Fast Post-Disaster Emergency Vehicle Scheduling

Roberto Amadini[1], Imane Sefrioui[2], Jacopo Mauro[1], and Maurizio Gabbrielli[1]

Abstract Disasters like terrorist attacks, earthquakes, hurricanes, and volcano eruptions are usually unpredictable events that affect a high number of people. We propose an approach that can be used as a decision support tool for a post-disaster response that allows the assignment of victims to hospitals and organizes their transportation via emergency vehicles. Exploiting Operational Research and Constraint Programming techniques we are able to compute assignments and schedules of vehicles that save more victims than heuristic based approaches.

1 Introduction

Disasters are unpredictable events that demand dynamic, real-time, effective and cost efficient solutions in order to protect populations and infrastructures, mitigate the human and property loss, prevent or anticipate hazards and rapidly recover after a catastrophe. Terrorist attacks, earthquakes, hurricanes, volcano eruptions etc. usually affect a high number of people and involve a large part of the infrastructures thus causing problems for the rescue operations which are often computationally intractable. Indeed, these problems have been tackled by using a plethora of different approaches and techniques, ranging from operational research to artificial intelligence and system management (for a survey please see [3]).

Emergency response efforts [15] consist of two stages: pre-event responses that include predicting and analyzing potential dangers and developing necessary action plans for mitigation; post-event response that starts while the disaster is still in

Department of Computer Science and Engineering/Lab. Focus INRIA
University of Bologna, Italy.
e-mail: {amadini,jmauro,gabbri}@cs.unibo.it

Information and Telecommunication Systems Laboratory
Faculty of Sciences, Abdelmalek Essaadi University, Tetuan, Morocco
e-mail: sefrioui.imane@gmail.com

progress. At this stage the challenge is locating, allocating, coordinating, and managing available resources.

In this paper we are concerned with post-event response. We propose an algorithm and a software tool that can be used as a decision support system for assigning the victims of a disaster to hospitals and for scheduling emergency vehicles for their transportation. Even though our algorithm could be used to handle daily ambulance responses and routine emergency calls, we target specifically a disaster scenario where the number of victims and the scarcity of the means of transportation are usually overwhelming. Indeed, while for normal daily operations the ambulances can be sent following the order of the arrival of emergency calls, when a disaster happens this First In First Out policy is not anymore acceptable, since the number of victims involved and the quantity of damages require a plan and a schedule of rescue operations, where usually priority is given to more critical cases, trying in any case to maximize the number of saved people. In this context there are clearly also essential ethical issues which we do not address in this paper (for example, is ethically acceptable not to save a person immediately if this behavior allow us to save two people later on?).

Our tool assumes a simplified scenario where the number, the position and the criticality of victims is known. The tool compute solutions that try to maximize the global number of saved victims. In many practical cases, finding the optimal solution in not computationally feasible, hence we use a relaxation of the pure optimization problem. Our approach uses a *divide-et-impera* technique that exploits both Mixed Integer Programming (MIP) and Constraint Programming (CP) in order to solve the underlining assignment and scheduling problem.

To evaluate the effectiveness of our approach we have compared it against a baseline greedy approach based on the heuristic that sends the ambulances first to the most critical victims and later on to the others. Empirical results based on random generated disaster scenarios show that our approach is promising: it is able to compute the schedule usually in less than a minute and almost always save more victims than the greedy approach.

The remaining of this paper is organized as follows. In Section 2 we define the model we are considering while in Sections 3 and 4 we present the algorithm and the test we have conducted. In Section 5 we present some related work. We conclude giving some directions for future work in Section 6.

2 Model

In the literature a lot of models have been proposed to abstract from a concrete disaster scenario. Some of them are extremely complex and involve a lot of variables or probability distributions [6, 7]. For the purposes of this paper we adapt one of the simplest models, following [5], which considers only three entities: victims, hospitals, and ambulances. Thus we assume to have the following data:

- the spatial coordinates (a two dimensional value) of every entity;
- the capacity of ambulances and hospitals;

- an estimate of the time to death of every victim;
- the dig-up time of every victim, i.e the time needed by the rescue team to be able to rescue the victim as soon as the ambulance arrives on the spot;
- the time needed by an ambulance to reach the hospitals or the victims;
- the initial time an ambulance becomes available (an ambulance may be dismissed or already busy when the disaster strikes).

