

# Decision Modeling for Assignments of Collaborative Rescue Units during Emergency Response

Felix Wex  
University of Freiburg  
[felix.wex@is.uni-freiburg.de](mailto:felix.wex@is.uni-freiburg.de)

Guido Schryen  
University of Regensburg  
[guido.schryen@wiwi.uni-regensburg.de](mailto:guido.schryen@wiwi.uni-regensburg.de)

Dirk Neumann  
University of Freiburg  
[dirk.neumann@is.uni-freiburg.de](mailto:dirk.neumann@is.uni-freiburg.de)

## Abstract

*Decision support systems play an increasingly important role in disaster management research. Coordination of rescue units during disaster response is one of the many areas which may benefit from this development. Time pressure, resource shortages, different capabilities of rescue units and the interdependence of scheduling and allocation tasks belong to the key challenges which emergency operation centers have to cope with. This paper proposes a non-linear optimization model and suggests a Monte Carlo-based heuristic solution procedure. We computationally benchmark our heuristic with a procedure that is applied in practice. Results of our study show that the Monte-Carlo heuristic is superior to the state-of-the art approach in terms of aggregated harm by up to 40%. However, our simulations also reveal that the time our heuristic needs to process medium-sized instances (100 incidents, 50 rescue units) on a PC is a few hours and that more powerful real-time computing capabilities are required.*

## 1. Introduction

Natural disasters, including earthquakes, Tsunamis, floods, hurricanes, and volcanic eruptions, have caused tremendous harm and continue to threaten millions of humans and various infrastructure capabilities each year. For example, according to the World Disaster Report of the International Federation of Red Cross and Red Crescent Societies [19], the megathrust earthquake centered near Sumatra on December 26, 2004, generated a tsunami that resulted in more than 220,000 deaths, the tropical cyclone Nargis on May 2, 2008, led to almost 140,000 deaths, and the Haiti earthquake on January 12, 2010 caused more than 220,000 deaths. Over all natural disasters within the period 2000-2009, the estimated number of people killed amounted to almost 1 million and the

estimated economic damage caused by natural disasters was calculated to almost US\$ 1,000 billion, respectively.

Immediate consequences of mid- to large-scale natural disasters (e.g. superregional earthquakes) can often be characterized by (a) an unknown large number of incidents (casualties, damage), (b) multiple, differently skilled rescue teams sent from all over the world, and (c) severe time constraints due to finite rescue times and ever-changing situations. The hidden challenge of natural disaster management (NDM) is to accept, and ideally to be prepared for, these characteristics by satisfying the special needs that are imposed by the set of incidents. This study attempts to tackle these characteristics by deterministically investigating different sets of scenarios each with different numbers of incidents and rescue units.

We use the term “incident” as a proxy for all synonyms indicating any immediate event of damage or loss caused by a natural disaster or its harmful consequences.

According to the literature [2, 10, 18, 19, 27], challenges and activities of natural disaster management can be classified along the pre-disaster phase (preparedness), the during-disaster phase (response), and the post-disaster phase (recovery) which can be arranged in a life-cycle [10]. In this paper, we focus on the response phase of NDM. Effective and efficient coordination efforts during emergency response are regarded as one of the critical tasks for emergency operations centers (EOCs). This fundamental challenge imposed on commanders is typically aggravated due to the lack of centralized command structure, which results from the involvement of many heterogeneous aid organizations, such as the Red Cross, technical relief organizations, and national guards [25]. In practice, the involvement of these organizations with different cultural backgrounds, disaster response policies, resources, and capabilities entails a distributed planning and implementation of response actions. It is not

astonishing that this organizational patchwork results in overall inefficient disaster response operations and redundancies in commands.

