	30	90	100
25	100	20	80	10	40	100
26	100	30	70	17	45	100
27	100	30	70	17	80	100
28	100	30	70	17	120	100
29	100	30	70	40	80	100
30	100	70	30	40	80	100
31	100	30	70	40	120	100
32	100	70	30	80	80	100
33	100	70	30	80	120	100
34	104	36	68	27	54	70
35	116	38	78	45	130	100
36	150	50	100	30	100	100
37	150	50	100	30	60	100
38	152	36	116	17	48	150
39	250	43	207	32	57	160

best solutions obtained by HATS. The CPU time required to obtain the optimum (or near-optimal) solution is reported in seconds. For HATS, the percentage deviation (%gap) between the best solution obtained over the 30 independent runs and the optimal solution is also reported. Based on the experimental results presented in Table 3, the following observations can be made.

First, MILP finds optimal solutions for all small-scale ARP instances (i.e., instances 1–13) in less than one minute. Although MILP obtains optimal solutions for all benchmark instances, its CPU time increases significantly as the number of patients and ambulances increases. For most large-scale ARP instances (i.e., instances 20–39),

MILP requires several minutes to obtain the optimal solution. Figures 2, 3, and 4 illustrate the optimal solutions for ARP instances 4 (small-scale), 10 (medium-scale) and 17 (large-scale), respectively.

Second, HATS identifies the optimal solutions for 20 out of the 39 benchmark instances within only a few seconds. For the remaining benchmark instances, HATS consistently produces near-optimal solutions, typically with an optimality gap below 0.5%. The CPU time of HATS remains consistently low, even for large-scale instances. For example, on the largest ARP instance (i.e., 39), HATS finds a solution with a %gap of 1.58 within 35.72 seconds, whereas MILP requires 19,333.05 seconds.

Table 3 Results and comparisons of MILP against HATS. The asterisk * indicates that MILP solved the problem to optimality and bold values indicate that HATS found the optimal solution at least once out of the 30 runs. The time for both MILP and HATS is given in seconds

instance	MILP		HATS			
	best	time	mean	best	%gap	time
1	102.35*	2.28	102.35±0.00	102.35	0.0%	0.01
2	105.45*	8.27	105.53±0.44	105.45	0.0%	0.49
3	143.52*	5.48	143.52±0.00	143.52	0.0%	0.62
4	163.22*	10.30	163.31±0.06	163.22	0.0%	28.94
5	181.83*	8.54	182.37±1.00	181.83	0.0%	13.46
6	91.82*	6.96	91.96±0.52	91.82	0.0%	0.66
7	197.30*	10.15	197.45±0.68	197.30	0.0%	0.84
8	249.46*	11.01	251.22±1.46	249.46	0.0%	2.98
9	304.69*	16.66	308.62±4.05	304.69	0.0%	3.73
10	386.26*	28.26	386.31±0.16	386.26	0.0%	21.01
11	145.41*	36.74	146.52±0.71	145.41	0.0%	51.73
12	158.13*	9.10	158.86±0.92	158.13	0.0%	2.00
13	412.89*	22.04	412.94±0.00	412.94	0.01%	0.38
14	450.24*	261.64	452.21±2.34	450.24	0.0%	80.81
15	382.36*	20.56	389.29±4.99	383.43	0.27%	6.18
16	440.65*	78.56	442.20±2.94	440.65	0.0%	3.70
17	651.24*	54.23	651.24±0.00	651.24	0.0%	117.98
18	505.81*	171.38	506.03±0.71	505.81	0.0%	47.51
19	411.94*	50.51	419.34±2.84	413.26	0.32%	10.43
20	992.78*	93.36	994.27±1.16	992.78	0.0%	101.49
21	790.83*	172.70	808.66±8.09	791.68	0.10%	48.61
22	436.58*	154.70	462.14±9.71	447.77	2.56%	16.15
23	476.42*	100.04	484.51±5.25	476.42	0.0%	13.56
24	1404.30*	301.65	1406.20±2.50	1404.30	0.0%	107.24
25	851.93*	80.41	869.33±7.93	852.05	0.01%	9.54
26	997.90*	82.26	1035.86±19.33	1007.60	0.97%	14.12
27	1623.41*	297.36	1676.41±18.37	1643.17	1.21%	50.62
28	1946.12*	435.80	1953.29±3.36	1946.12	0.0%	120.48
29	1302.87*	458.39	1348.47±15.68	1310.23	0.56%	47.34
30	1454.67*	734.53	1477.13±10.16	1466.54	0.81%	53.35
31	1386.25*	803.12	1393.15±7.19	1386.65	0.02%	120.69
32	1128.87*	1897.87	1174.11±23.89	1138.36	0.84%	66.58
33	1365.73*	1769.19	1378.32±6.87	1368.20	0.18%	120.70
34	726.20*	267.08	750.92±13.00	729.68	0.47%	46.15
35	1710.68*	1718.77	1720.20±2.99	1714.76	0.23%	120.71
36	1707.68*	833.25	1766.27±16.93	1730.92	1.36%	74.34
37	1206.43*	361.37	1240.52±16.76	1213.48	0.58%	32.13
38	1463.09*	317.24	1540.62±37.83	1479.60	1.12%	17.38
39	1570.42*	19333.05	1646.10±26.46	1595.31	1.58%	35.72

These results have important implications for intelligent transportation systems. In real-world EMS operations, rapid decision-making is critical, and the ability of HATS to consistently produce near-optimal solutions within seconds makes it well suited for (near) real-time deployment. In contrast, the MILP model, while valuable for benchmarking,

is less suitable for large-scale applications due to its computational requirements. Nevertheless, combining the MILP and HATS approaches in real-world EMS operations offers both decision-support accuracy and operational scalability, which are essential requirements for modern intelligent transportation systems.

