Ambulance routing for disaster response with patient groups

Reference:
Talarico Luca, Meisel Frank, Sörensen Kenneth.- Ambulance routing for disaster response with patient groups
Computers & operations research - ISSN 0305-0548 - 56(2015), p. 120-133
Full text (Publisher's DOI): https://doi.org/10.1016/J.COR.2014.11.006
To cite this reference: https://hdl.handle.net/10067/1236260151162165141

Ambulance Routing for Disaster Response with Patient Groups

Luca Talarico[a,*], Frank Meisel[b], Kenneth Sörensen[a]

[a] *University of Antwerp, Dept. ENM, Faculty of Applied Economics, Antwerp, Belgium*
[b] *Christian-Albrechts-University Kiel, Faculty of Business, Economics and Social Sciences, Kiel, Germany*

Abstract

We consider a routing problem for ambulances in a disaster response scenario, in which a large number of injured people require medical aid at the same time. The ambulances are used to carry medical personnel and patients. We distinguish two groups of patients: slightly injured people who can be assisted directly in the field, and seriously injured people who have to be brought to hospitals. Since ambulances represent a scarce resource in disaster situations, their efficient usage is of the utmost importance. Two mathematical formulations are proposed to obtain route plans that minimize the latest service completion time among the people waiting for help. Since disaster response calls for high-quality solutions within seconds, we also propose a Large Neighborhood Search metaheuristic. This solution approach can be applied at high frequency to cope with the dynamics and uncertainties in a disaster situation. Our experiments show that the metaheuristic produces high quality solutions for a large number of test instances within very short response time. Hence, it fulfills the criteria for applicability in a disaster situation. Within the experiments, we also analyzed the effect of various structural parameters of a problem, like the number of ambulances, hospitals, and the type of patients, on both running time of the heuristic and quality of the solutions. This information can additionally be used to determine the required fleet size and hospital capacities in a disaster situation.

Keywords: Ambulance routing, Disaster response, Service time, Local search, Large neighborhood search

1. Introduction

Recent examples such as Hurricane Katrina in 2005, the Indian Ocean tsunami in 2004, or any of the recent armed conflicts around the globe demonstrate that disasters can have a devastating impact on a society. Regardless of whether their cause is natural (e.g., earthquakes, floods, hurricanes, wildfires) or man-made (e.g., terrorist attacks, war situations), disasters can cause large-scale loss of life as well as damage to a society's infrastructure, housing, and industrial complex. It has been widely recognized (see e.g., Berkoune et al. (2012) and Holguín-Veras et al. (2012)) that the severity of a disaster can

[*] corresponding author, luca.talarico@uantwerpen.be, tel.: +32 (0)3 265 41 77, address: ANT/OR - University of Antwerp Operations Research Group, Prinsstraat 13, 2000 Antwerp, Belgium

Email addresses: luca.talarico@uantwerpen.be (Luca Talarico), meisel@bwl.uni-kiel.de (Frank Meisel), kenneth.sorensen@uantwerpen.be (Kenneth Sörensen)

be, to a large extent, influenced by the efficacy of the logistics operations during the response phase. Although the disaster *itself* can certainly cause a lot of casualties, a large fraction of the victims usually perish because of a lack of medical aid in the immediate aftermath of a disaster. Clearly, the post-disaster situation results in the response actions having to be executed under extremely challenging conditions: limited availability of resources (transportation, supplies, manpower, hospital capacity), damaged transportation and communication infrastructure, as well as uncertain information regarding the number and locations of people in need of medical assistance, see e.g., Najafi et al. (2013, 2014) and Yi et al. (2010). Despite these challenges, it is essential that the logistics relief operations are initiated quickly and well planned to be most effective. Hence, there is a strong need for decision support tools that generate solutions to the underlying optimization problems in a few seconds or less (Berkoune et al., 2012). However, research on transportation problems and vehicle fleet management for disaster response operations is emerging only recently, see de la Torre et al. (2012) and Pedraza-Martinez and van Wassenhove (2012). With this paper, we propose a decision support approach for the routing of ambulances in response to a disaster.

The central task of managing ambulances in a disaster response situation is to provide first aid to slightly injured people and to bring seriously injured people to operating hospitals. Managing the operations of ambulances in the immediate aftermath of a disaster is massively complicated by the dynamics and uncertainty with which the planning conditions (especially the relevant information) change over the course of time. The information required to support the planning of ambulances includes the number and location of people calling for help, the availability of ambulances, the capacity of the nearby hospitals, as well as the accessibility of incident sites due to the damaged infrastructure and the current traffic situation, see Jotshi et al. (2009). Another issue is that, in contrast to the daily operations in the public health care sector, the number of requests for help in a disaster situation strongly exceed the capacity of the available ambulance fleet. Hence, it is of utmost importance to use the ambulances efficiently in such a way that they provide as much medical aid as possible.

The response process that is executed by the responsible organizations in the aftermath of a disaster has to be designed in such a way that it is able to cope with the challenges of a dynamic planning situation for the scarce ambulance resource. The routing of ambulances in such a situation can be treated as a static or a dynamic routing problem. In the static case, a set of emergencies requests is collected first and, then, the routing problem is solved for this set of requests. In the dynamic case, the routes of ambulances are updated whenever new help requests arrive, which can reduce the response time. However, this approach requires that communication with the ambulances is possible at all time, which might not be the case in a disaster situation and, furthermore, the rescue teams may perceive this to be disturbing under stressful circumstances. Therefore, in this paper, we consider a three-step response process that aims at solving a static ambulance routing problem, see Fig. 1. The process is executed by a central dispatching unit, which collects requests and manages ambulance operations repeatedly until no further emergency requests are received. The first step is to answer incoming emergency calls and to collect relevant information like the location and the condition of the people being in need of help. The dispatcher collects several requests that are then classified according to their severity in a second process step. The classification reflects the priority with which a patient should receive help, which is

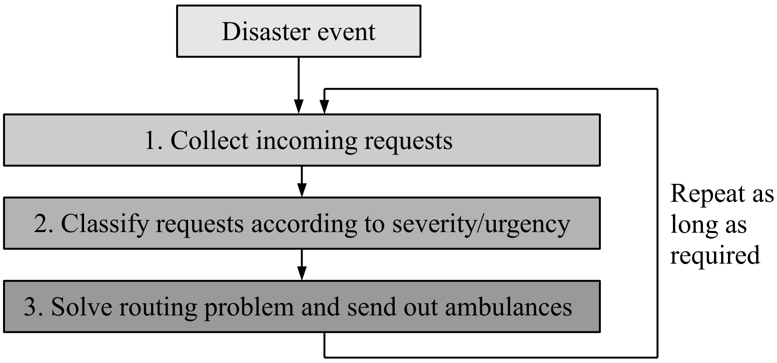
Disaster event
1. Collect incoming requests
2. Classify requests according to severity/urgency
3. Solve routing problem and send out ambulances
Repeat as long as required

Figure 1: Disaster response process.

taken into account when routing the ambulances in the third process step. Collecting and classifying a number of requests before actually sending out the ambulances supports an efficient use of the vehicles, because instead of dispatching ambulances on a first-come first-served basis they can be used to serve the most urgent requests first. Therefore, the first two process steps do not represent a waste of precious time but they collect valuable information to come up with high-quality route plans in the third process step. In fact, the time spent for the first two steps is rather short if numerous requests arrive within short time (as in the case of a disaster event) and if the classification of requests is performed directly while answering an emergency call or automatically from the collected data. Hence, the three-step process can be repeated at high frequency (for example each time a certain number of requests has been collected or a certain time limit has elapsed) such that it causes little delay in the service process. Clearly, if the dispatcher classifies an incoming request as so urgent that it cannot wait at all, a suitable ambulance may be deployed directly without waiting for further requests. This, however, would constitute a mixed static-dynamic response process, which is out of scope of this paper. A further advantage of the sketched three-step process is that up-to-date information regarding the availability of ambulances, infrastructure conditions, etc. can be included in the planning.

The scope of this paper is to investigate the routing problem that occurs in the third step of the response process. The ambulances are used to bring medical personnel to the casualties and to carry injured people to the hospitals. Each ambulance carries medical personnel that can provide first aid to slightly injured people in the field. Seriously injured individuals are accompanied by the medical staff on their way to the hospital where skilled doctors are available. According to this, we distinguish two types of patients:

- *Red code patient*: A person with red code classification is seriously injured and needs to be brought to a hospital by an ambulance.
- *Green code patient*: A person with green code classification is slightly injured and can be helped directly in the field.

There exist more detailed classification schemes for patients (see e.g., Andersson and Värbrand, 2007; Gennarelli and Wodzin, 2008) and several so-called *triage* systems have been developed for classifying and prioritizing patients rapidly in a mass-casualty incident with an overwhelming number of victims, limited time and scarce medical resources, see Killeen et al. (2006). The goal of triage is to allocate a limited set of medical resources to patients such that these resources are used as efficiently as possible, providing the best possible care to a large number of patients. The triage system therefore assigns priority to those patients who will substantially benefit from a rapid intervention, even if these patients are not the most critical ones. This makes disaster response different from civil health care where resources are usually not scarce and the most severe patients always receive highest priority. Typical triage systems classify and prioritizes patients based on their conditions into four groups (Garner et al., 2001): patients who require immediate transportation to a hospital, patients who can wait some time for transportation, patients who require no hospital treatment, and patients who are unlikely to survive at all. Including further categories (as done for example in Gennarelli and Wodzin, 2008) allows for a finer distinction of patients and their needs but makes the application of triage systems more difficult. However, the two types of patients considered in this paper are sufficient to distinguish the fundamental tasks that have to be performed by the ambulances, namely serving patients in the field and bringing them to hospitals. For this reason, we just consider two patient classes in this paper.

Concerning the routing of ambulances, we assume that each of them can carry one red code patient at a time and that each patient is directly brought to a hospital after having been picked up. The decision to which hospital to bring a patient is part of the routing problem and depends on the capacities of hospitals. Since green code patients can be helped on the field, an ambulance can go directly to the next patient after having served a green code patient. From this, an ambulance can provide help to multiple people on its route before returning to a hospital. In contrast, if an ambulance has to serve multiple red code patients, it has to visit several hospitals throughout the planning horizon. Therefore, for the purpose of clarity, we refer here to a route as a tour that begins at one hospital, visits one or more patients in a specified sequence, and ends at either the starting hospital or at some other hospital. Hence, an ambulance may perform multiple routes within a solution to the ambulance routing problem.