Some of the above characteristics in relief management efforts were apparent after catastrophes in the recent past (e.g. Haiti 2010, Chile 2010, and Japan 2011). Surprisingly, this sometimes leads to the suspicion that the coordination of rescue units during these large-scale emergencies is an even bigger problem than resource scarcity over all. Interviews with the German Federal Agency for Technical Relief (THW) approved this. The interviewees also revealed another factor when it comes to the coordination of resources during emergency response: (human) command is often either communicated redundantly or counteractively in an improvised and decentralized manner, which makes it difficult for rescue units to follow the right command and execute it reliably and in a timely manner. What is currently missing in practice are ways to make command processes in chaotic (large-scale) settings even more reliable. This may possibly be achieved by avoiding the shortcomings from above and enforcing a centralized command structure.

Coordination tasks can be split into operational and tactical procedures such as scheduling and the allocation of resources. We define both as most critical research issues in this paper. This is not only due to the underlying information that non-computer based coordination is currently done by experienced human reasoning. Yet, we question this expertise for large-scale scenarios when chaos and the pressure on individual commanders rise and dozens of incidents are confronted to a limited number of rescue teams, accounting for the necessity to co-allocate. This hypothesis is based on our assumption that computer-based heuristics may (a) improve human reasoning in small-scale scenarios by strictly obeying optimality criteria and (b) provide decision support even in more complex large-scale settings where human reasoning is naturally restricted.

In this study, we address the coordination problem during emergency response and propose a decision support system to assist in scheduling and assigning rescue units to incidents. We address this objective by suggesting a quantitative optimization model and one possible solution heuristic.

In the modeling process, we assume that harm can be reduced by minimizing overall completion times of incidents, weighted by the severity of incidents. Assuming that decision support systems may be notably useful in complex settings when human apprehension is finite, the solution of the optimization model may act not only as a research contribution but also as a decision support for decision-makers in

practice. Disaster-specific characteristics such as differences in severity levels between incidents, distances, processing times, and different kinds of incidents find reflection in the model. We define a benchmark heuristic, which mirrors decision reasoning by today's human commanders, to evaluate the quality of the performance of the proposed solution heuristic.

The paper is structured as follows: Section 2 presents requirements, which follow from a literature review. These requirements are subsequently integrated into the mathematical modeling process in section 3. Section 4 introduces the data environment and describes the experiments conducted. Section 5 evaluates the experimental results, which give insights into runtimes. The paper closes with a conclusion and an outlook into future work.

## 2. Literature Review and Requirements

As this paper focusses on the response phase of NDM, we present a literature review only of this phase; a literature overview of the preparedness phase and the recovery phase is provided in [31] and [26], for example. The methodology of our literature review is presented in appendix A.

Altay and Green III (2006) [4] accentuate a strong need for novel theory and methodology by the IS community (among others). Open issues include the design of organizational and network structures that facilitate communications and coordination in disaster response, and solutions to logistical problems in all phases of NDM.

It was interesting to see that most strands engage in information, communication systems, infrastructure [6, 9, 11, 13, 17, 21, 28], and management [1, 7], but less in decision support methodology. Some of the latter are subsequently introduced.

One of the many decision support streams we found combines methods from applied statistics and probability theory with mathematical programming approaches to establish novel codes of conduct and metrics that assist any commander in those critical minutes of the decision-making process [12, 23]. Competitive mechanisms (e.g. auctions) and cooperative mechanisms (e.g. multi-criteria approaches) are suggested. Another research stream follows guidelines from computational intelligence research [20, 29] to bridge the gap between information system design principles and decision support process architectures. A third group of researchers makes use of empirical investigations of past decision-making conclusions to establish innovative courses of action [15]. A fourth research

stream focuses on the decision-making process based on decentralized agents, e.g. Fiedrich et al. (2000) [16] introduce the usage of optimization modeling. The authors above discuss pros and cons of centralized versus distributed decision authorities. Distributed coordination (assignments and schedules) may remain dependent from redundancies and miscommunications whereas centralized instances (EOCs) may effectively enforce commands if essential infrastructure capabilities exist and EOC communication is prioritized. On the other hand, centralized command may act autonomously within closed operational areas. The possibility that several operational areas are located right next to each other exists with an equivalent number of centralized EOCs. Due to the above weaknesses of decentralized coordination, this study focusses on centralized command structures.