Fig. 2 Illustration of the optimal solution generated by the MILP approach on the ARP problem instance 4

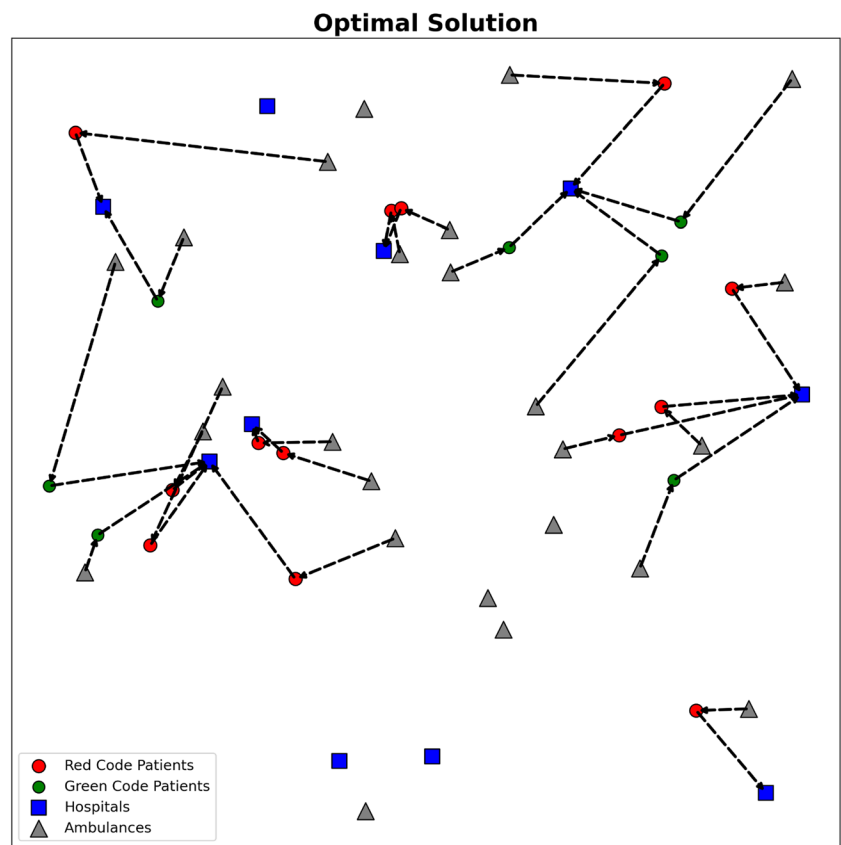


Fig. 3 Illustration of the optimal solution generated by the MILP approach on the ARP problem instance 10

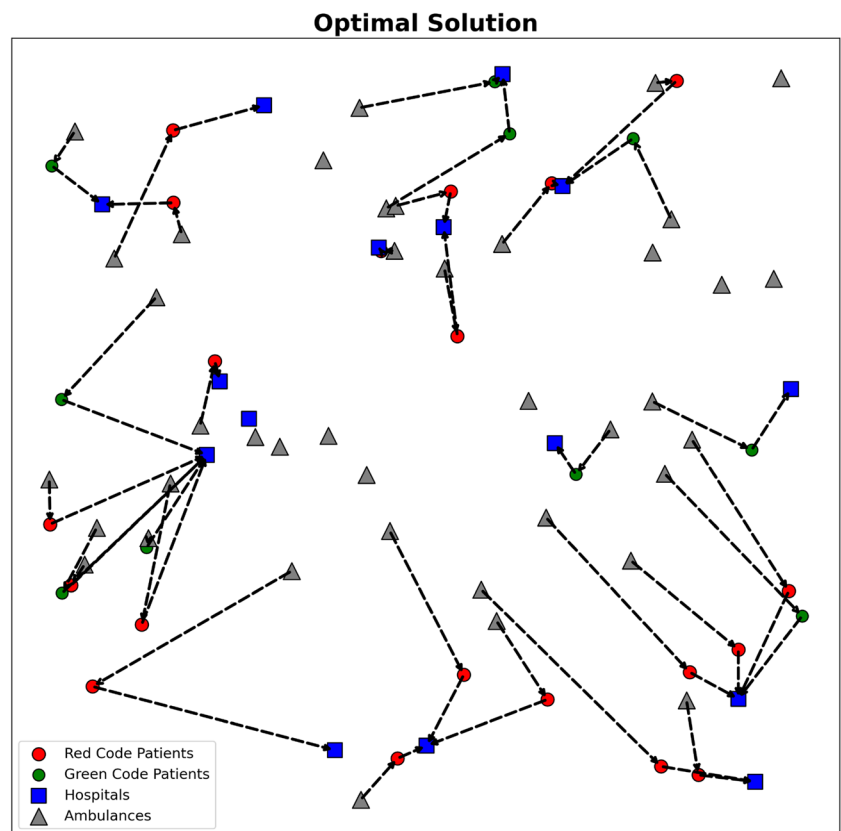
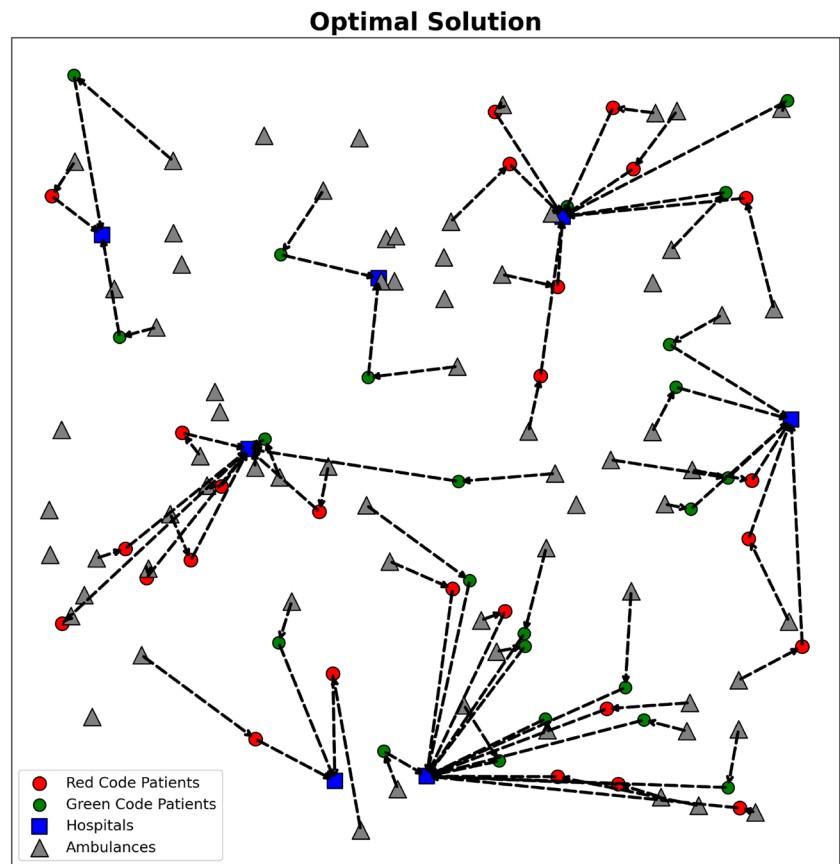


Fig. 4 Illustration of the optimal solution generated by the MILP approach on the ARP problem instance 17



5.4 Sensitivity Analysis

This section investigates how varying the three main characteristics (i.e., number of patients, number of hospitals, and number of ambulances) of the generated instances affect the solution quality, in terms of total travel time. ARP instance 5 from Table 2 is used as the baseline case, while each characteristic is varied systematically. The MILP solver was applied to the generated ARP instances with the modified feature values.