The optimization problem is then to determine ambulance routes to serve the two groups of patients, red code and green code patients, which have been determined in the first two steps of the sketched response process. The objective is to minimize the sum of the latest service completion time among the red code patients and the latest service completion time among the green code patients. The objective strives to reduce the longest waiting time faced by a patient in a group. Although some authors propose to minimize the average waiting time of patients (e.g., Créput et al. 2011), the objective pursued in our paper ensures that no patient suffers from an excessively long waiting time. This maximizes the probability of survival for the patient who has to wait longest to be served. Furthermore, we also introduce weights for the latest completion times of the two patient groups. These weights can be used to reflect the higher urgency of red code patients, but they can also result in green code patients being served before red code patients. The latter reflects real-world triage systems for mass-casualty incidents where priority is given to those patients who benefit most from a rapid medical treatment without expending valuable resources on those for whom there is little hope of recovery. Such a prioritisation is proposed, for example, by Benson et al. (1996) who compute the expected benefit of rendering care with the cost of achieving that benefit in order to assign the highest priority to those patients whose treatment yields the greatest value. The weighted objective function pursued in our paper supports such a tradeoff of the severity of a help request and the medical resources required for it. In order to assess a solution with regard to this performance measure, not only the routes but also the service start and completion times, i.e., the scheduling of the ambulance operations, have to be determined. With this paper, we provide the first models and algorithms to solve this problem. Since in a disaster situation high-quality ambulance routes have to be determined in short response times, we particularly strive to develop powerful heuristic solution methods.

The paper is organized as follows. In Section 2 we review the relevant literature and relate it to our study. The routing problem is described and modeled in Section 3 together with an illustrating example. In Section 4, we present a Large Neighborhood Search metaheuristic to solve the emergency routing problem. The models and the metaheuristic are computationally tested in Section 5. Section 6 concludes the paper.

2. Literature Review

There exist several streams of research that address locating, dispatching, and routing ambulances and supplies in public health care and in disaster response situations.

Locating ambulances entails finding deployment sites for the vehicles within an (urban) area such that a certain response time is guaranteed to reach the potential emergency sites within this area. Surveys of models and algorithms developed in this field of research are provided by Brotcorne et al. (2003) and Farahani et al. (2012). The approaches typically belong to the class of covering location models, see for example the early work of Fitzsimmons and Srikar (1982). Relocation of ambulances comes into the play when the coverage becomes inadequate due to ambulances that are currently dispatched to incidents. In this case, idle vehicles may have to be relocated to fill gaps in the coverage, which leads to a dynamic ambulance location problem, see Gendreau et al. (2001). Recent research aims at capturing realistic planning situations like traffic-dependent traveling times and congestion phenomena. For example, Schmid and Doerner (2010) consider travel times that vary in the course of a day within an ambulance location problem. From such variations, the coverage achieved throughout a day by a certain deployment of ambulances changes dynamically which calls for relocations. The authors propose a model and a Variable Neighborhood Search metaheuristic to simultaneously optimize the coverage for various points in time with varying traffic volumes. Knight et al. (2012) present a model to locate ambulances in such a way that the expected survival probability of heterogeneous patients is maximized. The patients differ in the targeted response time and in their medical conditions. An approximation method is proposed to solve this type of ambulance location problem.

Dispatching is the task of assigning incoming emergency requests to ambulances. It is sometimes solved in combination with the ambulance location problem. For example, Toro-Díaz et al. (2013) present an integrated location and dispatching model that captures the impact of queuing patients in congested server systems on the achieved response time and coverage. A Genetic Algorithm is proposed to assign locations and requests to the vehicles. Andersson and Värbrand (2007) dispatch ambulances according to the urgency of requests and the closeness of a vehicle to the site of an incident. The authors combine the dispatching with a relocation of ambulances in order to maintain the coverage of the service area when some of the ambulances are busy serving patients. Also Schmid (2012) combines dispatching and relocation where approximate dynamic programming is used to minimize the expected total response time of requests that occur within the planning horizon. The routing of ambulances is out of scope of these papers, because dispatching is concerned with assigning a single emergency request to a suitable ambulance.

Ambulance routing is considered in some studies as the problem of finding a shortest (fastest) path from one location to another taking into account traffic conditions and the infrastructure damage caused by a disaster, see e.g., Jotshi et al. (2009) and Goldberg and Listowsky (1994). In our paper, ambulance routing is considered as the problem of finding vehicle routes for a set of ambulances to serve a given set of patients. Such problems are often seen as dynamic or real-time vehicle routing problems since emergencies occur in the course of time at unforeseeable locations, see e.g., the surveys of Ghiani et al. (2003) and Pillac et al. (2013). If there is a strong degree of dynamism and stochasticity, the routing problem may be solved by a reactive dispatching policy, cf. Bertsimas and van Ryzin (1991). However, if several requests occur in short time, as is assumed in our study, the problem is to find routes each comprising several patients such that all requests are served. Such a problem is solved in Créput et al. (2011) by means of a multi-agent approach and local search heuristics with the objective of minimizing the average waiting time of patients. Wex et al. (2014) investigate a multiple traveling salesman problem that finds its application in the routing of rescue units that have to serve a given set of incidents. The authors propose several (meta-)heuristics to find routes that minimize the total weighted completion time of the incidents. It is assumed that all patients receive aid in the field such that transportation to hospitals is not part of the problem. Transportation of non-urgent patients among hospitals, from homes to hospitals, or vice versa can be considered as a dial-a-ride problem, which is to relocate patients from their individual origin location to their destination, see e.g., Parragh (2011) and Parragh et al. (2012). A dynamic stochastic version of this problem arises if patients brought to a hospital are discharged the same day with a certain probability such that their return has to be added to the vehicle routes, see Schilde et al. (2011). Since the transport requests are not urgent in these problems, the typical objective is to minimize the travel effort of vehicles or the tardiness of violated time windows. Also, the destination of each patient is prescribed in these problems whereas in disaster response it needs to be decided to which hospital to bring a patient.

Another stream of research is on disaster relief routing for which de la Torre et al. (2012) provide a recent literature survey. Here, the scope is on the distribution of humanitarian aid supplies like water, food, medicine, and survival equipment from distribution centers to demand points like refugee camps with respect to the available transport capacities, see e.g., Berkoune et al. (2012). In this field, various routing problems have been investigated. For example, Campbell et al. (2008) present models and heuristics for traveling salesman and vehicle routing problems that aim at minimizing the latest arrival time or the average arrival time at demand locations as is of relevance in time-critical disaster response actions. Huang et al. (2012) investigate a vehicle routing problem to distribute supplies from a depot with the goal of a fair allocation of scarce supplies if not all demands can be met. Rath and Gutjahr (2014) combine distribution planning with locating supply depots such that a cost measure is minimized and a maximal coverage is achieved. A hierarchical traveling salesman problem where demand locations require supplies with different urgency levels is investigated in Panchamgam et al. (2013). A multi-commodity flow problem to serve demand locations in multiple truck trips at minimum time is presented in Berkoune et al. (2012). However, the transportation of patients is out of scope of all these papers.

A few papers propose multi-commodity flow models to combine the distribution of supplies with the transportation of patients. Yi and Özdamar (2007) present such a model with the objective of minimizing the weighted sum of unsatisfied demands and waiting times of injured people. The patients have to be brought to hospitals and to emergency centers that are set up temporarily to cope with the disaster. Özdamar and Demir (2012) provide a similar model that minimizes the total vehicle travel time in order to ensure an efficient utilization of transport capacity and a fast delivery of supplies. Najafi et al. (2013, 2014) provide extensions of the model of Yi and Özdamar (2007) to cope with different vehicle types and to support re-planning and robust solutions in dynamic and stochastic planning situations. The approaches investigated in these papers are all based on multi-commodity network flow problems where a detailed routing of vehicles is typically out of scope. Furthermore, although different categories of patients are distinguished, all patients have to be brought to a medical station to be treated. Considering different types of services (first aid in the field for slightly injured people, transportation to hospitals for heavily injured people) is not supported by these models.

From this literature review, we observe that disaster response management is a very active field of research. However, research mainly concentrates on locating and dispatching of ambulances and on the distribution of supplies. Ambulance routing has received only some attention in the literature. In particular, the routing problem investigated in this study, where some patients can be served on the field whereas others need to be brought to hospitals, has not been treated so far.

3. The Ambulance Routing Problem

3.1. Problem Description

The aim of the ambulance routing problem is to find routes for a fleet of ambulances in order to give aid to a set of patients. We formalize this problem using the notation shown in Table 1. Let $\mathcal{R}$ denote the set of red code patients who have to be picked up by ambulances to be brought to the hospitals in set $\mathcal{H}$. Let $\mathcal{G}$ denote the set of green code patients who can receive aid directly in the field. The set of all patients is denoted by $\mathcal{P} = \mathcal{R} \cup \mathcal{G}$. The fleet of ambulances available to give aid to patients is denoted by $\mathcal{K}$. Each ambulance is initially located at a hospital. We denote by $\mathcal{K}_h \subseteq \mathcal{K}$ the subset of ambulances that are initially located at hospital $h \in \mathcal{H}$. A corresponding binary parameter f_h^k indicates whether ambulance k is initially located at hospital h ($f_h^k = 1$) or not ($f_h^k = 0$). Furthermore, we denote by $\mathcal{A} = \{\mathcal{P} \times \mathcal{P}\} \cup \{\mathcal{H} \times \mathcal{P}\} \cup \{\mathcal{P} \times \mathcal{H}\}$ the set of arcs that are of relevance for the routing problem where t_{ij} is the travel time needed by an ambulance to traverse arc $(i, j) \in \mathcal{A}$. A service time d_i is associated to each patient $i \in \mathcal{P}$. For red code patients, d_i denotes the time required to prepare the patient for transportation to a hospital. For green code patients, d_i denotes the time needed to give first aid to the patient in the field. For the ease of notation, we also define a transfer time d_h for each hospital $h \in \mathcal{H}$, which represents the time required to drop off a red code patient at this hospital. Finally, c_h denotes the capacity of hospital $h \in \mathcal{H}$ in terms of the maximum number of red code patients who can be brought to this location. We assume that the total capacity of all hospitals is sufficiently large to host all red code patients, i.e., $\sum_{h\in\mathcal{H}} c_h \geq |\mathcal{R}|$. We also assume that each ambulance can carry at most one red code patient at a time and that an ambulance has to go directly to a hospital with residual capacity after taking up a red code patient. In contrast, having served a green code patient in the field, an ambulance

Table 1: Notation used to model the ambulance routing problem.

Sets:	
$\mathcal{R}$	set of red code patients
$\mathcal{G}$	set of green code patients
$\mathcal{P}$	set of all patients, $\mathcal{P} = \mathcal{R} \cup \mathcal{G}$
$\mathcal{H}$	set of hospitals
$\mathcal{K}_h$	set of ambulances that are initially located at hospital $h \in \mathcal{H}$
$\mathcal{K}$	set of all ambulances, $\mathcal{K} = \cup_{h \in \mathcal{H}} \mathcal{K}_h$
$\mathcal{A}$	set of arcs in a problem, $\mathcal{A} = \{\mathcal{P} \times \mathcal{P}\} \cup \{\mathcal{H} \times \mathcal{P}\} \cup \{\mathcal{P} \times \mathcal{H}\}$
Parameters:	
f_h^k	binary parameter, 1 iff ambulance k is initially located at hospital h (i.e. $k \in \mathcal{K}_h$)
t_{ij}	travel time from i to j with $(i,j) \in \mathcal{A}$
d_i	service time of patient $i \in \mathcal{P}$
d_h	transfer time to drop off a red code patient at hospital $h \in \mathcal{H}$
c_h	capacity of hospital $h \in \mathcal{H}$
w_{R}	priority given to red code patients
w_{G}	priority given to green code patients
Decision variables:	
x_{ij}^k	binary, 1 iff ambulance k serves patient i directly before patient j (3-Index Model)
x_{ij}	binary, 1 iff any ambulance serves patient i directly before patient j (2-Index Model)
u_{ih}	binary, 1 iff red code patient i is brought to hospital h
b_i	visiting time of patient $i \in \mathcal{P}$, $b_i \geq 0$
e_{R}	latest service completion time among all red code patients
e_{G}	latest service completion time among all green code patients

can go directly to the next patient (green code or red code) on its route. We assume that each ambulance finishes its last route at any hospital.