The authors of [14] propose an optimization model for scheduling volunteers during emergency response where the subjects feature time windows. Another paper [24] promotes centralized coordination by applying a mathematical programming model for scheduling distributed rescue units and the assignments of incidents to these units. However, the suggested model uses time periods of fixed length, and does not account for the fact that incidents may have different levels of severity. Wex et al. [31] introduce an optimization model in a centralized way that matches incidents by clearly assigning a single rescue unit per incident. Fuzzy optimization is used in [32] in order to handle the high level of informational uncertainty that occurs during any emergency. All of the above centralized optimization models lack the eventuality that rescue units need to be assigned to incidents collaboratively.

We hence deduce that the research objective has been understudied in former scholarship so far. The literature review together with interviews with practitioners (THW) led to a distinct set of requirements. The artifact, which we propose in the next section is being sought to fulfill the following six requirements in order to solve the problem of efficiently and effectively coordinating rescue units to incidents:

1. Timeliness in decision provisioning
2. Autonomy of centralized decision-makers
3. (In-)Completeness of centralized information
4. Heterogeneous rescue units and incidents
5. Non-preemptiveness
6. Ability to co-allocate rescue units to an incident

### 3. Decision Model

As we are examining modeling approaches of a real world scenario, which is both a scheduling and an assignment problem, we screened relevant literature on the multiple traveling salesman problem (mTSP) and from job scheduling theory. [5] proposes modeling variants and solution procedures for the mTSP. Yet, our scheduling and assignment problem is only related to the mTSP in terms of constraints but varies significantly regarding the objective function because of dependencies between processing sequences. That is, it does make a difference for the overall harm to process an incident before a less severe incident.

Our problem is also related to a problem in the scheduling literature. If we assume that travel times between two incident locations does not depend on the particular type of rescue unit that travels, then our problem is equivalent to the “parallel-machine scheduling problem with unrelated machines, non-batch sequence-dependent setup times, and a weighted sum of completion times as the objective”, classified as  $R/ST_{sd}/\sum w_j C_j$  in the scheduling literature [3]. However, this assumption is rarely met in practice so that heuristics suggested for this problem [30] are inappropriate. Thus, we are bound to alternative solution heuristics.

Our artifact, in terms of a quantitative decision model, is modeled as a variant of a (job) scheduling model for unrelated, parallel machines (rescue units) [8]. The model is non-preemptive (Requirement 5): a rescue unit cannot interrupt processing an incident (job) before its complete release. We hereby allow for parallel processing of one incident by several rescue units. An incident is not regarded as being completely processed unless all required rescue units have finished their work. But, once a rescue unit has finished a job it can be assigned to another incident again. Furthermore, we do not require specific processing orders (task windows). All relevant information (processing times, severity of incidents, and travel times) is expected to be available in order to make the model work. Even though this may seem unrealistic, we assume that we can trust reports from on-site agents about incidents and status updates of rescue units and regard information as complete. In cases where uncertainty prevails, we refer to a non-probabilistic, fuzzy optimization model presented in [32] even though this model is not able to co-allocate rescue units. Using probabilistic factors or fuzzy numbers as proxies for uncertainty would also imply other challenges, such as appropriate parameter settings, applicability, interpretation value, and an increase in model complexity.

We thus explicitly introduce our model for centralized coordination within clearly defined operational areas. In a superregional disaster, we assume to implement our model in n-decentralized areas given that the autonomous command zones have clear boundaries.

In cases when the disaster itself is very confusing and the situation is changing continuously (e.g. updates and new incidents are continuously reported), we abort the current optimization process and restart it with the new parameters (continuous planning property). On the other hand, all tasks of rescue units which are already processing incidents or sent out to do so cannot be aborted if the optimization is started anew once any scenario has altered. That is, rescue units can only be assigned to new incidents when they become idle.

Besides the fulfillment of previously established requirements, this binary, non-linear optimization model pursues two goals: (1) generation of valid schedules and assignments for rescue units; (2) minimization of the total harm occurring during the scene. We assume that harm can reasonably be modeled by the sum of completion times over all incidents multiplied by weighting factors that account for their destructiveness. The model especially accounts for co-allocation which appears when incidents require various, differently-skilled rescue personnel and punishes waiting times that occur when incidents are not processed immediately after their appearance.