5.4.1 Impact of Red-Code Patients

The number of red-code patients is varied over the values $r \in \{0, 5, 10, 15, 18, 20\}$, while the number of green-code patients ($n - r$) changes accordingly. All generated ARP problem instances consist of the same patient locations, 8 hospitals and 25 ambulance starting locations. The effect of the number of red-code patients on the total travel time is plotted in Fig. 5. The results show that the number of red-code patients does not affect the total travel time. This is because the number of ambulances is sufficient to serve all the patients (both red- and green-code), and since patient locations remain unchanged, the total travel time is not affected by the patient type. In case the number of ambulances is

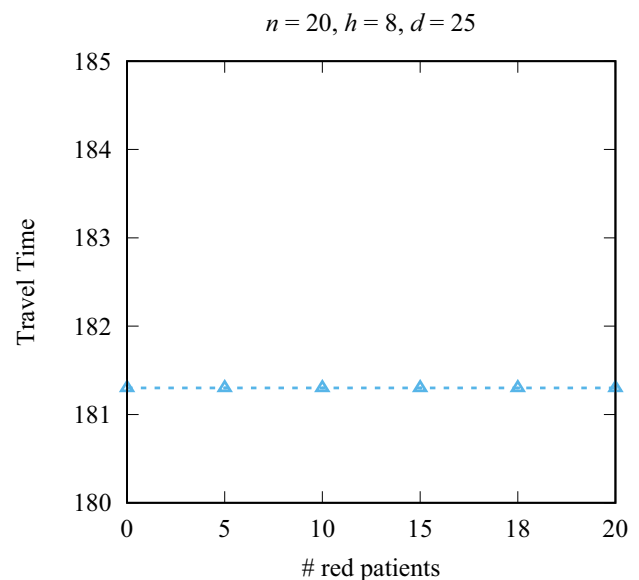


Fig. 5 Objective function against varying number of red-code patients

smaller than the number of patients, then the total travel time will be affected since some ambulances will perform more than one route (i.e., the multi-stage optimization scenario demonstrated in Sect. 5.5). The effect of increasing the number of ambulances is discussed in the next subsection.

5.4.2 Impact of Ambulance Number

The number of ambulances is varied over the values $d \in \{25, 30, 35, 40\}$ on the same set of patients and hospitals. The generated ARP benchmark instances consist of the same 20 patients (10 red-code and 10 green-code) and 8 hospitals. The effect of the number of ambulances available on the total travel time is plotted in Fig. 6. From Fig. 6 it can be observed that as the number of ambulances increases, the total travel time decreases. This behavior is expected because there are more available ambulances to transfer patients to hospitals.

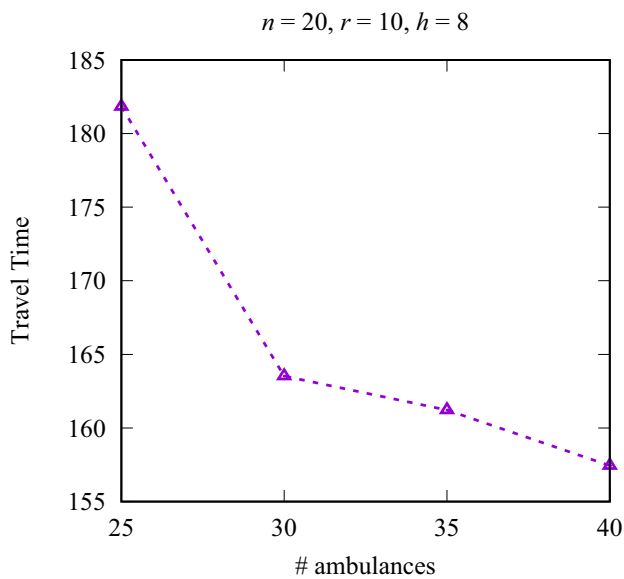


Fig. 6 Objective function against varying number of ambulances

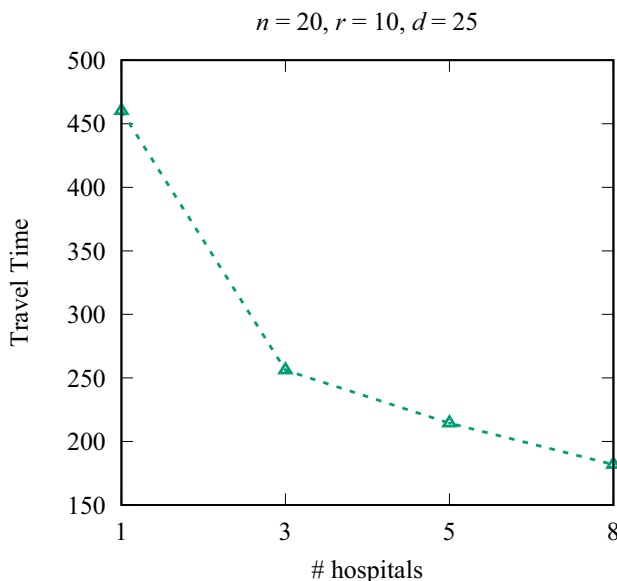


Fig. 7 Objective function against varying number of hospitals

5.4.3 Impact of Hospital Number

The number of hospitals is varied over the values $h \in \{1, 3, 5, 8\}$ on the same set of patients and ambulance starting locations. The generated ARP problem instances consist of the same 20 patients (10 red-code and 10 green-code) and 25 ambulances. The effect of the number of hospitals on the total travel time is plotted in Fig. 7. From Fig. 7 it can be observed that as the number of hospitals increases, the total travel time decreases. This behavior is expected because the ambulances have more hospitals to which they can transport patients.

5.5 Multi-Stage Optimization

As discussed earlier, if the number of patients exceeds the number of available ambulances (i.e., if $d < n$), some patients will not be assigned an ambulance during the first stage of optimization. In this case, we can apply multi-stage optimization with $\lceil \frac{n}{d} \rceil$ stages, where at each stage we remove the patients scheduled to be transported to hospitals in previous stages, adjust the starting positions of ambulances to their latest locations (which typically will be the hospitals where they will transfer their last patients in the previous stage), solve MILP again, and repeat until all patients are assigned an ambulance.