We are well aware that, especially in a disaster scenario, these data may be difficult to retrieve, imprecise and unreliable. Nevertheless our model can use these data to compute a first solution and then later, when the information become more precise, it can be rerun to improve the computed solution. Moreover, in order to get these informations one can use the results of such works like [11, 13, 14] that allow to esteem the time to death of a civilian or to find the best routes to reach the victims.

Assuming that all the above information are known, our goal is then to find an optimal scheduling of the ambulances in order to bring the maximal number of (alive) victims to the hospitals.

3 Procedure

Solving optimally the scheduling of ambulances may be computationally unfeasible, especially in case of a large number of victims. Moreover, in our scenario, a fast response of the scheduling algorithm is important for different reasons. First of all, the quicker the response is, the faster we can move the ambulances and therefore more victims may be saved. Moreover, waiting for a long time may be useless because usually information rapidly changes (i.e. more victims come, the criticality of the patients vary, the hospitals may have damages or emergencies, ambulances can be broken). Hence, spending a lot of time for computing an optimal solution that in few seconds could become non optimal may result in a waste of resources and then lead to the impossibility of saving some victims. On the other hand, a purely greedy approach that at each stage makes the locally optimal choice (according to heuristics such as the seriousness or the location of the victims) would be definitely faster, but could result in a smaller global number of victims saved.

For these reasons we developed an approach that at the same time allows to compute a solution within a reasonable time limit and still allows us to save more victims than greedy strategies. Our approach basically lies in the interaction of two phases: the *allocation phase*, in which we try to allocate as many victims as possible to ambulances and hospitals, and the *scheduling phase*, in which we compute the path that each ambulance must follow in order to bring the victims to the hospitals.

In the allocation phase, we relaxed some constraints of the problem assuming that every ambulance can save in parallel all the victims it contains (in other terms, each ambulance with capacity c can be seen as the union of c distinct ambulances with capacity 1). The allocation of every victim to an ambulance and a hospital is performed by solving a Mixed Integer Programming problem. More precisely, the constraints that we enforced are the following:

- a victim can not be assigned to more than one ambulance and hospital;

- the number of patients on a given ambulance and in a hospital must not exceed its capacity;
- a victim assigned to an ambulance is also assigned to a hospital and vice versa;
- the time an ambulance needs to reach a victim, dig up and bring her to a hospital, is enough to save the victim.

Since the objective of the MIP problem is to be able to maximize the number of rescued victims, we defined an objective function which takes into account both the seriousness and the location of the victims. Solving this problem gives an esteem of the victims that could be saved and a preliminary allocation of every victim to an ambulance and a hospital. It is worth noticing that since this is a relaxation of the original problem, it may be possible that not all the victims allocated to an ambulance may be saved. Anyway, the allocation guarantees that at least one victim for ambulance can be rescued. Also, there are no restrictions on the number of hospitals that an ambulance can visit.

Once the victims have been allocated by the first phase, the scheduling phase allows to define the path that each ambulance must follow in order to maximize the number of victims saved. In other words, for each ambulance we have to solve a scheduling problem that can be seen as the problem of finding a minimal cost Hamiltonian path in a direct and weighted graph where:

- each node represents the spatial coordinates of one entity: either an ambulance, or a victim, or a hospital. In particular, the start node of the path has to be the one representing the start position of the ambulance, while the end node must represent the coordinates of an hospital.
- each arc represents the possibility of moving from one node to another, while its weight represents the estimated time to do so.

The resulting path must be constrained with respect to the time to death of each victim. The scheduling phase can therefore be mapped into a Constraint Optimization Problem (COP) and solved by using constraint programming techniques.