The objective function seeks to minimize total weighted completion times which are necessary to process all incidents  $j$ . Schedules and assignments are generated by two binary decision variables  $X_{ij}^k$  and  $Y_{ij}^k$ , which indicate if an incident  $i$  is an immediate predecessor of  $j$  or a mediate predecessor in the list of incidents that are processed by rescue unit  $k$ , respectively. A weighing factor  $w_j$  is introduced which depicts the level of severity of incident  $j$ . For the parameterization of factors  $w_j$ , we make use of the classification introduced by the U.S. Department of Homeland Security (2008) which distinguishes between different (terrorism) alert levels. Other parameters in use are: processing times  $p_j^k$  which denote how much time rescue unit  $k$  requires to process incident  $j$ . Travel times  $s_{ij}^k$  measure the time needed for rescue unit  $k$  to move from the location of incident  $i$  to the location of incident  $j$ . We introduce two fictitious incidents '0' and 'n+1' for technical modeling reasons, where using incident 0 allows for considering the depots (starting locations) of rescue units ( $p_0^k = p_{n+1}^k = 0, k = 1, \dots, m; s_{i(n+1)}^k = 0, i = 0, \dots, n, k = 1, \dots, m$ ).

$cap_{k,l}$  is a binary parameter with  $cap_{k,l}=1$  if and only if rescue unit  $k$  has capability  $l$  (e.g. firemen, paramedics). Our modeling also provides for those situations in which a rescue unit can have more than one capability. The binary parameter  $cat_{i,l}$  equals 1 if and only if the processing of incident  $i$  requires characteristics of rescue units which have to be matched by rescue units' capabilities. This explicitly includes the case that an incident requires the capabilities of more than one rescue units. To sum up, both relationships (rescue\_units[capabilities] and incidents[capabilities]) are of type (m:n).

$$\min \sum_{j=1}^n w_j \left( \sum_{i=0}^n \sum_{k=1}^m \left[ p_i^k Y_{ij}^k + (p_j^k + s_{ij}^k) X_{ij}^k + Y_{ij}^k \left( \sum_{l=0}^n X_{il}^k s_{li}^k \right) \right] \right) \quad (C0)$$

$$\text{s.t.} \quad \sum_{i=0}^n X_{ij}^k \leq 1, \quad j = 1, \dots, n; k = 1, \dots, m \quad (C1)$$

$$\sum_{j=1}^{n+1} X_{ij}^k \leq 1, \quad i = 1, \dots, n; k = 1, \dots, m \quad (C2)$$

$$\sum_{j=1}^{n+1} X_{0j}^k = 1, \quad k = 1, \dots, m \quad (C3)$$

$$\sum_{i=0}^n X_{i(n+1)}^k = 1, \quad k = 1, \dots, m \quad (C4)$$

$$Y_{il}^k + Y_{lj}^k - 1 \leq Y_{ij}^k, \quad \begin{aligned} & i = 0, \dots, n; j \\ & = 1, \dots, n+1; k = 1, \dots, m; l \\ & = 1, \dots, n \end{aligned} \quad (C5)$$

$$\sum_{i=0}^n X_{il}^k = \sum_{j=1}^{n+1} X_{ij}^k, \quad l = 1, \dots, n; k = 1, \dots, m \quad (C6)$$

$$X_{ij}^k \leq Y_{ij}^k, \quad \begin{aligned} & i = 0, \dots, n; j = 1, \dots, n+1; k \\ & = 1, \dots, m \end{aligned} \quad (C7)$$

$$Y_{il}^k = 0, \quad i = 0, \dots, n+1; k = 1, \dots, m \quad (C8)$$

$$\sum_{j=1}^{n+1} \sum_{k=1}^m cap_{k,l} X_{ij}^k \geq cat_{i,l}, \quad \begin{aligned} & i = 1, \dots, n; l = 1, \dots, d \end{aligned} \quad (C9)$$