To illustrate this scenario, we have selected ARP instance 25, which involves 100 patients to be served by a fleet of 40 available ambulances, and applied the MILP successively three times, as described above. The optimal solutions for all three optimization stages are shown in Figs. 8, 9, and 10. Specifically, the first 40 patients (including all 20 red-code patients) are scheduled at the first stage, as shown in Fig. 8, with objective 851.93. The next 40 green-code patients are scheduled at the second stage (Fig. 9), with objective 1320.82. The remaining 20 green-code patients are scheduled at the third stage (Fig. 10), with objective 1002.03. The corresponding total travel time for all three stages is 3174.78.

5.6 Dynamic Optimization

So far the optimization model assumes that all incidents are known in advance, i.e., a static optimization problem. In practice, emergency incidents arise dynamically, and their severity is determined when they are reported. Accordingly, the dynamic problem is modeled as a sequence of static problem instances $P(1), P(2), \dots, P(t)$ at time period t [22].

Each static instance $P(t)$ is generated with all the unserved patients (either green-code or red-code) including new patients and the unserved patients from the previous static instance $P(t-1)$. In practice, whenever a new patient appears, a new problem instance is generated. In addition,

Fig. 8 Multi-stage optimization example: Illustration of the optimal solution generated by MILP on ARP instance 25 (stage 1 of 3)

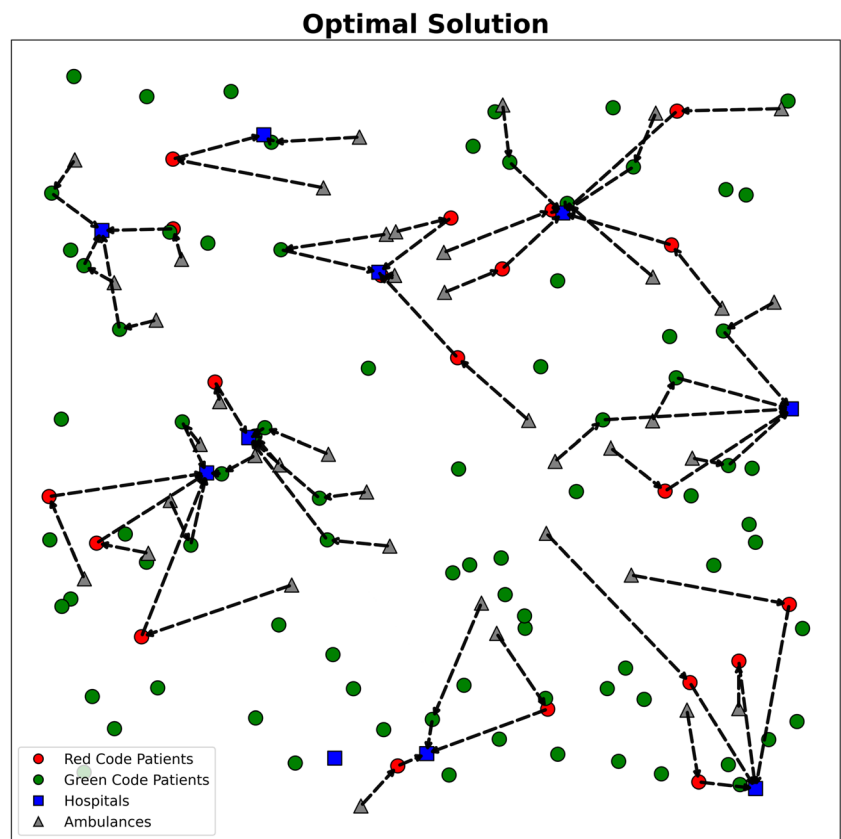


Fig. 9 Multi-stage optimization example: Illustration of the optimal solution generated by MILP on ARP instance 25 (stage 2 of 3)

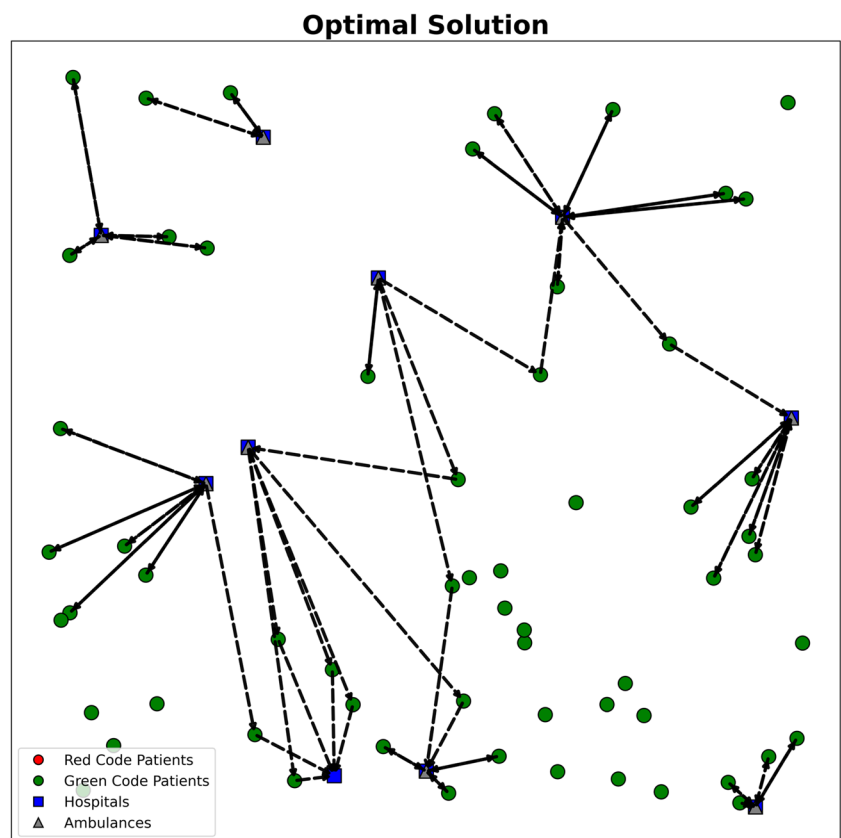
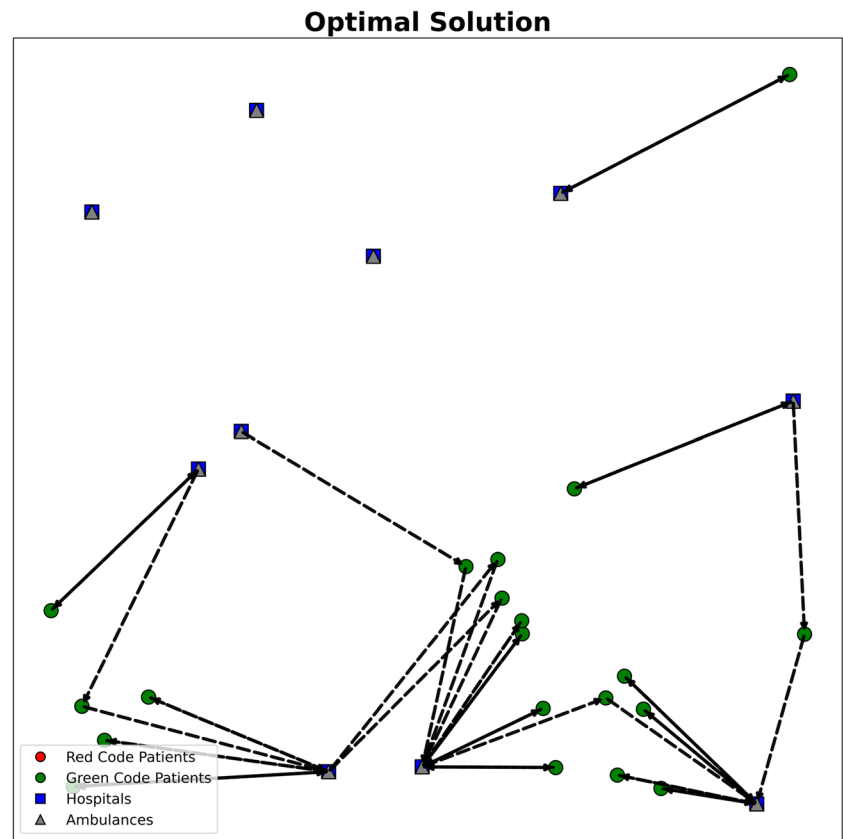