In order to evaluate the quality of a solution, we define the *service completion time* of a red code patient as the point in time when the patient is dropped off at the assigned hospital. The service completion time of a green code patient is given by the completion of the first aid. The objective of the ambulance routing problem is to minimize a weighted linear combination of the latest service completion time e_{R} among all red code patients and the latest service completion time e_{G} among all green code patients. The latest service completion time among the patients of a group is considered here because it minimizes the worst case waiting time. Furthermore, e_{R} and e_{G} are weighted by parameters w_{R} and w_{G}, respectively, to express the relative importance that a decision maker probably assigns to the patient groups. In particular, if red code patients shall be served with utmost priority, a setting $w_{\mathrm{R}} \gg w_{\mathrm{G}}$ will ensure that these patients are served early in the routing. We next provide an illustrative example. Afterwards, two mathematical models of the problem are presented respectively in Sections 3.3 and 3.4.

3.2. Illustrative Example

We illustrate the problem on a small artificial example. This instance contains three red code patients $\mathcal{R} = \{r_1, r_2, r_3\}$ and seven green code patients $\mathcal{G} = \{g_1, g_2, \ldots g_7\}$. Two hospitals $\mathcal{H} = \{h_1, h_2\}$ are available to take up the red code patients, with respective capacities $c_{h_1} = c_{h_2} = 3$. Two ambulances a_1 and a_2 are initially located at hospital h_1, i.e., $\mathcal{K}_{h_1} = \{a_1, a_2\}$. A third ambulance a_3 is initially located at h_2, i.e., $\mathcal{K}_{h_2} = \{a_3\}$. Figure 2 illustrates the locations of all patients and hospitals. We assume here that the travel times t_{ij} of arcs $(i,j) \in \mathcal{A}$ correspond to the Euclidean distance between locations

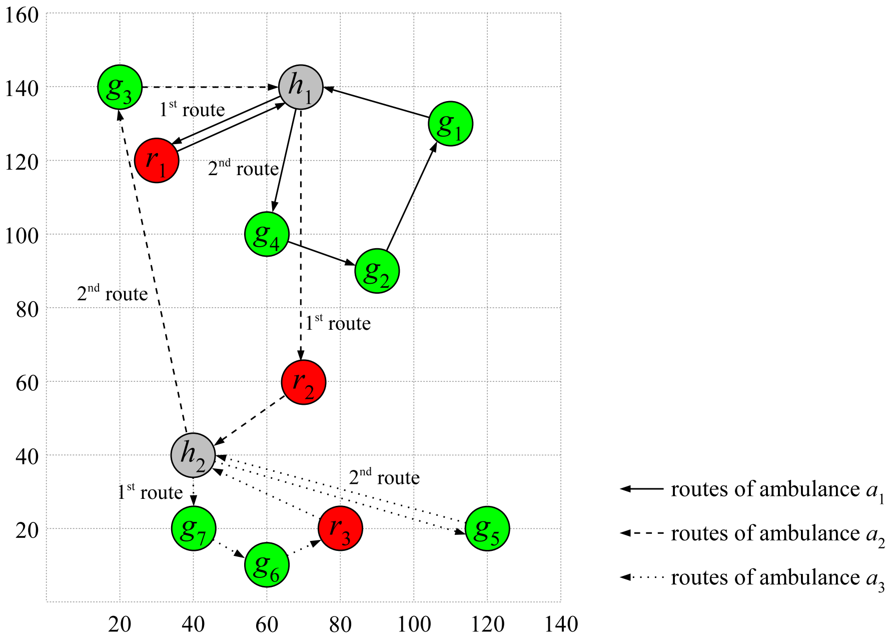

Figure 2: Example solution.

i and j. The service times d_i are set to 30 time units for red code patients $i \in \mathcal{R}$ and to 10 time units for green code patients $i \in \mathcal{G}$. We assume that dropping off a patient at a hospital $h \in \mathcal{H}$ can be done in no time, i.e., $d_h = 0$.

Figure 2 shows a potential route plan for the three ambulances. In this solution, each ambulance performs two routes. Ambulance a_1 starts its first route at hospital h_1, picks up red code patient r_1 and brings this patient to hospital h_1. On its second route, ambulance a_1 serves three green code patients g_1, g_2, and g_4. Ambulance a_2 first picks up red code patient r_2 and brings it to hospital h_2. Afterwards, it serves patient g_3 before returning to hospital h_1. Ambulance a_3 starts at hospital h_2 and combines the service of two green code

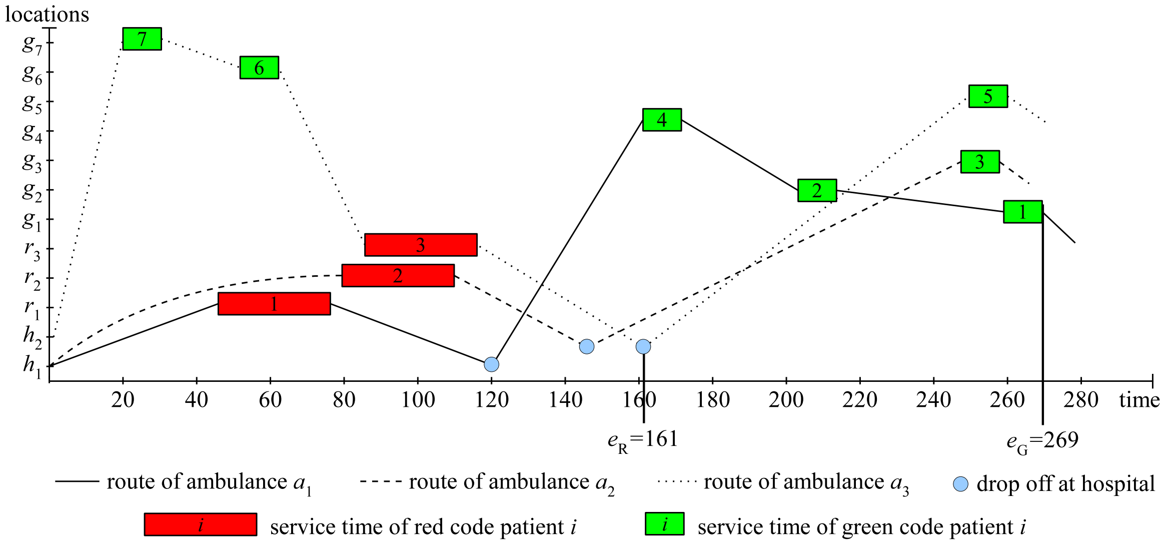

Figure 3: Time-space representation of the example solution.

patients with the service of red code patient r_3. After having brought r_3 to hospital h_2, patient g_5 is served in a second route. In order to determine the quality of this solution, the time-space diagram in Fig. 3 shows the positions of all ambulances over the course of time. It can be seen that the latest drop off of a red code patient at a hospital takes place at time $e_{\mathrm{R}} = 161$. The latest completion time of serving a green code patient is $e_{\mathrm{G}} = 269$. Note that a route can start and end at different hospitals, which enables solutions of high quality where patients are served as quickly as possible. In our example, this is the case for the routes of ambulance a_2.

3.3. Mathematical Formulation: A 3-Index Model

Using the notation introduced in this section, we propose a mathematical formulation of the routing problem. The model uses 3-indexed binary decision variables x_{ij}^k, which take value 1 if ambulance k serves patient i directly before patient j and 0 otherwise. Binary variables u_{ih} take value 1 if red code patient $i \in \mathcal{R}$ is brought to hospital h and 0 otherwise. The visiting time of patient i, i.e., the arrival time of the ambulance that gives aid to this patient is represented by a continuous variable $b_i \geq 0$. The ambulance routing problem is modeled by (1)–(13).

$$\min \quad w_{\mathrm{R}} \cdot e_{\mathrm{R}} + w_{\mathrm{G}} \cdot e_{\mathrm{G}} \tag{1}$$

s.t.

$$\sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{hj}^k = f_h^k \qquad \forall h \in \mathcal{H}; k \in \mathcal{K} \tag{2}$$

$$\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{ji}^k = 1 \qquad \forall i \in \mathcal{P} \tag{3}$$

$$\sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{ji}^k = \sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{ij}^k \qquad \forall i \in \mathcal{P}; k \in \mathcal{K} \tag{4}$$

$$\sum_{h \in \mathcal{H}} u_{ih} = 1 \qquad \forall i \in \mathcal{R} \tag{5}$$

$$\sum_{i \in \mathcal{R}} u_{ih} \leq c_h \qquad \forall h \in \mathcal{H} \tag{6}$$

$$b_i + d_i + t_{ij} \leq b_j + \left(1 - \sum_{k \in \mathcal{K}} x_{ij}^k\right) \cdot M \qquad \forall i \in \mathcal{G} \cup \mathcal{H}; j \in \mathcal{P} \tag{7}$$

$$b_i + d_i + t_{ih} + d_h + t_{hj} \leq b_j + \left(2 - \sum_{k \in \mathcal{K}} x_{ij}^k - u_{ih}\right) \cdot M \qquad \forall i \in \mathcal{R}; j \in \mathcal{P}; h \in \mathcal{H} \tag{8}$$

$$e_{\mathrm{G}} \geq b_i + d_i \qquad \forall i \in \mathcal{G} \tag{9}$$

$$e_{\mathrm{R}} \geq b_i + d_i + u_{ih} \cdot (t_{ih} + d_h) \qquad \forall i \in \mathcal{R}; h \in \mathcal{H} \tag{10}$$

$$b_i \geq 0 \qquad \forall i \in \mathcal{P} \cup \mathcal{H} \tag{11}$$

$$u_{ih} \in \{0, 1\} \qquad \forall i \in \mathcal{R}; h \in \mathcal{H} \tag{12}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall (i, j) \in \mathcal{A}; k \in \mathcal{K} \tag{13}$$

The objective function (1) aims to minimize the weighted sum of the latest service completion time among all red code patients and the latest service completion time among all green code patients. Constraints (2) ensure that each ambulance originates from the hospital where it is initially located. According to Constraints (3), each patient is visited exactly once by one of the ambulances. Constraints (4) enforce that an ambulance visiting a patient also has to leave that patient's location. Consequently, ambulances finish their routes in one of the hospitals. Constraints (5) and (6) enforce that each red code patient is assigned to exactly one hospital and that the capacity of each hospital is respected. Constraints (7) and (8) propagate the arrival times of ambulances at the patient locations. According to (7), the arrival time b_j of an ambulance at a patient j is determined by the arrival time b_i at the location i (a green code patient or a hospital) visited immediately prior to patient j, the service time d_i at location i, and the travel time t_{ij} to go from i to j. If, however, i represents a red code patient, the detour to bring i to its assigned hospital needs to be included into the calculation of the arrival time at j. This is ensured by (8). Here, if an ambulance serves a red code patient i immediately prior to patient j (i.e., $\sum_{k\in\mathcal{K}} x_{ij}^k = 1$) and if i is assigned to hospital h (i.e., $u_{ih} = 1$) then (8) ensures that the arrival time b_j at patient j is at least as large as $b_i + d_i + t_{ih} + d_h + t_{hj}$, which also includes the time to go to hospital h and drop off patient i before proceeding to patient j. This approach allows to include multiple intermediate returns of an ambulance to a hospital into a solution. Constraints (9) determine the latest service completion time e_G among all first aid services provided to green code patients. Constraints (10) determine the latest service completion time e_R of all red code patients. Note that the service of a red code patient is completed at the time when the patient is dropped off at the assigned hospital. Constraints (11)-(13) define the domains of the decision variables.