As already stated, it may be the case that not all the victims allocated to an ambulance may be saved, since differently to what happen in the relaxed problem now an ambulance has to save the victims sequentially. When this happens we have to compute a schedule that saves a maximal subset of such victims. However, instead of considering as maximal subset the one which contains the greater number of elements, we chose the one which has the maximum *priority value* which is calculated as follows. We first compute the remaining time (RT) of each victim by subtracting her dig-up time from the expected time to death. Then, given a subset of victims, we set its priority to the sum of the reciprocals of their RT (bigger values means higher priority). We decided to use this sum to evaluate the priority because, analogously to what happens for the harmonic average, the sum of the reciprocal gives priority to the victim having least RT and it mitigates at the same time the influence of large outliers (i.e. victims with big RT that can be easily saved later).

When an ambulance is scheduled, the model is updated and the allocation phase is possibly restarted in order to try to allocate the victims which have not yet been allocated. The procedure ends when no more victims can be saved.

From the computational point of view this procedure cyclically solves MIP and COP problems. We are aware that these are well known NP-hard problems. However, by exploiting the relaxation of the MIP problem, on one hand, and the limited size of the COP problems, on the other (the capacity of an ambulance is usually a small number) it is possible to get quickly optimal solutions by using current MIP and COP solvers.

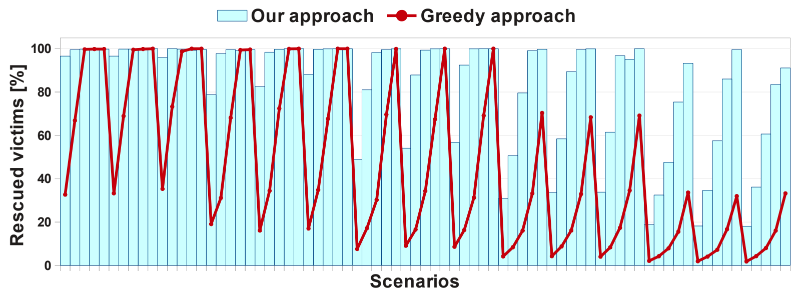
4 Tests Results

We did not find in the literature extensive benchmarks of disaster scenarios that we could use to evaluate and compare the performances of our approach. For this reason, in order to evaluate our approach, we built some random generated scenarios obtained by varying the number of hospitals in the set $\{1,2,4\}$, the number of ambulances in $\{4,8,16,32,64\}$, and the number of victims in $\{32,64,128,256,512\}$. The position of each entity was randomly chosen in a grid of 100×100 by using the euclidean distance to estimate the time needed for moving from one point to another. The capacities of the ambulances and the hospitals were selected randomly in the intervals $[1..4]$ and $[300..1000]$, respectively, while the dig-up time and the time to death of every victim were randomly chosen in $[5..30]$ and $[100..1000]$, respectively.

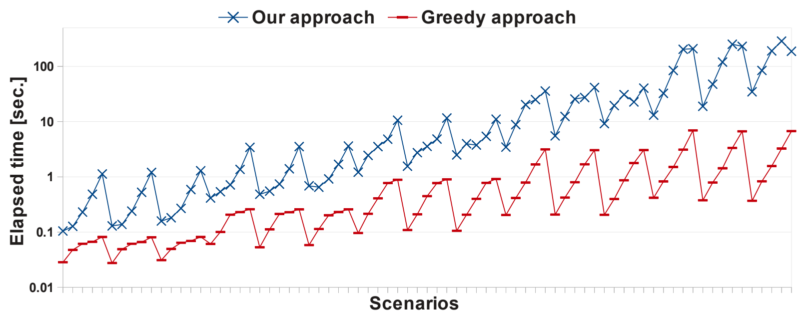
To increase the accuracy and the significance of the results we tested our approach by running the experiments 20 times for each different scenario and measuring the average number of rescued victims as well as the time required to solve the problem. For every scenario we compared the results obtained with those obtained by running a greedy algorithm that at each time assigns the most critical victims to the closest available ambulance and then the ambulance to the closest available hospital. In total we tested 75 different scenarios.