Fig. 10 Multi-stage optimization example: Illustration of the optimal solution generated by MILP on ARP instance 25 (stage 3 of 3)



the available ambulances at time t are included in the problem instance. Note that, even if the ambulance is assigned to a patient but has not yet arrived at the incident it will be considered as an available ambulance (with a new starting position). This allows an ambulance to be assigned to a green-code patient, but a new red-code patient may appear while the ambulance is traveling to serve the green-code patient. Hence the new optimal route will consider the reassignment of the ambulance from the green-code patient to the red-code patient because of the patient triage priority constraint. Finally, the hospital locations are also included in the generation of the problem instance (i.e., to complete the I set in Table 1).

Specifically, the goal is to obtain new optimal routes and patient assignments as new patient incidents occur, while satisfying all the constraints of the described ARP problem. To illustrate a dynamic scenario, we have selected a scenario with 10 patients (2 green-code and 8 red-code), 3 hospitals and 10 ambulances. Suppose that at time t all 10 patients of the problem instance are assigned to an ambulance as presented in Fig. 11. If at time $t + 1$ a new red-code patient appears, i.e., patient 11 as shown in Fig. 12, then a new problem instance is optimized as follows. The assigned ambulance of green-code patient 9 is assigned to red-code patient 11 demonstrating the patient triage priority. Furthermore, if at time $t + 2$ another red-code patient appears, i.e., patient 12 as shown in Fig. 13, then a new problem is optimized again as follows. The

assigned ambulance of the green-code patient 10 is assigned to red-code patient 5 and the assigned ambulance of red-code patient 5 is reassigned to the new red-code patient 12, who is geographically closer to the reassigned ambulance than patient 5 is, thereby minimizing the total response time at $t + 2$.

6 Conclusion

This paper investigates the ARP with patient triage priority. We propose a MILP model that solves the ARP to optimality as well as a metaheuristic approach, namely HATS. Although the MILP guarantees optimality, it becomes intractable for large-scale instances and is therefore impractical for real-world ARP settings. On the other hand, the proposed HATS metaheuristic efficiently explores the solution space with tailored search operators, and avoids getting trapped in local optima by using occasional diversification strategies. The proposed framework integrates strict triage-based decision rules with joint assignment and routing optimization. The experimental results demonstrate that HATS consistently produces solutions with an optimality gap of only a few percentage points, even when the MILP solver requires several minutes to reach the optimum. Furthermore, HATS runs in seconds for problem sizes likely to be encountered in practice (e.g., multiple simultaneous calls).

Fig. 11 Dynamic optimization example: Illustration of the optimal solution generated by MILP at time t

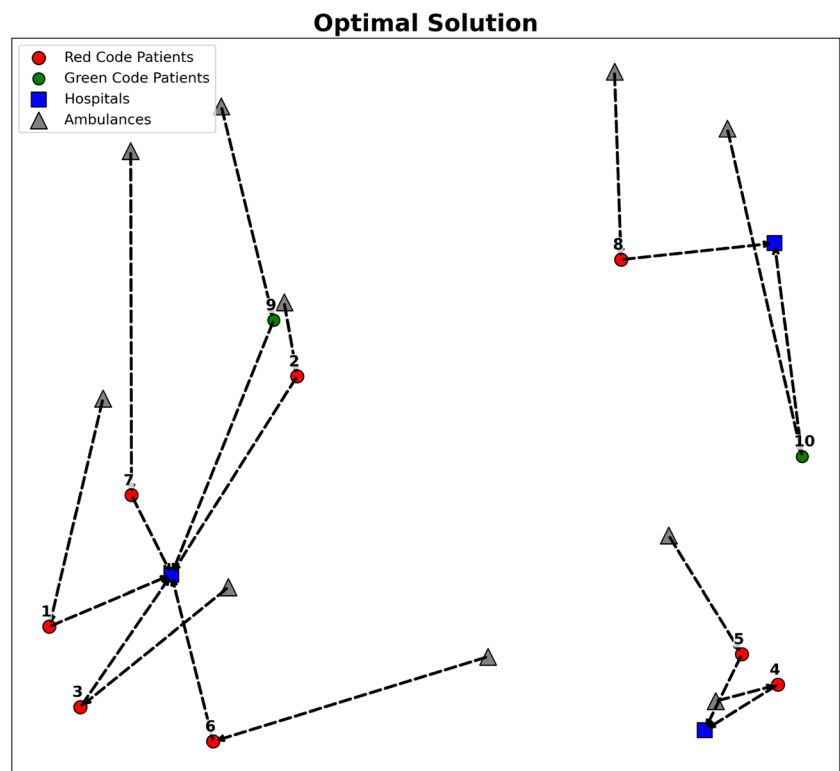


Fig. 12 Dynamic optimization example: Illustration of the optimal solution generated by MILP at time $t + 1$ with the arrival of a new red-code patient 11