Considering the special case of the routing problem with a single ambulance, a single hospital, no red code patients $\mathcal{R} = \emptyset$, and neglected service times $d_i = 0$ of green code patients $i \in \mathcal{G}$, the problem reduces to the traveling salesman problem which is known to be NP-hard (Karp, 1972). Therefore, the problem studied in this paper is also NP-hard.

3.4. A 2-Index Model

Model (1)–(13) uses 3-indexed variables x_{ij}^k for the routing of ambulances $k \in \mathcal{K}$. However, since all ambulances are identical except for their initial locations, we can reformulate the model by dropping index k. Hence, the size of the model can be reduced with an expected positive impact on the computation time needed by a MIP solver to find an optimal solution to the routing problem. In order to reformulate the model, we introduce the binary variable $x_{ij} \in \{0, 1\}$, which takes value 1 if any ambulance serves patient i directly before patient j. All further notation is as before.

$$\min \quad w_\mathrm{R} \cdot e_\mathrm{R} + w_\mathrm{G} \cdot e_\mathrm{G} \tag{14}$$

s.t.

$$\sum_{j\in\mathcal{P}\cup\mathcal{H}} x_{hj} \leq |\mathcal{K}_h| \qquad \forall h \in \mathcal{H} \tag{15}$$

$$\sum_{j\in\mathcal{P}\cup\mathcal{H}} x_{ji} = \sum_{j\in\mathcal{P}\cup\mathcal{H}} x_{ij} = 1 \qquad \forall i \in \mathcal{P} \tag{16}$$

$$\sum_{h\in\mathcal{H}} u_{ih} = 1 \qquad \forall i \in \mathcal{R} \tag{17}$$

$$\sum_{i\in\mathcal{R}} u_{ih} \leq c_h \qquad \forall h \in \mathcal{H} \tag{18}$$

$$b_i + d_i + t_{ij} \leq b_j + (1 - x_{ij}) \cdot M \qquad \forall i \in \mathcal{G} \cup \mathcal{H}; \forall j \in \mathcal{P} \qquad (19)$$

$$b_i + d_i + t_{ih} + d_h + t_{hj} \leq b_j + (2 - x_{ij} - u_{ih}) \cdot M \qquad \forall i \in \mathcal{R}; j \in \mathcal{P}; h \in \mathcal{H} \qquad (20)$$

$$e_{\mathrm{G}} \geq b_i + d_i \qquad \forall i \in \mathcal{G} \qquad (21)$$

$$e_{\mathrm{R}} \geq b_i + d_i + u_{ih} \cdot (t_{ih} + d_h) \qquad \forall i \in \mathcal{R}; h \in \mathcal{H} \qquad (22)$$

$$b_i \geq 0 \qquad \forall i \in \mathcal{P} \cup \mathcal{H} \qquad (23)$$

$$u_{ih} \in \{0, 1\} \qquad \forall i \in \mathcal{R}; h \in \mathcal{H} \qquad (24)$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in \mathcal{A} \qquad (25)$$

The objective function in (14) is identical to the objective function (1) of the 3-index model. Constraints (15) ensure that at most $|\mathcal{K}_h|$ ambulances start from hospital h. Constraints (16) conserve the flow of ambulances at patient locations. Constraints (19) and (20) determine the arrival times at the patients based on the new routing variables x_{ij}. Constraints (25) define these binary variables. Constraints (17)-(18) and (21)-(24) are taken from the 3-index model.

3.5. Model Refinements

The proposed models contain M-terms that are used to compute arrival times at patient locations. In order to support MIP solvers in coping with these formulations, in this paragraph we describe how to determine a value that is sufficiently large to serve as M. The general idea is to compute for each patient i the maximum time $t_i^{\max}$ that is needed to reach and to serve this patient. For green code patients $i \in \mathcal{G}$, $t_i^{\max} = \max_{k \in \mathcal{G} \cup \mathcal{H}}\{t_{ki}\} + d_i$ because the corresponding ambulance can be either located at another green code patient or at some hospital right before going to patient i. For red code patients $i \in \mathcal{R}$, $t_i^{\max} = \max_{k \in \mathcal{G} \cup \mathcal{H}}\{t_{ki}\} + d_i + \max_{h \in \mathcal{H}}\{t_{ih} + d_h\}$, including also the longest possible time that is needed to drop off patients i at any of the hospitals. Supposing that in the worst case all patients are served by the same ambulance, therefore $M = \sum_{i \in \mathcal{P}} t_i^{\max}$ represents an upper bound on the arrival time at any patient.

The models may further be extended in different ways. One issue is to have a good distribution of ambulances across the region at the end of the service process, especially if the methodology is used within a repetitive process. This can be achieved by enforcing that ambulances (i.) return to the depots where they started from, or (ii.) are equally distributed across hospitals, or (iii.) are located close to the area where additional patients are most likely to appear. For the 3-index model, these goals are modeled by Constraints (26) to (28).

$$\sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{jh}^k = f_h^k \qquad \forall h \in \mathcal{H}; k \in \mathcal{K} \qquad (26)$$

$$\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{jh}^k \leq \left\lceil \frac{|\mathcal{K}|}{|\mathcal{H}|} \right\rceil \qquad \forall h \in \mathcal{H} \qquad (27)$$

$$\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{jh}^k = a_h \qquad \forall h \in \mathcal{H} \qquad (28)$$

Constraints (26) enforce that each ambulance finally returns to its initial hospital location whereas (27) equally distribute the ambulances among the hospitals. Constraints (28) can be used to enforce a certain number a_h of ambulances at hospital h. In order to locate ambulances close to locations where additional patients are most likely to appear, one can add artificial hospitals h' to set $\mathcal{H}$. Such an artificial hospital has no capacity ($c_{h'} = 0$) and no initial ambulances ($\mathcal{K}_{h'}$) but it can be used for absorbing a certain number $a_{h'}$ of ambulances at the end of the service process. The corresponding constraints for the 2-index model are given by (29) to (31). Constraints (29) guarantee that the initial number of ambulances is finally located at each hospital again whereas Constraints (30) balance the number of ambulances among the hospitals at the end of the service process. Constraints (31) guarantee a certain number of ambulances, which can also be used for deploying ambulances at arbitrary locations that are represented by artificial hospitals added to set $\mathcal{H}$.

$$\sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{jh} = |\mathcal{K}_h| \qquad \forall h \in \mathcal{H} \tag{29}$$

$$\sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{jh} \leq \left\lceil \frac{|\mathcal{K}|}{|\mathcal{H}|} \right\rceil \qquad \forall h \in \mathcal{H} \tag{30}$$

$$\sum_{j \in \mathcal{P} \cup \mathcal{H}} x_{jh} = a_h \qquad \forall h \in \mathcal{H} \tag{31}$$

Please note that in the remainder of the paper, we consider the basic models described in Sections 3.3 and 3.4 without the extensions described by Constraints (26)-(31).

4. Solution Approach

The circumstances in which the ambulance routing problem described in this paper is solved require a fast and robust solution approach. To be useful in disaster situations, the ambulance routing problem must be solved within seconds in order to respond properly to the emergency requests and to replan the routing if updated information becomes available. In addition, the quality of the solutions is an important aspect determining the waiting times for the patients, which should obviously be as short as possible. These reasons warrant the development of a (meta)heuristic solution approach, which is usually faster than an exact approach, and is expected to produce solutions of near-optimal quality.

We propose here a Large Neighborhood Search (*LNS*) metaheuristic to solve the ambulance routing problem. The iterative nature of the *LNS* metaheuristic and the presence of diversification mechanisms allow the procedure to escape from local optima such that various parts of the solution space can be explored in a limited amount of computing time. The operating principle of the *LNS* metaheuristic is based on three stages:

- *Initial stage*: An initial solution for the ambulance routing problem is generated by one of two randomly selected heuristic approaches, which are described in Section 4.1.
- *Intensification stage*: The current solution is improved by a large scale neighborhood search (so-called *Variable Neighborhood Descent (VND)* heuristic), which uses nine different local search operators. The *VND* heuristic is described in Section 4.2.
- *Diversification stage*: To reach unexplored areas of the solution space, the current solution is first partially destroyed by selecting randomly one of three destroy operators described in Section 4.3. Afterwards it is reconstructed by applying a repair operator. The modified solution then becomes the input of the intensification stage.