$$X_{ij}^k, Y_{ij}^k \in \{0,1\}, \quad \begin{aligned} & i = 0, \dots, n; j \\ & = 1, \dots, n+1; k = 1, \dots, m \end{aligned} \quad (C10)$$

$$cat_{i,l} \in \{0,1\}, \quad i = 1, \dots, n; l = 1, \dots, d \quad (C11)$$

$$cap_{k,l} \in \{0,1\}, \quad k = 1, \dots, m; l = 1, \dots, d \quad (C12)$$

$$w_j, p_j^k, s_{ij}^k \in \mathbb{R}^{\geq 0} \quad (C13)$$

Constraint (C1) ensures the correct alignment of immediate predecessor relationships between incidents that are processed successively by one specific rescue unit  $k$ ; (C2) addresses the immediate successor relationships analogously. Both constraints permit that an incident may be processed by more than one rescue unit (co-allocation) but prohibit that a rescue unit processes more than one incident at the same time.

Constraints (C3)–(C4) guarantee that rescue units start from their depot (fictitious incident ‘0’) and end in ‘n+1’ (fictitious incident ‘n+1’). (C5) declares that predecessor relationships are transitive. Additionally, if an immediate predecessor exists, there also has to be a successor (C6). (C7) indicates that any immediate predecessor is also a general predecessor. (C8), in conjunction with (C5) prohibits a reflexive, direct or indirect predecessor relationship. (C9) ensures that all capabilities required to process incident  $i$  are jointly covered by the rescue units that process incident  $i$ . In addition, the model still remains valid if rescue units possess more than one capability.

Trivially, (C10) defines the two binary decision variables and implies non-preemption. (C11), (C12), and (C13) define all other parameters used. In our sense, the so-called factor of destruction  $w_j$  represents, and is apt to model, the severity level of an incident. An explanation of how model instances are parameterized is presented in the next section.

Each feasible solution of the minimization model represents a valid schedule and assignment for all rescue units. We illustrate this in the exemplary scenario depicted in Figure 1. Two differently-skilled rescue teams face (at least) five incidents out of which only incident 4 requires the skills of both units (need for co-allocation). Incidents  $j_1, j_2 \in I$  both denote the last real incidents which need to be processed by the medical and the firefighting unit, respectively, before ending the process with fictitious incident n+1.

In detail, a schedule is proposed for the medical unit to process incident 1 before processing incidents 3 and 4 due to the above optimality criterion (order: 0-1-3-4-...- $j_1$ -(n+1)). The fire brigade would adopt an identical schedule vice versa (0-2-4-5-...- $j_2$ -(n+1)). Following such an approach would entail an objective value of ‘323’. In contrast, processing incident 3 immediately before incident 1 by the medical unit would result in a worse value of ‘328’. For reasons of clarity, incident 4 is regarded as uncovered until not all or parts of jobs have been finished, that is, until all collaboration units are done processing.

It is not astonishing that such an illustrative example evolves confusing – and thus more complex – when more incidents or rescue units are involved, especially under the premise that some incidents

require several capabilities of rescue units and others not.

**Proof of Complexity.** Our decision model is a generalization of the emergency response decision model suggested in [32]. The generalization lies in the fact that our model additionally allows for various capabilities per rescue unit and per incident. Since the model suggested in [32] is NP-hard, our model is NP-hard, too.