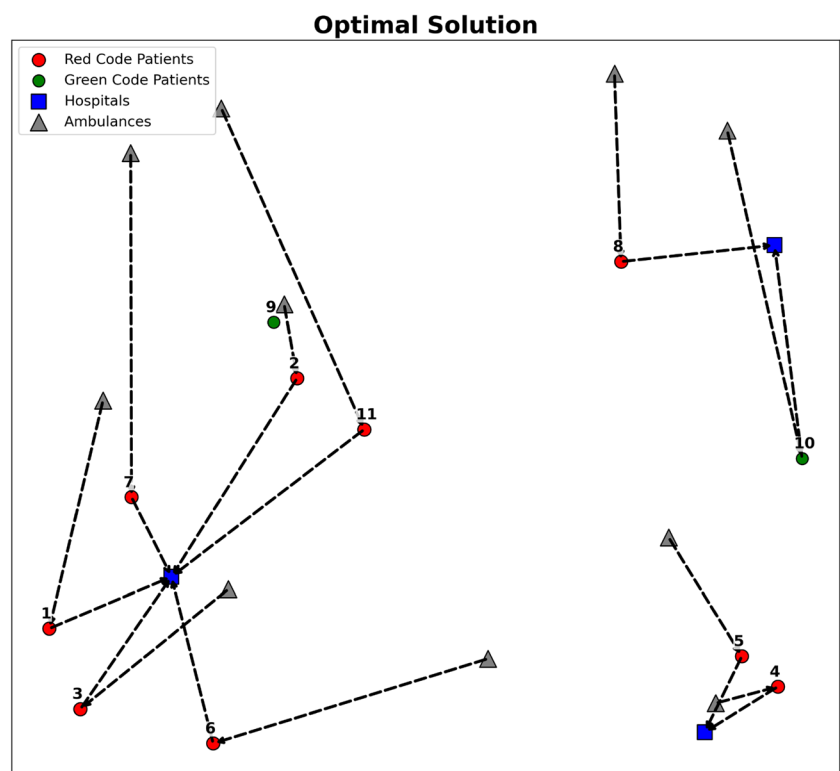
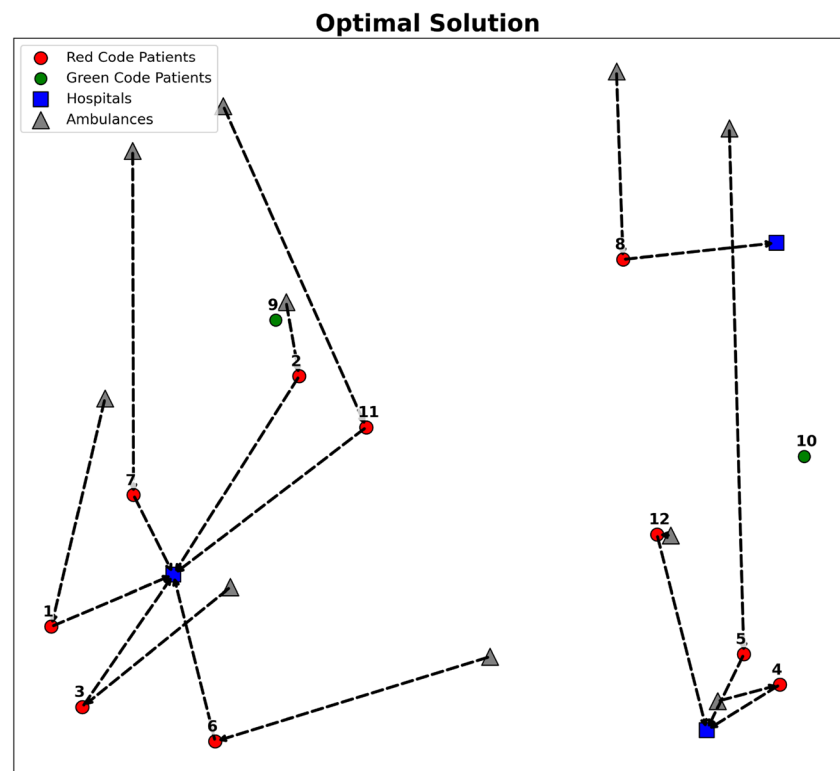


Fig. 13 Dynamic optimization example: Illustration of the optimal solution generated by MILP at time $t + 2$ with the arrival of another new red-code patient 12



Overall, both MILP and HATS approaches demonstrate that the simultaneous use of exact optimization and advanced metaheuristics bridges the gap between theoretical optimality and real-world applicability in EMS systems. This dual capability is particularly valuable for decision support in intelligent transportation systems, where both solution quality and computational efficiency are essential. Although the main experimental studies on ARP assume a static environment, which means that all incidents appear simultaneously and optimization is performed once, we have demonstrated that the ARP application can be easily extended to a dynamic environment, i.e., a series of static instances in a rolling-horizon manner that require continual re-optimization as new information becomes available (i.e., new incidents) [37], which better reflects real-world EMS operations.

In real-time operational EMS problems, the ambulance dispatching problem addressed in this study is only one of the key operational challenges. Another equally important problem is the ambulance-relocation problem, which involves strategically repositioning idle ambulances to waiting stations in order to improve service coverage and reduce response times [36]. However, effective relocation decisions depend on several factors, including population density, historical incident patterns, and geographical characteristics, making it difficult to identify the most appropriate waiting station locations. Machine learning techniques can help address this challenge by identifying hidden patterns in historical and spatial data, thereby providing valuable

recommendations to support EMS operators in their strategic decision-making [35]. Therefore, a promising direction for future research is to optimize both ambulance dispatching and relocation problems simultaneously, with the ultimate objective of further reducing ambulance response times and improving the overall efficiency of EMS operations.

Acknowledgements This research is part of the SMARTPLAIGO project funded by the European Union Recovery and Resilience Facility of the NextGenerationEU instrument through the Research and Innovation Foundation under the Grant Agreement No CODEVELOPAG-SH-HE/0823/0112 and the 'EXCELSIOR': ERATOSTHENES: Excellence Research Centre for Earth Surveillance and Space-Based Monitoring of the Environment H2020 Widespread Teaming project (www.excelsior2020.eu). The 'EXCELSIOR' project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 857510, from the Government of the Republic of Cyprus through the Directorate General for the European Programmes, Coordination and Development and the Cyprus University of Technology.