Two parameters I and L determine the behavior of the *LNS* procedure. I is the total number of iterations performed by the algorithm. L is a limit on the number of iterations without improvement. If L iterations have been performed without finding a new best solution, *LNS* restarts the search process by generating a new initial solution. This generation process is randomized to produce a solution that has not been investigated earlier in the search process. The *LNS* method is outlined in further detail in Algorithm 1. After the initialization phase, the method starts its first iteration ($i = 0$) by constructing an initial solution, see lines 9 to 17. Here, one of the two available heuristics for the generation of new solutions is picked randomly. Furthermore, a parameter α (described in

Algorithm 1: *LNS* Metaheuristic

```
1  Initialize metaheuristic parameters $I$ and $L$;
2  Let $s^*$ be the best solution found so far and $f(s^*)$ be its objective function value;
3  Let $s$ be the current solution and $f(s)$ be its objective function value;
4  $s^*, s \leftarrow \emptyset$, $f(s^*), f(s) \leftarrow \infty$;
5  Let $i$ be the iteration counter;
6  Let $l$ be the counter for iterations without improvement;
7  $i \leftarrow 0$ , $l \leftarrow 0$;
8  while ($i < I$) do
9      if ($i = 0 \vee l = L$) then                                   // Initial stage
10         $random \leftarrow RandInt([0,1])$;
11         $\alpha \leftarrow RandInt([2,5])$;
12         switch ($random$) do
13             case ($random = 0$)
14                 $s \leftarrow InsertionHeuristic(\alpha)$;
15             case ($random = 1$)
16                 $s \leftarrow ConstructiveHeuristic(\alpha)$;
17         $l \leftarrow 0$;
18     else                                        // Diversification of existing solution
19         $random \leftarrow RandInt([0,2])$;
20         switch ($random$) do
21             case ($random = 0$)
22                 $s \leftarrow Rem_2(s)$;
23             case ($random = 1$)
24                 $s \leftarrow Rem_{rand}(s)$;
25             case ($random = 2$)
26                 $s \leftarrow Rem_{all}(s)$;
27         $s \leftarrow Repair(s)$;
28     $s \leftarrow VND(s)$;                                          // Intensification
29     if ($f(s) < f(s^*)$) then                                // New best solution?
30         $s^* \leftarrow s$;
31         $l \leftarrow 0$;
32     else
33         $l{+}{+}$;
34     $i{+}{+}$;
35 Return $s^*$.                                                // Return best solution
```

Section 4.1) is determined and given to these procedures to guide the randomized solution construction process. In later iterations, when a solution already exists, the search process is diversified to escape from local optima by randomly selecting a destroy operator and by repairing the resulting solution, see lines 19 to 27. The search is intensified by applying the *VND*-local search method to the current solution in line 28. In line 29, it is checked whether a new best solution is found, which is then stored. In this case, counter l for the number of iterations without improvement is reset, otherwise incremented (line 33). If the counter reaches the limit L, the *LNS* method generates a new initial solution, see again line 9. After a total of I iterations, the *LNS* metaheuristic terminates by returning the best solution found. The components of the *LNS* metaheuristic are described in detail in the following sections.

4.1. Initial Stage

The *LNS* procedure embeds two different heuristics to generate initial feasible solutions to the ambulance routing problem. The first heuristic is referred to as *Insertion heuristic.* It starts by building a single route for all green code patients and then inserts the red code patients one after the other. The second heuristic is called the *Constructive heuristic.* It builds routes simultaneously for green and red code patients. Both procedures are designed to respect the different services required by green code patients (which can be served in the field where, afterwards, the ambulance can go directly to the next patient) and red code patients (which have to be picked up and brought to a hospital with free capacity). Furthermore, both methods contain random components to deliver different solutions as is needed to exploit the restart capability of the *LNS* metaheuristic.

The *Insertion heuristic* is outlined in Algorithm 2. It initially produces a single giant route that connects all green code patients $i \in \mathcal{G}$, see line 2. For this purpose, we solve a traveling salesman problem (TSP) using the well known heuristic of Lin and Kernighan (1973). We have chosen the Lin-Kernighan heuristic as it is considered to be one of the most effective methods for the TSP, and has found the best-known solutions to a large number of benchmark problems. The method used in our paper is the modified Lin-Kernighan

Algorithm 2: Insertion heuristic

```
1  Let $a$ be a randomly selected ambulance and let $h$ be the hospital where $a$ is located;
2  Let $r$ be a TSP route that starts and ends at $h$ visiting all patients in $\mathcal{G}$;
3  Assign route $r$ to ambulance $a$;
4  while (not all patients have been visited) do
5      Let $j$ be a randomly selected and so far unvisited red code patient;
6      Let $C$ be a candidate list of $\alpha$ least-cost positions to insert $j$ into current routes;
7      Randomly select an insertion position $i \in C$;
8      Split the corresponding route right after position $i$;
9      Append $j$ to the first sub-route and close this route by appending the nearest hospital
       with free capacity;
10     Let $\hat{a}$ be the ambulance that becomes available earliest in the current solution;
11     Let $\hat{h}$ be the current location of $\hat{a}$;
12     Let the second sub-route start at $\hat{h}$ and assign this route to $\hat{a}$;
13 Return solution $s$.
```

Algorithm 3: Constructive heuristic

```
1  while (not all patients have been visited) do
2      Let $a$ be the ambulance that becomes available earliest in the current solution;
3      Let $h$ denote the hospital where ambulance $a$ is currently located;
4      Define a new route $r$ for ambulance $a$ with starting point $h$;
5      Let $C^P$ be a candidate list of $\alpha$ unserved patients who are located closest to $h$;
6      Let $i$ be a patient randomly selected from $C^P$;
7      Append $i$ to route $r$;
8      if ($i \in \mathcal{R}$) then                    // Red code patient requires closing the route
9          Let $C^H$ be a candidate list of $\alpha$ available hospitals that are closest to $i$;
10         Let $h$ be a hospital randomly selected from $C^H$;
11         Close route $r$ by appending hospital $h$;
12     else                                              // Route $r$ is potentially extendible
13         while (not all the patients have been visited) do
14             Let $C^P$ be a candidate list of $\alpha$ unserved patients who are located closest to $i$;
15             Let $j$ be a patient randomly selected from $C^P$;
16             if ($j \in \mathcal{R}$) then             // Red code patient requires closing the route
17                 Append $j$ to route $r$;
18                 Let $C^H$ be a candidate list of $\alpha$ available hospitals that are closest to $j$;
19                 Let $h$ be a hospital randomly selected from $C^H$;
20                 Close route $r$ by appending hospital $h$;
21                 break;                                // Restart from line 1
22             else                                      // Route $r$ can be extended
23                 Let $\hat{a} \neq a$ be the ambulance that becomes idle earliest in current solution;
24                 Let $T_{\hat{a}}$ denote the time at which $\hat{a}$ becomes idle;
25                 Let $\hat{h}$ be the hospital where ambulance $\hat{a}$ is currently located;
26                 Let $b_i$ be the time at which patient $i$ is visited by ambulance $a$;
27                 if ($b_i + d_i + t_{ij} \leqslant T_{\hat{a}} + t_{\hat{h}j}$) then   // Extend route $r$ of ambulance $a$
28                     Append $j$ to route $r$;
29                     $i \leftarrow j$;
30                 else                                  // Close route $r$ of ambulance $a$
31                     Let $C^H$ be a list of $\alpha$ hospitals that are closest to $i$;
32                     Let $h$ be a hospital randomly selected from $C^H$;
33                     Close route $r$ by appending hospital $h$;
34                     break;                            // Restart from line 1
35 Return solution $s$.
```

heuristic proposed in Helsgaun (1998, 2000, 2006), which is a highly efficient implementation from a computational point of view. Once the giant route has been generated for the green code patients, the red code patients are inserted using a variant of the insertion heuristic proposed by Solomon (1987). Since red code patients have to be brought to hospitals, including such a patient in a route actually means to split this route into two new ones. More precisely, the insertion procedure iterates through the red code patients and includes one patient $j \in \mathcal{R}$ into the route plan per iteration. For this purpose it determines a candidate list C of α feasible and least-cost insertion positions to include j into one of the current routes, see line 6. Then, the procedure randomly selects an insertion position $i \in C$ and splits the corresponding route into two sub-routes, see lines 7 and 8. The first sub-

Table 2: Local search operators used in the *VND* heuristic.

N_λ	Intra Route Operators	N_λ	Inter Route Operators
N_1	Internal Patients Relocate	N_5	External Patients Relocate
N_2	Internal Patients Swap	N_6	External Patients Swap
N_3	Internal Patients 2-Opt	N_7	External Patients 2-Opt
N_4	Single Hospital Change	N_8	Hospitals Swap
		N_9	Route reassignment

route contains the patients up to and including i. The second sub-route contains patient $i + 1$ and the following ones. Then, red code patient j is appended to the first sub-route and this route is closed by selecting the hospital that takes up this patient, see line 9. For the second sub-route, a new ambulance and, thus, starting location is determined in lines 10 to 12. The procedure ends if all patients have been added to the solution.

The *Constructive heuristic* is outlined in Algorithm 3. It builds a solution step-by-step by adding one patient at a time to a route plan until all patients $\mathcal{P}$ are served by the ambulances. The procedure starts by creating a new route for one of the ambulances in lines 2 to 4. Then, using a greedy randomized selection process, a patient i is randomly selected from a restricted candidate list C^P of α unserved patients and added to the route, see lines 5 to 7. If the selected patient i is a red code patient, the route is closed by selecting a hospital from a candidate list C^H of α closest hospitals with free capacity, see lines 8 to 11. Otherwise, if the patient i is a green code patient, the route is potentially extendible by adding further patients, which is investigated in lines 12 to 34. For this purpose, an unserved patient j that is nearby patient i is randomly selected in lines 14-15. If patient j is a red code patient, j is added to the route, a hospital is selected, the route is closed, and the procedure restarts a new iteration, see lines 16 to 21. Otherwise, if j is a green code patient, it is checked in lines 23 to 26 whether there exists an alternative ambulance $\hat{a}$ that can reach j at an earlier point in time than the currently considered ambulance a. If this is not the case, j is appended to the current route r of ambulance a, see lines 27 to 29. If, however, ambulance $\hat{a}$ can reach j earlier than ambulance a, route r of ambulance a is closed and the algorithm starts a new iteration, see lines 30 to 34. Patient j is then added to a new route in one of the following iterations of the *Constructive heuristic*. The procedure ends if all patients have been added to the solution.

4.2. Intensification Stage

During the intensification stage of the *LNS* metaheuristic, solutions are improved by means of local search. For this purpose, we have adopted several of the most common local search operators for vehicle routing problems (Bräysy and Gendreau, 2005). In total, we use nine local search operators that are listed in Table 2 and described afterwards. The four *Intra Route Operators* search for improvements within a route:

- *Internal Patients Relocate:* A green code patient contained in a route is relocated to another position within this route. If the route contains a red code patient, the green code patient is not allowed to be placed behind the red code patient, because the latter needs to be brought to a hospital directly. Therefore, red code patients appear only at the end of a route and, hence, relocating them using the local search is not an option.

- *Internal Patients Swap:* The positions of two green code patients who both belong to the same route are exchanged.

- *Internal Patients 2-Opt:* Two edges $(i, i{+}1)$ and $(j, j{+}1)$ contained in one route are replaced by edges (i, j) and $(i{+}1, j{+}1)$. Such a neighborhood move also reverses the order in which the patients in between $i{+}1$ and j are served. Again, moves that would relocate a red code patient are forbidden.

- *Single Hospital Change:* The hospital at which the route ends is replaced by another hospital. If the route involves a red cod patient, the new destination hospital must have at least one free capacity unit.

The following five *Inter Route Operators* address the routings of two different ambulances:

- *External Patients Relocate:* A patient i is removed from an ambulance route and inserted into another route. Let j denote the patient in the new route behind which patient i is inserted. The neighborhood move replaces edges $(i{-}1, i)$, $(i, i{+}1)$ and $(j, j{+}1)$ by edges $(i{-}1, i{+}1)$, (j, i) and $(i, j{+}1)$. Note that the operation is only feasible if j is not a red code patient.