Fig. 1a shows, for each scenario, the average percentage of rescued victims obtained by using our approach and the greedy one. The `x`-axis represents the scenarios sorted lexicographically by increasing number of ambulances, victims and hospitals (labels are omitted for the sake of readability, since each `x`-value is actually a triple of values). In only 4 cases (5.3% of the scenarios) the greedy algorithm is better than our approach. However in this few cases the difference of saved victims is less than 1% while the gap between the greedy approach and our approach can reach peaks of about 80%. In average our approach is able to solve 37.32% of more victims than the greedy approach. From the plot we can also see that our approach is especially better for scenarios involving a large number of victims. Indeed in these case the greedy algorithm usually makes local choices that have a huge impact on the total number of the victims that could be saved. On the other hand, our approach in these cases tries to come up with a better global choice and therefore it can be far superior than the greedy strategy.

In Fig. 1b we show the time needed to compute the entire schedule of the ambulances (please note the logarithmic scale). Although it is not surprising that the greedy approach uses less time, it can however be observed that our approach takes reasonable times. In fact, in average the ambulances are allocated in 32.34 seconds,



(a) Average percentage of rescued victims.



(b) Average computation time.

Fig. 1: Test results.

which means that on average in little more than half a minute all the ambulances will be able to know the path that should be followed.

All the experiments were done by using an Intel®Core™ 2.93 GHz computer with 6 GB of RAM and Ubuntu operating system. We used Gurobi optimizer for solving the MIP problems and Gecode solver for the COP problems. The code developed to conduct the experiments, the experimental results and technical details are available at `http://www.cs.unibo.it/~amadini/dcai_2013.zip`

5 Related work

In the literature many techniques from operational research and artificial intelligence have been used to tackle different aspects of the disaster management problem. Most approaches are trying to develop and study pre-event solution to decrease the severity of the disaster outcome. As an example, in [4] the authors study the best allocation of deposits that allows to handle in the most efficient way the rescue operations in case natural disaster happens. In [7] the authors use MIP in order to schedule the operation rooms and the hospital facilities in case of a disaster. These paper however have a different goal from ours: we are not concerned with considering preventing measures that could allow to mitigate the consequences of future disasters. We are instead concerned with saving more victims, after the disaster happened.

There is a large literature also related to the problem of deciding the initial location of ambulances in order to decrease the average response time for ambulance calls. However, very few papers deal with the computation of the schedule for an ambulance. Some authors focus just on computing the best path for an ambulance toward the victim. For instance in [11] the authors use graph optimization algorithms in order to find a path for an ambulance while in [9] a multi-agents system is used to retrieve the best route for organ transportation. In our work we assume to have such a path and we are concerned with the problem of defining the order of the victims that an ambulance should pick up. In [8, 16] the authors propose a routing algorithm for ambulances but, differently from our case, their model is probabilistic.

In [5] the authors proposed the use of an interactive learning approach which allows rescue agents to adapt their preferences following strategies suggested by experts. The decision of the ambulance is based on a utility function incrementally improved through expert intervention. Differently from our approach the authors here use an heuristic to dispatch the ambulances which rely on expert decision makers, while we rely only on optimization techniques.

In [12] is solved a task scheduling problem in which rescuing a civilian is considered as a task and the ambulances are considered as resources that should accomplish the task. The goal is to perform as many tasks as possible by using the Hogdson's scheduling algorithm to compute the solutions. Differently from our case, the authors considered here only the execution cost of the task and its deadline, ignoring important constraints such as the capacity of the hospitals and ambulances.

The authors in [10] proposed a model based on a Multi-Objective Optimization Problem. They adjust controllable parameters in the interaction between different classes of agents (hospitals, persons, ambulances) and resources, in order to minimize the number of casualties, the number of fatalities, the average ill-health of the population, and the average waiting time at the hospitals. Then, they use Multi-Objective Evolutionary algorithms for producing good emergency response plans. Their underline model is completely different from ours and we argue that is not very adaptable to deal with continuous changes and unexpected situations.