Author Contributions SPZ: Methodology, Writing & Editing, Data Collection, Analysis. MNA: Supervision, Investigation, Funding Acquisition. EK: Conceptualization, Analysis, Supervision. CC: Validation, Writing & Editing. MM: Methodology, Conceptualization, Supervision, Writing & Editing.

Funding This research is funded by the European Union Recovery and Resilience Facility of the NextGenerationEU instrument through the Research and Innovation Foundation under the Grant Agreement No CODEVELOPAG-SH-HE/0823/0112 and the 'EXCELSIOR': ERATOSTHENES: Excellence Research Centre for Earth Surveillance and Space-Based Monitoring of the Environment H2020 Widespread

Teaming project (www.excelsior2020.eu). The ‘EXCELSIOR’ project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No 857510, from the Government of the Republic of Cyprus through the Directorate General for the European Programmes, Coordination and Development and the Cyprus University of Technology.

Data availability and Materials Data are contained within the article.

Declarations

Ethics Approval and Consent to Participate Not applicable.

Consent for Publication Not applicable.

Competing Interests The authors declare no non-financial interests.

References

- Andersson, T., Värbrand, P.: Decision support tools for ambulance dispatch and relocation. *J. Operat. Res. Soc.* **58**(2), 195–201 (2007)
- Andersson, T., Värbrand, P.: Decision support tools for ambulance dispatch and relocation. In: Mustafee, N. (ed.) *Operational Research for Emergency Planning in Healthcare*, vol. 1, pp. 36–51. Palgrave Macmillan UK, London (2016)
- Aringhieri, R., Bigharaz, S., Duma, D., Guastalla, A.: Fairness in ambulance routing for post disaster management. *CEJOR* **30**, 189–211 (2022)
- Aringhieri, R., Bruni, M., Khodaparasti, S., van Essen, J.: Emergency medical services and beyond: Addressing new challenges through a wide literature review. *Comput. Operat. Res.* **78**, 349–368 (2017)
- Bazelmans, C.H., ter Braake, S.L., Schrottenboer, A.H., van Lieshout, R.N., Van Woensel, T.: The vehicle routing problem with pickups, deliveries, and scheduled linehauls. *Eur. J. Oper. Res.* (2025). <https://doi.org/10.1016/j.ejor.2025.07.070>
- Brotcorne, L., Laporte, G., Semet, F.: Ambulance location and relocation models. *Eur. J. Oper. Res.* **147**(3), 451–463 (2003)
- Devarenne, I., Mabed, H., Caminada, A.: Adaptive tabu tenure computation in local search. In: van Hemert, J., Cotta, C. (eds.) *Evolutionary Computation in Combinatorial Optimization*, pp. 1–12. Springer, Berlin Heidelberg, Berlin, Heidelberg (2008)
- Fogue, M., Sanguesa, J.A., Naranjo, F., Gallardo, J., Garrido, P., Martinez, F.J.: Non-emergency patient transport services planning through genetic algorithms. *Expert Syst. Appl.* **61**, 262–271 (2016)
- Garner, A., Lee, A., Harrison, K., Schultz, C.H.: Comparative analysis of multiple-casualty incident triage algorithms. *Ann. Emerg. Med.* **38**(5), 541–548 (2001)
- Gendreau, M., Hertz, A., Laporte, G.: A tabu search heuristic for the vehicle routing problem with time windows. In: *Les Cahiers du GERAD G-94-13*, , Montreal (1994)
- Gendreau, M., Laporte, G., Semet, F.: A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Comput.* **27**(12), 1641–1653 (2001)
- Glover, F.: Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **13**(5), 533–549 (1986)
- Glover, F.: Tabu search—Part I. *ORSA J. Comput.* **1**(3), 190–206 (1989). <https://doi.org/10.1287/ijoc.1.3.190>
- Glover, F., Laguna, M.: *Tabu Search, Operations Research/Computer Science Interfaces Series*, vol. 39. Springer (1997). <https://doi.org/10.1007/978-1-4615-6089-0>
- Haddad, A., Tlili, T., Nouaouri, I., Krichen, S.: Solving the multi-objective ambulance routing problem using nsga iii. In: *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 2296–2300 (2024)
- Jagtenberg, C., Bhulai, S., van der Mei, R.: An efficient heuristic for real-time ambulance redeployment. *Operat. Res. Health Care* **4**, 27–35 (2015)
- Khoshgebari, F., Mirzapour Al-e-Hashem, S.M.J.: Ambulance location routing problem considering all sources of uncertainty: Progressive estimating algorithm. *Comput. Operat. Res.* **160**, 106400 (2023)
- Krauss, A.: A branch and price algorithm for a static ambulance routing problem. Master’s thesis, PUC-Rio, Dept. of Informatics (2023)
- Li, X., Zhao, Z.H., Zhu, X., Wyatt, T.: Covering models and optimization techniques for emergency response facility location and planning: a review. *Math. Methods Oper. Res.* **74**(3), 281–310 (2011). <https://doi.org/10.1007/s00186-011-0354-8>
- Maxwell, M., Restrepo, M., Henderson, S., Topaloglu, H.: Approximate dynamic programming for ambulance redeployment. *INFORMS J. Comput.* **22**(2), 266–281 (2010)
- Mitchell, S., O’Sullivan, M., Dunning, I.: *Pulp: a linear programming toolkit for python*. The University of Auckland, Auckland, New Zealand **65**, 25 (2011)
- Montemanni, R., Gambardella, L., Rizzoli, A., Donati, A.: Ant colony system for a dynamic vehicle routing problem. *J. Comb. Optim.* **10**, 327–343 (2005)
- Ni, E.C., Hunter, S.R., Henderson, S.G., Topaloglu, H.: Exploring bounds on ambulance deployment policy performance. In: *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pp. 1–12 (2012)
- Oberscheider, M., Hirsch, P.: Analysis of the impact of different service levels on the workload of an ambulance service provider. *BMC Health Serv. Res.* **16**(487), (2016)
- Rabbani, M., Oladzad, N., Akbarian, N.: Ambulance routing in disaster response considering variable patient condition: Nsga-ii and mopso algorithms. *J. Indust. Manage. Optim.* **18**(2), 1035–1062 (2022)
- Ramirez-Nafarrate, A., Fowler, J.W., Wu, T.: Design of centralized ambulance diversion policies using simulation-optimization. In: *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pp. 1251–1262 (2011)
- Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **40**(4), 455–472 (2006)
- Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: *International conference on principles and practice of constraint programming*, pp. 417–431. Springer (1998)
- Talarico, L., Meisel, F., Sörensen, K.: Ambulance routing for disaster response with patient groups. *Comput. Operat. Res.* **56**, 120–133 (2015)
- Tavakkoli-Moghaddam, R., Memari, P., Talebi, E.: A bi-objective location-allocation problem of temporary emergency stations and ambulance routing in a disaster situation. In: *2018 4th International Conference on Optimization and Applications (ICOA)*, pp. 1–4 (2018)
- Tikani, H., Setak, M.: Ambulance routing in disaster response scenario considering different types of ambulances and semi soft time windows. *J. Indust. Syst. Eng.* **12**(1), 95–128 (2019)
- Tlili, T., Abidi, S., Krichen, S.: A mathematical model for efficient emergency transportation in a disaster situation. *Am. J. Emerg. Med.* **36**(9), 1585–1590 (2018)
- Tlili, T., Harzi, M., Krichen, S.: Swarm-based approach for solving the ambulance routing problem. In: *Proceedings of KES 2017, Procedia Computer Science*, vol. 112, pp. 350–357 (2017)