- *External Patients Swap:* Two patients i and j that are served in different routes are exchanged. The neighborhood move replaces edges $(i{-}1, i)$, $(i, i{+}1)$, $(j{-}1, j)$ and $(j, j{+}1)$ by edges $(i{-}1, j)$, $(j, i{+}1)$, $(j{-}1, i)$ and $(i, j{+}1)$.

- *External Patients 2-Opt:* This operator considers two patients i and j belonging to different routes. Both routes are split right after patients i and j and the detached sub-routes are exchanged. In other words, all patients who followed patient i on its original route now follow patient j and vice versa. The move is performed by replacing edges $(i, i{+}1)$ and $(j, j{+}1)$ by edges $(i, j{+}1)$ and $(j, i{+}1)$.

- *Hospitals Swap:* The destination hospitals at which two different routes end are exchanged. For example, consider two routes r_1 and r_2 that end at hospitals h_1 and h_2, respectively. The hospitals are swapped such that route r_1 now ends at h_2 whereas route r_2 now ends at h_1. A swap is only feasible if it does not violate the hospital capacities. For example, if route r_1 includes a red code patient and r_2 does not, the new destination hospital h_2 of route r_1 must have a free capacity unit.

- *Route reassignment:* This operator removes a route from its ambulance and assigns it to another ambulance.

The local search operators are combined in a *Variable Neighborhood Descent (VND)* heuristic that is outlined in Algorithm 4. The *VND* heuristic improves the current solution by exploring the nine neighborhoods one after the other. Each neighborhood is examined by a first-improvement descent strategy, accepting only feasible moves that lead to an improvement of the current solution. Although the order in which the neighborhoods are investigated may have an impact on the quality of the obtained solutions, we did not observe such an effect in some preliminary experiments. Therefore, *VND* explores the neighborhoods in the order shown in Table 2. If, during the search process, a solution is found that is better than the best solution known so far, the best solution is updated

Algorithm 4: *VND* heuristic

```
Let $s$ be the current solution and $f(s)$ be its objective function value;
Let $s^*$ be the best solution found so far and $f(s^*)$ be its objective function value;
$\lambda \leftarrow 1$;                                    // Start with first neighborhood
repeat
    $s \leftarrow N_\lambda(s)$;                           // Find best solution within neighborhood
    if $(f(s) < f(s^*))$ then                              // New best solution?
        $s^* \leftarrow s$;
        $\lambda \leftarrow 1$;                            // Restart with first neighborhood
    else
        $\lambda{+}{+}$;                                   // Continue with next neighborhood
until $(\lambda = 9)$;
Return $s^*$.
```

and *VND* intensifies the search by restarting from the first neighborhood. The procedure terminates if the current solution cannot be further improved by any of the local search operators and, thus, a common local optimum has been reached for all the neighborhoods.

Note that the quality of a solution can only be improved by modifying routes of those ambulances that serve the green code patient and the red code patient with the latest service completion times e_{G} and e_{R}. We refer to the vehicles serving these patients as the *critical green ambulance* and the *critical red ambulance*. In order to speed up the *VND* heuristic, we restrict the local search to the routes of the two critical ambulances. Since each of these ambulances may have assigned multiple routes in a solution, all nine neighborhoods can yield improvements even if the search is restricted to the two vehicles.

4.3. Diversification Stage

In order to escape from the local optima that are reached in the *intensification stage*, a diversification mechanism is used to reach unexplored areas of the solution space. The diversification strategy consists of a *destroy step* that eliminates some of the routes contained in the best solution found so far. Afterwards, a *repair step* is performed to assign the now unserved patients to non-destroyed routes or to build new routes for them. Then, *LNS* again applies the *intensification stage* to improve the obtained solution and so on. For the *destroy step*, three different operators have been implemented:

- *Remove two routes (Rem$_2$)*: This operator destroys two routes of the current solution. The first route to be destroyed is the one that contains the green code patient with the latest service completion time, i.e., the patient who determines e_{G}. The second route is the one that contains the red code patient who determines value e_{R}. If both these patients are served in the same route, only this single route is destroyed.
- *Remove a random number of routes (Rem$_{rand}$)*: Let the *critical green ambulance* (*critical red ambulance*) be the vehicle that serves the green (red) patient who determines the objective value e_{G} (e_{R}). Each of these vehicles can perform more than one route in the current solution to bring various red code patients to hospitals and to serve a number of green code patients. Therefore two random numbers $rand_{\mathrm{G}}$ and $rand_{\mathrm{R}}$ are uniformly generated in the ranges $[1, N_{\mathrm{G}}]$ (for the critical green ambulance) and $[1, N_{\mathrm{R}}]$ (for the critical red ambulance) respectively, where N_{G} represents

the maximum number of routes contained in the critical green ambulance, while N_{R} is the maximum number of routes assigned to the critical red ambulance. Then the destroy operator removes $rand_{\mathrm{G}}$ and $rand_{\mathrm{R}}$ of these routes from the green and red critical ambulances.

- *Remove all routes (Rem$_{all}$)*: All the routes of both critical ambulances are destroyed. If the critical green ambulance coincides with the critical red ambulance, only the routes performed by this single ambulance are removed.

Each time the *diversification stage* is applied, one of the aforementioned destroy operators is randomly selected. The destroyed solution is then repaired using the greedy randomized selection mechanism proposed for the *Constructive heuristic* of Section 4.1.

5. Computational Study

In this section, we describe the computational experiments that were executed to evaluate the performance of the proposed models and heuristics. Since the ambulance routing problem described in this paper has not been studied before, no benchmark instances are available in the literature. We therefore generate a large set of test instances in Section 5.1, which are made available to other researchers upon request.

The computational experiments are divided into three parts. In the first experiment, described in Section 5.2, we compare the two optimization models regarding their potential to produce optimal solutions for the ambulance routing problem. In Section 5.3, we assess the performance of the *LNS* metaheuristic. In the third experiment, presented in Section 5.4, we perform sensitivity tests to analyze the relationship between the structure of a problem instance and its solution.

5.1. Test Instances and LNS Parameter Setting

In order to test both the mathematical models and the metaheuristic algorithm presented in this paper, a large set of test instances has been generated to capture various planning situations. We have produced instances with a varied number of red code and green code patients, hospitals, hospital capacities, and ambulances. These parameters are varied as follows:

- Total number of patients:
  low ($|\mathcal{P}| = 10$), *medium* ($|\mathcal{P}| = 25$), *high* ($|\mathcal{P}| = 50$)
- Percentage of red patients:
  low ($|\mathcal{R}|=25\%\cdot|\mathcal{P}|$), *medium* ($|\mathcal{R}|=50\%\cdot|\mathcal{P}|$), *high* ($|\mathcal{R}|=75\%\cdot|\mathcal{P}|$)
- Number of hospitals: $|\mathcal{H}| = 1$, 2, 3 or 4
- Hospital capacity:
  low ($\sum_{h\in\mathcal{H}} c_h = 1 \times |\mathcal{R}|$), *medium* ($\sum_{h\in\mathcal{H}} c_h = 1.5 \times |\mathcal{R}|$), *high* ($\sum_{h\in\mathcal{H}} c_h = 2 \times |\mathcal{R}|$)
- Number of ambulances:
  low ($|\mathcal{K}| = 0.05 \times |\mathcal{P}|$), *medium* ($|\mathcal{K}| = 0.25 \times |\mathcal{P}|$), *high* ($|\mathcal{K}| = 0.5 \times |\mathcal{P}|$)

One test instance was produced for each combination of the above parameters, which yields a total of 324 instances. For each instance, the locations of hospitals and patients was randomly drawn in an area of size 200 × 200, with travel times t_{ij} corresponding to

Table 3: Variation of weights evaluated in the experiments.

Weights		Relative importance of	# of
w_{G}	w_{R}	red code patients	instances
1	1	50%	324
1	2	67%	324
1	5	83%	324
1	10	91%	324
Total			1296

the Euclidean distance. The service duration d_i of red code patients $i \in \mathcal{R}$ was randomly drawn from the interval $[2, 15]$. The service duration d_i of green code patients $i \in \mathcal{G}$ was selected in the interval $[5, 35]$. Dropping off a red code patient at a hospital is done in no time, i.e., $d_h = 0 \ \forall h \in \mathcal{H}$. The available $|\mathcal{K}|$ ambulances and the hospital capacity are shared randomly among the hospitals contained in an instance.

In order to test the impact of the priority assigned to red and green code patients, we further associated the 324 instances to different combinations of weights w_{G} and w_{R}. The four combinations shown in Table 3 were considered. Associating these parameters to each of the test instances yielded a total of 1296 instances for the experiments.

To perform the experiments, the *LNS* metaheuristic was coded in Java. The metaheuristic requires two parameters to be set: the number of iterations I to perform and the number of non-improving iterations L after which the heuristic generates another initial solution. It is clear that a higher value I makes finding better solutions more likely at the expense of a longer computation time. In preliminary experiments, we observed that the heuristic converges quickly, and that $I{=}200$ offers a good compromise between runtime and solution quality. Moreover, we set $L = I/10$ as this delivered good results in the pretests. Finally, since the metaheuristic involves various elements of randomness, finding better solutions may also be achieved by repeating the solution process a number of times. For this purpose, we repeat the heuristic 50 times when solving an instance. In the remainder of this section, the presented *LNS*-solutions are the best out of these 50 runs and the reported *cpu*-times are the total for all runs. All experiments have been performed on an Intel core i7-2760QM 2.40 GHz processor with 4 GB RAM.

5.2. Evaluation of the Optimization Models

To compare the two models presented in Section 3, we solve them using the MIP-solver CPLEX 12.4, see Ilog (2013). More precisely, we apply CPLEX to solve both models for each of the 324 test instances with weights w_{G} and w_{R} both set to 1. For each instance, a maximum computation time of one hour was defined as a stopping condition whenever the optimal solution is not obtained. We report in Table 4 aggregated results for each model and for the sets of 108 instances with a low ($|\mathcal{P}| = 10$), a medium ($|\mathcal{P}| = 25$), and a high ($|\mathcal{P}| = 50$) number of patients. The table shows the number of integer feasible solutions (*#feas*) found for an instance set, the number of optimal solutions (*#opt*), the average lower bound (*LB*), the average objective function value (*obj*), and the average computation time (*cpu*) required by CPLEX. Column *imp* provides the percentage gap between the average objective function values of the solutions found by the 2-index model and the 3-index model.

From the results, we see that the 3-index model consistently delivers feasible solutions only for the small sized instances, whereas the 2-index model delivers feasible solutions for

Table 4: Computational results achieved by the two proposed models.