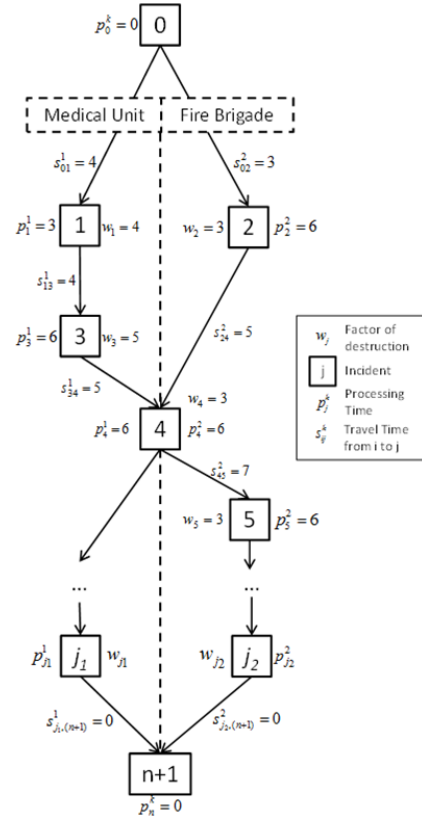


Figure 1. Example schedule and assignment for a medical and a firefighting unit.

## 4. Computational Evaluation

Due to the computational hardness and related computational inefficiency of the decision model, we suggest two heuristic approaches for solving model instances. We first describe the heuristics, then, we present our framework for evaluating the heuristic and the technical infrastructure of our simulation.

## 4.1. Heuristics

A Monte-Carlo based heuristic is suggested as one possibility to solve the above optimization model. Monte-Carlo is chosen for several reasons:

1. Our decision model is too complex (NP-hard) to be solved (optimally) in reasonable time.
2. We expect a high number of local optima. Deterministic heuristics might get stuck in these.
3. Monte Carlo allows to adapt runtimes by altering the number of its iterations

The key idea of generating a feasible solution in our Monte Carlo simulation is that incidents are iteratively scheduled in two stages: in stage one, an incident is assigned randomly to one of the  $D\%$  most appropriate rescue units, where appropriateness is determined based on the required capabilities (skills) and processing times. The motivation of this procedure is based on avoiding both a) assignments of incidents to units that require an extremely long time for processing (thus, a parameter  $D$  in  $[0; 100]$  is used), and b) myopic assignments of incidents to units that require the shortest processing time among all units (thus, randomness is included). If there is no rescue unit that has the capability to process the incident, the algorithm terminates unsuccessfully.

In stage 2, the chosen incident is inserted into the incident queue of the previously selected rescue unit. The criterion for determining the position of the new incident in the queue is based on a weighted ratio of the severity of incident  $w$  and the time  $p$  it takes the selected rescue unit to process this incident. Each queue lists its incidents in descending order of  $(w/p)$ -values. The algorithm terminates successfully if feasible solutions have been generated.

The Monte Carlo heuristic requires two input parameters:  $D \in [0; 100]$  is used for the selection of rescue units and the number of iterations which is the number of feasible solutions generated. We set  $D = 90$  and the number of iterations to 500,000 based on results of pretests. As initialization, the currently best solution value is set to infinity and the currently best solution is set to undefined, the current number of iterations is set to 0, the cumulated processing times are set to 0 for each rescue unit, the current incident queues are set to empty for each rescue unit, and we define  $I^*$  as the set of currently unassigned incidents. The incidents in  $I^*$  are now processed iteratively:

For all categories  $d \in L$ , we define  $K(d)$  as the set of all rescue units that are capable of processing category  $d$  required by incident  $i$ . If incident  $i$  cannot be classified by category  $d$ , we set  $K(d)=\{\}$  and

proceed. We rearrange all  $K(d)$  in ascending order of cumulative processing times. If there are not enough rescue units that possess the capabilities to completely process incident  $i$ , the algorithm terminates unsuccessfully. In each  $K(d)$ , the algorithm randomly selects a rescue unit with one of the  $D\%$  lowest cumulative processing times. The purpose of introducing this element of randomization is the avoidance of greedy assignments of units to incidents while contemporaneously avoiding assignments of rescue units with extremely high cumulative processing times. The cumulative processing time of the selected unit is then updated, which concludes stage 1. In stage 2, the current incident  $i$  is inserted into the queue of the selected rescue unit  $queue(unit)$  such that the queue is ordered in ascending order of values  $(fact\_destruct(i)/processing\_time(unit, i))$ , and incident  $i$  is removed from the set of incidents that still need to be assigned. If all incidents have been assigned and all required categories have been matched by rescue units' capabilities, then the current schedule is compared with the best known schedule, which is contingently updated. The algorithm terminates successfully if enough feasible solutions have been generated, equaling the number of iterations.