34. Tlili, T., Nasser, S.B., Chicano, F., Krichen, S.: Tabu search-based hyper-heuristic for solving the heterogeneous ambulance routing problem with time windows. *Intern. J. Intell. Transport. Syst. Res.* **22**, 446–461 (2024)
35. Tlili, R., Badawy, A., Salem, S., Barhamgi, M., Mohamed, A.: A survey of machine learning innovations in ambulance services: Allocation, routing, and demand estimation. *IEEE Open J. Intell. Transport. Syst.* **5**, 842–872 (2024). <https://doi.org/10.1109/OJIT.2024.3514871>
36. Yavari, M., Maihami, R., Esmaili, M.: Ambulance dispatching and relocation problem considering overcrowding of emergency departments. *IIEE Trans. Healthcare Syst. Eng.* **12**(4), 263–274 (2022). <https://doi.org/10.1080/24725579.2022.2064008>
37. Yoon, S., Albert, L.A.: A dynamic ambulance routing model with multiple response. *Transport. Res. Part E: Logist. Transport. Rev.* **133**, 101807 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Sofoclis P. Zambirinis received the B.Sc. degree in Mathematics from the National and Kapodistrian University of Athens, Greece, in 2009 (GPA 9.4/10, top 1% of his class), the M.Phil. degree in Statistical Science from the University of Cambridge, UK, in 2010, and the Ph.D. degree in Management Science from Lancaster University, UK, in 2019, with a specialization in optimization for vehicle routing and scheduling problems. He is currently a Postdoctoral Researcher at the

ERATOSTHENES Centre of Excellence, Cyprus, in the Department of Big Earth Data Analytics. His research interests include mathematical programming, combinatorial and multi-objective optimization, metaheuristic algorithms, vehicle routing and scheduling, disruption management, hyperbolic geometric graphs, network geometry, temporal and complex networks, and their applications to real-world optimization and decision-making problems.



Maria N. Anastasiadou holds a PhD in Electrical and Computer Engineering from the University of Cyprus, an MSc in Medical Physics from the University of Surrey, and a BSc in Physics from the National and Kapodistrian University of Athens. She is currently a Researcher B at the ERATOSTHENES Centre of Excellence, where she leads and contributes to research activities in project management, Earth observation applications, and data-driven decision-support systems.

With over six years of experience managing and coordinating large-scale research and innovation projects funded by the European Union, national authorities, and industry, she has played key roles in the preparation, technical coordination, and implementation of complex interdisciplinary initiatives. Prior to her current role, she worked as a Research Associate at the KIOS Research and Innovation Center of Excellence, overseeing major ICT, public health, and smart infrastructure projects. Her research interests include AI-driven optimization, public health surveillance systems, crisis management applications, and the integration of Earth observation and data analytics for operational decision-making.



Efthymios Kyriacou is an Assistant Professor in the Department of Electrical, Computer Engineering and Informatics at the Cyprus University of Technology and Head of the University's eHealth lab (<https://ehealth.cut.ac.cy/>). He has over 25 years of experience in eHealth systems, emergency telemedicine, medical imaging, and intelligent medical applications. He has more than 200 scientific papers published in refereed journal, conference proceedings and book chapters in

these areas with an h-index: 35 at Google Scholar and h-index: 27 at Scopus, Nov. 2025. He has participated in numerous EU- and nationally funded research projects and has served as Guest Co-Editor for seven Special Issues. He is co-editor of the book *Ultrasound and Carotid Bifurcation Atherosclerosis* (Springer, 2012, 78K downloads Bookmetrix Sept. 2025). He is a Senior Member of IEEE and a founding member and first Chair (2008–2019) of the IEEE Cyprus Engineering in Medicine and Biology / Signal Processing Chapter.



Charalambos Chrysostomou received the B.Sc. degree in Computer Science from the University of Nicosia, Cyprus, in 2006, the M.Sc. degree in Computational Intelligence and Robotics from De Montfort University, U.K., in 2007, and the Ph.D. degree in Bioinformatics from De Montfort University, U.K., in 2012. He is currently a Research Scientist at the ERATOSTHENES Centre of Excellence, Limassol, Cyprus. Prior to this role, he served as an Associate Research

Scientist at The Cyprus Institute (2019–2024) and as a Postdoctoral Research Associate at the University of Leicester, U.K. (2012–2016). His research interests include machine learning, deep learning, bioinformatics, signal processing, and the application of data-driven methodologies in medical imaging and Earth observation. He has served as a session chair and editor for various IEEE Engineering in Medicine and Biology Society (EMBC) conferences and journals.



Michalis Mavrovouniotis received the B.Sc. (Hons) degree in Computer Science from the University of Leicester, UK (2008), the M.Sc. degree in Natural Computation from the University of Birmingham, UK (2009), and the Ph.D degree in Computer Science from the University of Leicester, UK (2013). He is the Coordinator of the Big Earth Data Analytics Department of the ERATOSTHENES Centre of Excellence. His research interests include evolutionary computation,

neural computation, swarm intelligence, memetic computing, combinatorial optimization problems, heuristic and metaheuristics, artificial intelligence in dynamic and uncertain environments, and relevant real-world applications. He has contributed to several funded projects with partners in both academia and industry. Dr. Mavrovouniotis has more than 70 peer-reviewed publications with h-index of 29 according to Google Scholar. He is currently the Chair of the IEEE Task Force on Evolutionary Computation in Dynamic and Uncertain Environments, a member of the IEEE Computational Intelligence Society and an Associate Fellow of the UK Higher Education Academy.