Finally we are aware of the existence of commercial application for Emergency Dispatching (e.g. [1, 2]). The technical details explaining how these software are working unfortunately are always missing.

6 Conclusions

In this work we have described a procedure that can be used as a decision support tool for a post-disaster event when a big number of victims need to be transported to the hospitals. The proposed algorithm takes into account the position of the victims and their criticality, and schedule the ambulance to maximize the number of saved victims. Even though there is no guarantee that the solution obtained is the optimal one, experimental tests confirm that the number of saved victims is greater than the one that could be obtained by using a greedy, priority based heuristic. Moreover the proposed solution is usually fast enough to assign all the available ambulances in less than a minute.

As a future work we are planning to evaluate our approach in a more dynamic scenario. In particular we are planning to develop a discrete event simulator that will allow the comparison of our decision support tool with stochastic approaches such as the one described in [6, 10] or other disaster scenario models such as [12]. Moreover we would like to integrate the model with heuristics developed by domain experts. Adopting these heuristics will allow the system to be able to react to a change very quickly, by using a default behavior that later can be changed if a better solution is found solving the optimization problem. Another direction worth investigating is to study the performance and the scalability of the algorithm proposed taking into account also the robustness of the solutions.

References

1. GeoFES website: `http://www.dhigroup.com/MIKECUSTOMISEDbyDHI/GeoFES.aspx`.
2. Odyssey website: `http://www.plain.co.uk/index.php?option=com_content&task=view&id=67&Itemid=98`.
3. N. Altay and W. G. Green. OR/MS research in disaster operations management. *EJOR*, 175(1):475–493, 2006.
4. T. Andersson, S. Petersson, and P. Värbrand. *Decision Support for Efficient Ambulance Logistics*. Department of Science and Technology (ITN), Linköping University, 2005.
5. T.-Q. Chu, A. Drogoul, A. Boucher, and J.-D. Zucker. Interactive Learning of Independent Experts' Criteria for Rescue Simulations. *J. UCS*, 15(13):2701–2725, 2009.
6. G. Erdogan, E. Erkut, A. Ingolfsson, and G. Laporte. Scheduling ambulance crews for maximum coverage. *JORS*, 61(4):543–550, 2010.
7. N. Issam, N. Jean-Christophe, and D. Jolly. Reactive Operating Schedule in Case of a Disaster: Arrival of Unexpected Victims. WCE, pages 2123–2128, 2010.
8. B. Lopez, B. Innocenti, and D. Busquets. A Multiagent System for Coordinating Ambulances for Emergency Medical Services. *Intelligent Systems*, 23(5):50 –57, 2008.
9. A. Moreno, A. Valls, and J. Bocio. Management of Hospital Teams for Organ Transplants Using Multi-agent Systems. AIME, pages 374–383, 2001.
10. G. Narzisi, V. Mysore, and B. Mishra. Multi-Objective Evolutionary Optimization of Agent Based Models: an application to emergency response planning. Computational Intelligence, pages 228–232, 2006.
11. N. Nordin, N. Kadir, Z. Zaharudin, and N. Nordin. An application of the A* algorithm on the ambulance routing. CHUSER, pages 855 –859, 2011.
12. S. Paquet, N. Bernier, and B. Chaib-draa. Multiagent Systems Viewed as Distributed Scheduling Systems: Methodology and Experiments. AI, pages 43–47, 2005.
13. S. A. Suárez, C. G. Quintero, and J. L. de la Rosa. A Real Time Approach for Task Allocation in a Disaster Scenario. PAAMS, pages 157–162, 2010.
14. M. Teruhiro, S. Weihua, Y. Keiichi, and I. Minoru. Transportation Scheduling Method for Patients in MCI using Electronic Triage Tag. eTELEMED, pages 1–7, 2011.
15. S. Tufekci and W. Wallace. The Emerging Area Of Emergency Management And Engineering. *Engineering Management*, 45(2):103 –105, 1998.
16. K. Ufuk, T. Ozden, and T. Saniye. Emergency Vehicle Routing in Disaster Response Operations. POMS, 2012.