As another possible solution method to our model, we select a heuristic which can be found in practice, usually in a manually operated and non-automated decision-making processes. We gained information on this heuristic through interviews with the THW. The key ideas of the EOC heuristic are that a) incidents are assigned to rescue units in descending order of the factor of destruction, and b) that each incident  $j$  is assigned to those rescue units  $k$  that are (i) capable of processing incident  $j$  and (ii) that can start processing incident  $j$  at the earliest point of time, with assignment history and updated travel times being considered. That is, the heuristic computes schedules which arise when greedily assigning the most severe incident to the closest, idle rescue units. An incident is regarded as fully processed until all of its categories are completely matched by rescue units' capabilities.

We do not only assume that this approach can be found in practice but we also hypothesize that it can serve as a well-defined benchmark. In absence of lower bound solutions, the results of both heuristics build the basis for the evaluation of our proposed Monte Carlo based solution heuristic in the follow-up.

## 4.2. Experiment Setup

Due to the lack of real-time data we randomly generated different mid- to large-scale disaster scenarios: for each instance size, defined by the

number of incidents and rescue units, we generated ten instances, which resulted in an overall number of 120 instances. We excluded more facile settings in which rescue units numerically outnumber the number of incidents since this setting seems to be unrealistic. Table 1 provides an overview of how the instances were generated. In all Monte Carlo experiments, we used 500,000 iterations. Larger numbers of iterations did not result in better solutions in reasonable time.

**Table 1. Partly random generation of scenarios.**

Parameter	Values/Distribution
Numbers of Rescue Units (RU)	$K \in \{10,20,50\}$
Numbers of Incidents (Inc.)	$I \in \{10,20,50,100,200\}$
Replications of each scenario (RU Inc.)	10 (total: 120 different scenarios)
Processing times $p_j^k$	$\mu=20, \sigma=10$ (normal distribution)
Travel times $s_{ij}^k$	$\mu=1, \sigma=0.3$ (normal distribution)
Factors of destruction $w_j$	Random Integer $w_j \in \{1,2,3,4,5\}$ (discrete uniform distribution)
Capabilities of rescue units	$cap_{k,l} = \begin{cases} 1, & \text{if rescue unit } k \\ & \text{possesses capability } l \\ 0, & \text{else} \end{cases}$ $k \in K, l \in \{1, \dots, 5\}$ l=1; Search and Rescue Unit l=2; Medical Unit (Paramedics) l=3; Fire Brigade Unit l=4; Police Unit / Wardens l=5; Special Access Unit (discrete uniform distribution)
Categories of incidents (capabilities required)	$cat_{i,l} = \begin{cases} 1, & \text{if incident } i \\ & \text{requires capability } l \\ 0, & \text{else} \end{cases}$ $i \in I, l \in \{1, \dots, 5\}$
Iterations	500,000

We choose the crucial factor time not only to quantify our objective value but also to measure distances between (depots and) incidents. We have such an understanding of large-scale disasters that travel times are significantly shorter than times which are needed to process incidents (mean ratio: 1:20), yet less volatile. We make a sharp distinction between well-established skills of rescue units (e.g. medical or

firefighting). We classify rescue units as ‘‘Special Access Unit’’ if it cannot be assigned to any of the other classes (see Table 1 and [22]). Unlike in the model, we exclude the possibility that rescue units possess more than one capability.

The model was evaluated using a two-cored machine (2.53GHz, 2GB RAM). We chose this elementary environment to get insights into ‘‘poor’’ command centers equipped with household computers only and a missing link to high-speed infrastructure. Realistic results and runtimes may persuade to implement our approach in disaster-struck countries where sufficient computing facilities are missing. This information is essential to consecutively underline our research contribution and the fulfillment of requirement 1. Both heuristics have been implemented in MATLAB.