# of patients	# of instances	3-Index Model *#feas.* [-]	*#opt.* [-]	*LB* [-]	*obj* [-]	*cpu* [sec.]	2-Index Model *#feas.* [-]	*#opt.* [-]	*LB* [-]	*obj* [-]	*cpu* [sec.]	*imp* [%]
low	108	108	79	611	881	1416	108	79	617	881	1165	0.0
medium	108	107	9	279	1399	3524	108	21	359	1069	3011	23.5
high	108	82	0	256	2205	3600	108	3	270	1363	3504	38.2
Δ							+27	+15	+33	-391	-287	

all instances. The 2-index model also yields more optimal solutions for medium and large instances. The obtained lower bounds are stronger but still too weak for assessing the quality of the solutions, in particular for the larger instances. From the objective function values, we take that the 3-index model is clearly outperformed by the 2-index model for medium and large instances, where the 2-index model achieves average improvements of up to 38.2%. Row Δ in Table 4 aggregates the key performance measures. It shows that the 2-index model delivers additional 27 feasible and 15 optimal solutions. The lower bound increases by 33 units on average and the objective function value reduces by 391 units on average. Furthermore, the average computation time is about five minutes lower than for the 3-index model. It becomes clear that the 2-index model is superior with respect to all the key performance measures. However, even the computation time required by this model clearly exceeds what is considered applicable in a disaster response process. For this reason, the model's results can be used for an assessment of heuristics but it appears inappropriate to apply the model itself to solve the ambulance routing problem in practice.

5.3. Evaluation of the LNS Metaheuristic

For the second experiment, all 1296 instances are solved using the *LNS* metaheuristic. The heuristic is evaluated by comparing its results to those obtained by the 2-index model. Table 5 reports key performance measures for both approaches and each subset of 108 test instances with differing weights w_{R} and differing instance size. We see that the number of optimal solutions and the *cpu* times observed for the 2-index model are hardly affected

Table 5: Comparison of results delivered by the 2-index model and the *LNS* metaheuristic.

Weight w_{R}	# of patients	# of instances	2-Index Model *#opt* [-]	*obj* [-]	*cpu* [sec.]	*LNS* *#opt* [-]	*obj* [-]	*cpu* [sec.]	*#imp* [-]	*rel. imp.* *worst* [%]	*avg* [%]	*best* [%]
1	*low*	108	79	881	1165	79	881	2	2	-0.2	0.0	1.9
1	*medium*	108	21	1069	3011	19	1026	24	60	-3.1	3.0	17.1
1	*high*	108	3	1363	3504	3	1085	126	79	-2.0	12.6	37.1
2	*low*	108	80	1473	1236	80	1475	1	5	-2.5	-0.1	0.3
2	*medium*	108	18	1870	3112	16	1785	24	64	-4.0	3.3	21.8
2	*high*	108	3	2233	3505	3	1854	135	74	-2.2	10.9	32.4
5	*low*	108	80	3086	1164	79	3091	2	2	-3.4	-0.1	0.8
5	*medium*	108	19	4183	3034	17	3981	26	64	-2.1	3.4	23.7
5	*high*	108	0	4532	3600	0	4013	145	76	-1.3	7.4	29.3
10	*low*	108	84	5655	1046	83	5660	4	0	-2.3	0.0	0.0
10	*medium*	108	16	7834	3135	14	7543	27	68	-1.0	2.8	17.4
10	*high*	108	3	8348	3520	3	7493	159	70	-4.4	6.7	32.2

by the weight $w_{\mathcal{R}}$, indicating that the urgency of red code patients has a limited effect when solving the problem using CPLEX. The *LNS* metaheuristic produces a slightly lower number of optimal solutions. Actually, *LNS* finds 396 out of the 406 optimal solutions identified by CPLEX. Furthermore, the average solution quality is better for all sets of medium sized and large sized instances, i.e. the average objective value *obj* of *LNS* is lower for these sets compared with CPLEX. This shows that the meta-heuristic provides a systematic advantage for these problems. Furthermore, *LNS* is much faster than CPLEX. The *cpu* time for repeating *LNS* 50 times ranges from one second to about 2.5 minutes depending on the instance size. A single solution to a large instance is produced in less than 3 seconds. These computation times show that the *LNS* method is applicable in a dynamic disaster response process, as it produces solutions quasi-instantaneously.

The improvement potential of the heuristic is further analyzed in the last four columns of Table 5. Column *#imp* shows the number of instances of a set for which LNS produces a better solution than CPLEX. We observe that the metaheuristic can hardly achieve improvements for instances of small size whereas it achieves a substantial number of better solutions for instances of medium and large size. In particular, for the problems with a high number of patients, *LNS* delivers better solutions for up to two thirds of the instances in a set. Columns *worst*, *avg*, and *best* reveal the extreme values and the average value of the relative improvements observed over all instances of a set. Note that a negative value in these columns indicates that the heuristic delivered a solution with an objective value larger than the one achieved by CPLEX. The results confirm that LNS and CPLEX produce solutions of almost identical quality for the small instances, but LNS requires just a few seconds for the computation. For instances of medium size, LNS delivers solutions that show a 3%-improvement on average, with a maximum improvement of 23.7% if red code patients are considered urgent ($w_{\mathcal{R}} = 5$). For the large instances, the LNS is clearly advantageous with average improvements ranging from 6.7% ($w_{\mathcal{R}} = 10$) to 12.6% ($w_{\mathcal{R}} = 1$) and maximum improvements of up to 37.1%. These results confirm that the developed heuristic is a powerful solution method in particular when it is required to solve large instances to good quality within short response time.

Since the metaheuristic involves several elements of randomness, we also determine the contribution of these techniques to the generation of high quality solutions. In particular, the metaheuristic randomly decides whether to use the *Insertion heuristic* or the *Constructive heuristic* to generate initial solutions. Moreover, in the diversification stage, the destroy operator used is randomly selected. Finally, the *Insertion heuristic* and the *Constructive heuristic* both use greedy randomized mechanisms to select patients, hospitals, and insertion positions from restricted candidate lists of size α. The value α is a random integer in the interval $[2, 5]$, which is drawn by the *LNS* metaheuristic each time a new initial solution is generated. In order to evaluate the contribution of the different randomization

Table 6: Contribution of *LNS*-components to finding best solutions.

Initial solution		Destroy step		α	
Method	Frequency	Operator	Frequency	Value	Frequency
Insertion heuristic	55%	Rem_2	34%	2	40%
Constructive heuristic	45%	Rem_{rand}	23%	3	32%
		Rem_{all}	43%	4	18%
				5	10%

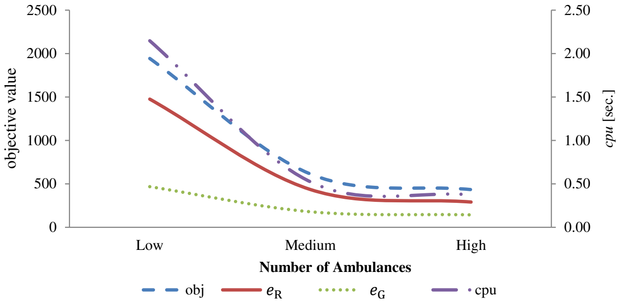

Figure 4: Impact of the number of ambulances.

techniques, the following methodology was used. For each of the final solutions of the 1296 instances, the heuristic that was used to generate the corresponding initial solution, the applied destroy operator, and the value of α were tallied. Table 6 illustrates the relative frequencies observed for the different settings of these parameters while we solved the 1296 instances. It can be seen that the *Insertion heuristic*, the Rem_{all} destroy operator, and a value of $\alpha = 2$ most often result in the best solution. Nevertheless, the other components are involved in the generation of a sizeable number of best solutions.

5.4. Problem Structure and Solution Quality

In the third experiment, we analyze the relationship between the structure of an instance and the best solution found by the *LNS* metaheuristic. We first investigate the impact of the number of ambulances. For this purpose, we distinguish subsets of instances with a low, a medium, and a high number $|\mathcal{K}|$ of ambulances as defined in Section 5.1. The objective weights w_R and w_G are both set to 1. Figure 4 shows the following performance measures averaged over all instances belonging to a subset: objective function value *obj*, latest service completion times e_R and e_G of red code and green code patients, and computation time *cpu* required to produce one solution for a problem instance. Note that e_R is larger than e_G in these results, because the service of red code patients ends at their delivery to a hospital whereas the service of green code patients ends directly after having been treated in the field. As expected, a larger number of ambulances results in a better service (lower objective values). We see that the latest completion times of both red code patients and green code patients benefit from a medium number of ambulances. However, the marginal contribution of additional vehicles decreases such that the objective values do not decline further if a high number of ambulances is available. These computations show that the presented approach can be useful in determining the fleet size required in a disaster situation.

Figure 5 analyzes the impact of the number of hospitals (left figure) and the impact of

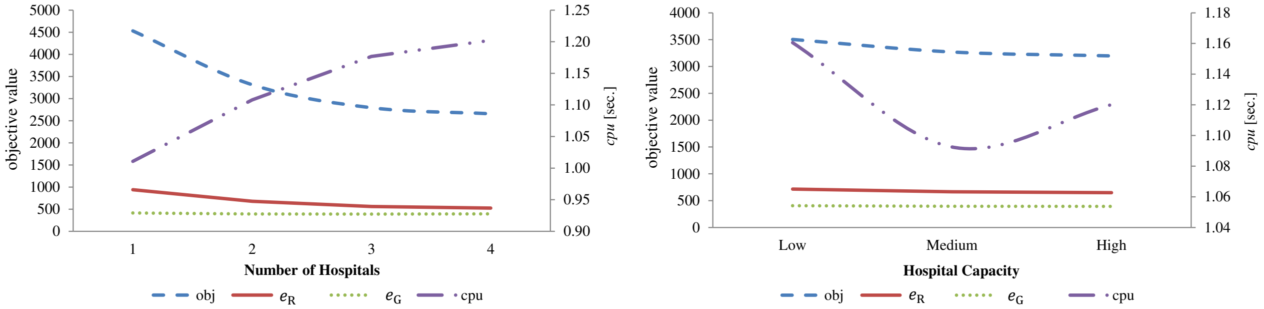

Figure 5: Impact of the number of hospitals (left) and the hospital capacity (right).

the total hospital capacity (right figure). Regarding the number of hospitals, we distinguish instances with $|\mathcal{H}| = 1, 2, 3$, and 4 hospitals. The total hospital capacity $\sum_{h\in\mathcal{H}} c_h$ is varied from low to medium and to high in relation to the number of red code patients as described in Section 5.1. The considered test instances comprise all the combinations of objective weights w_{R} and w_{G} from Table 3. According to Figure 5 (left), a larger number of hospitals clearly helps serving red code patients by reducing the trip duration to transport them to a hospital of free capacity. In fact, the larger the number of hospitals is, the lower the value e_{R} becomes. This experiment shows that the method can also be used to determine whether patients would benefit from setting up temporary hospitals like medical camps. Although the average computation time grows with a larger number of hospitals, it stays around one second even for the largest number of hospitals. The capacity of hospitals seems to have only a minor effect on the obtained solutions, see Figure 5 (right). Although a higher capacity means that there is a higher chance for red code patients to find free capacity at a nearby hospital, we do not observe a significant reduction in the latest service completion time here. The explanation is that the patients and hospitals are widely spread over the whole area in these instances such that most red code patients find a hospital in their surrounding even if the overall capacity is low.

Finally, we investigate the impact of having a low, a medium, or a high percentage of red code patients, see Figure 6(left), and of considering red code patients equally important as green code patients ($w_{\mathrm{R}} = w_{\mathrm{G}} = 1$) or more important ($w_{\mathrm{R}} = 2, 5, 10$), see Figure 6(right). As expected, if the percentage of red code patients increases, the latest service completion time among these patients increases because the ambulances have to bring more patients to the hospitals. At the same time, the latest service completion time e_{G} of green code patients reduces, because fewer such patients need assistance. Interestingly, the computation time decreases although a higher percentage of red code patients means more decisions to assign patients to hospitals. However, an increase in the percentage of red code patients also means a reduction in the number of green code patients, which in turn reduces the number of decisions to group and sequence green code patients on a same ambulance route. The relative importance of red code patients in a solution is controlled by parameter w_{R}. In Figure 6 (right) we see that a larger value w_{R} indeed reduces the latest service completion time e_{R} of red code patients at the cost of the latest service completion time e_{G} of green code patients. Also the average objective function value increases with a larger value of w_{R}, but this can be attributed to the fact that the objective function sums up the two weighted service completion times. This experiment confirms that the metaheuristic effectively considers the weights w_{R} and w_{G} to produce ambulance routes that reflect the different priorities of red code patients and green code patients.

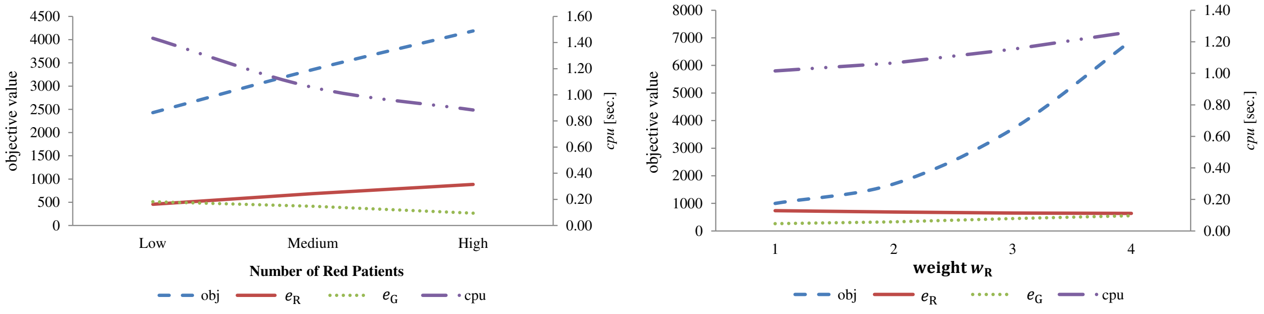

Figure 6: Impact of the number of red code patients (left) and the objective weight w_{R} (right).

6. Conclusions

In this paper an ambulance routing problem for disaster response is investigated, where patients require different types of services. We have proposed two mathematical models. Computational tests show that a 2-index formulation outperforms a 3-index formulation. However, although problem instances will be of rather small size because the routing problem is solved at high frequency in disaster response, the exact solution of the optimization model takes an unacceptably long time. Therefore, a Large Neighborhood Search metaheuristic has been proposed to solve the ambulance routing problem in very short response time, with the aim to assist all patients as fast as possible. Experiments on a very large set of test instances show that the heuristic delivers solutions of excellent quality. Several further experiments demonstrate that the proposed planning approach can be used to support decisions regarding the fleet size of ambulances and the number and capacities of hospitals. Furthermore, the metaheuristic can be controlled to produce routes that take into account the different priority of slightly and seriously injured patients.

Future research may aim at incorporating further aspects such as different types of ambulances, time windows, or constraints on the route length, e.g., before refueling is needed. The models might be extended for example to support ambulances that are capable of transporting more than one red code patient at a time, such that help can be provided faster to people in need of medical assistance. Moreover, the ambulance routing problem may be adapted to the public health care sector. In fact the two classes of patients considered in this paper may also be used to model patients requiring services at their homes (like, for example, physiotherapy) and patients who have to be transported to certain health care facilities to receive hospital treatment.

References

T. Andersson and P. Värbrand. Decision support tools for ambulance dispatch and relocation. *Journal of the Operational Research Society*, 58(2):195–201, 2007.

M. Benson, K. L. Koenig, and C. H. Schultz. Disaster triage: START, then SAVE - a new method of dynamic triage for victims of a catastrophic earthquake. *Prehospital and disaster medicine*, 11(02):117–124, 1996.

D. Berkoune, J. Renaud, M. Rekik, and A. Ruiz. Transportation in disaster response operations. *Socio-Economic Planning Sciences*, 46(1):23–32, 2012.

D. J. Bertsimas and G. van Ryzin. A stochastic and dynamic vehicle routing problem in the euclidean plane. *Operations Research*, 39(4):601–615, 1991.

L. Brotcorne, G. Laporte, and F. Semet. Ambulance location and relocation models. *European Journal of Operational Research*, 147(3):451–463, 2003.

O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118, 2005.

A. M. Campbell, D. Vandenbussche, and W. Hermann. Routing for relief efforts. *Transportation Science*, 42(2):127–145, 2008.

J.-C. Créput, A. Hajjam, A. Koukam, and O. Kuhn. Dynamic vehicle routing problem for medical emergency management. In J. I. Mwasiagi, editor, *Self Organizing Maps - Applications and Novel Algorithm Design*, pages 233–250. 2011.

R. Z. Farahani, N. Asgari, N. Heidari, M. Hosseininia, and M. Goh. Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62(1):368–407, 2012.

J. A. Fitzsimmons and B. N. Srikar. Emergency ambulance location using the contiguous zone search routine. *Journal of Operations Management*, 2(4):225–237, 1982.

A. Garner, A. Lee, K. Harrison, and C. H. Schultz. Comparative analysis of multiple-casualty incident triage algorithms. *Annals of emergency medicine*, 38(5):541–548, 2001.

M. Gendreau, G. Laporte, and F. Semet. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Computing*, 27(12):1641 – 1653, 2001.

T. A. Gennarelli and E. Wodzin, editors. *The Abbreviated Injury Scale 2005. Update 2008.* American Association for Automotive Medicine (AAAM), Barrington, Illinois, 2008.

G. Ghiani, F. Guerriero, G. Laporte, and R. Musmanno. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 151(1):1–11, 2003.

R. Goldberg and P. Listowsky. Critical factors for emergency vehicle routing expert systems. *Expert Systems with Applications*, 7(4):589–602, 1994.

K. Helsgaun. *An Effective Heuristic Algorithm for the Traveling-Salesman Problem.* Number 81. Datalogiske Skrifter, Roskilde University, 1998.

K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

K. Helsgaun. *An Effective Implementation of K-opt Moves for the Lin-Kernighan TSP Heuristic.* Number 109. Datalogiske Skrifter, Roskilde University, 2006.

J. Holguín-Veras, M. Jaller, L. N. Van Wassenhove, N. Pérez, and T. Wachtendorf. On the unique features of post-disaster humanitarian logistics. *Journal of Operations Management*, 30(7-8):494–506, 2012.

M. Huang, K. Smilowitz, and B. Balcik. Models for relief routing: Equity, efficiency and efficacy. *Transportation Research Part E: Logistics and Transportation Review*, 48(1): 2–18, 2012.

Ilog. IBM Ilog CPLEX Optimizer. `http://www.ilog.com/products/cplex/`, 2013. accessed 7.12.2013.

A. Jotshi, Q. Gong, and R. Batta. Dispatching and routing of emergency vehicles in disaster mitigation using data fusion. *Socio-Economic Planning Sciences*, 43(1):1–24, 2009.

R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum, New York, 1972.

J. P. Killeen, T. C. Chan, C. Buono, W. G. Griswold, and L. A. Lenert. A wireless first responder handheld device for rapid triage, patient assessment and documentation during mass casualty incidents. In *AMIA annual symposium proceedings*, volume 2006, pages 429–433. American Medical Informatics Association, 2006.

V. A. Knight, P. R. Harper, and L. Smith. Ambulance allocation for maximal survival with heterogeneous outcome measures. *Omega*, 40(6):918–926, 2012.

S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

M. Najafi, K. Eshghi, and W. Dullaert. A multi-objective robust optimization model for logistics planning in the earthquake response phase. *Transportation Research Part E: Logistics and Transportation Review*, 49(1):217–249, 2013.

M. Najafi, K. Eshghi, and S. de Leeuw. A dynamic dispatching and routing model to plan/ re-plan logistics activities in response to an earthquake. *OR Spectrum*, 36(2):323–356, 2014.

L. Özdamar and O. Demir. A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):591–602, 2012.

K. Panchamgam, Y. Xiong, B. Golden, B. Dussault, and E. Wasil. The hierarchical traveling salesman problem. *Optimization Letters*, 7(7):1517–1524, 2013.

S. N. Parragh. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies*, 19 (5):912–930, 2011.

S. N. Parragh, J.-F. Cordeau, K. F. Doerner, and R. F. Hartl. Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR Spectrum*, 34 (3):593–633, 2012.

A. J. Pedraza-Martinez and L. N. van Wassenhove. Transportation and vehicle fleet management in humanitarian logistics: challenges for future research. *EURO Journal on Transportation and Logistics*, 1(1-2):185–196, 2012.

V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

S. Rath and W. J. Gutjahr. A math-heuristic for the warehouse location-routing problem in disaster relief. *Computers & Operations Research*, 42(1):25–39, 2014.

M. Schilde, K. F. Doerner, and R. F. Hartl. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & Operations Research*, 38(12):1719–1730, 2011.

V. Schmid. Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European Journal of Operational Research*, 219(3): 611–621, 2012.

V. Schmid and K. F. Doerner. Ambulance location and relocation problems with time-dependent travel times. *European Journal of Operational Research*, 207(3):1293–1303, 2010.

M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

H. Toro-Díaz, M. E. Mayorga, S. Chanta, and L. A. McLay. Joint location and dispatching decisions for emergency medical services. *Computers & Industrial Engineering*, 64(4): 917–928, 2013.

L. E. de la Torre, I. S. Dolinskaya, and K. R. Smilowitz. Disaster relief routing: Integrating research and practice. *Socio-Economic Planning Sciences*, 46(1):88–97, 2012.

F. Wex, G. Schryen, S. Feuerriegel, and D. Neumann. Emergency response in natural disaster management: Allocation and scheduling of rescue units. *European Journal of Operational Research*, 235(3):697–708, 2014.

P. Yi, S. K. George, J. A. Paul, and L. Lin. Hospital capacity planning for disaster emergency management. *Socio-Economic Planning Sciences*, 44(3):151–160, 2010.

W. Yi and L. Özdamar. A dynamic logistics coordination model for evacuation and support in disaster response activities. *European Journal of Operational Research*, 179(3):1177–1193, 2007.