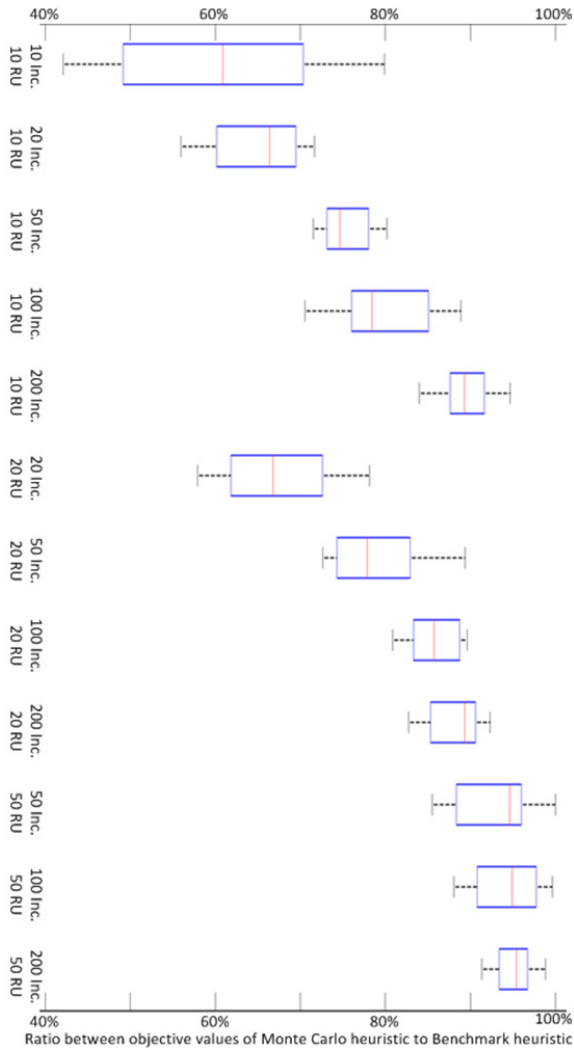
## 5. Results

Results of both heuristics are depicted in figure 2. The boxplots display most relevant statistical figures (means, quartiles, whiskers). Boxplots have been sketched for 12 different scenarios depending on the number of incidents (Inc.) and the number of rescue units (RU). Each box integrates the results of ten replications thus calculating to  $12 \times 10 = 120$  problem instances. The figure itself reads as follows: the scale represents the ratio between the two heuristics. Entries on the very right-hand side are to be understood that we are not able to substantially improve the benchmark by our Monte Carlo based heuristic. Data points on the very left-hand side can be interpreted as a benefit in comparison to the benchmark.

At a first glance, we notice that no outliers in the regular sense exist beyond all whiskers of the boxplots. It also seems that variances seem to be reasonably small (except for results of the first scenario) since boxplots are thin and results stay within a 10% interval. Coefficients of variation range between 3% (20Inc|50RU) and 10% (20Inc|20RU). This observation may induce that we can make reliable statements about the performance. Only the (10Inc|10RU) scenario has a coefficient of variation of 22%.

Apparently, all results of the proposed Monte-Carlo based heuristic are better than those of the benchmark. Some of them tend to excessively improve the benchmark especially in more straightforward scenarios (up to 57% (left whisker) in the 10Inc|10RU setting). The objective value can be improved to up to 30-40% in a (20Inc|20RU) environment. The performance adapts towards the benchmark in more

complex settings with more incidents evolving. All results have been statistically analyzed using a one-sample t-test to prove the superiority of the proposed heuristic. Testing leads to the conclusion that all Monte-Carlo based results do outperform the benchmark within a 95% level of significance.



**Figure 2. Results of the Monte Carlo based solution heuristic compared to the benchmark heuristic.**

Table 2 depicts the runtime behavior of the Monte-Carlo heuristic. In the smallest scenarios, results were generated within 12min, whereas in the most complex setting the computational time endured 2.5 hours. Results for all scenarios, which have been dealing with 50 incidents or less, were computed within one hour.

Yet, we hypothesize that 2.5 hours of waiting are too long for the generation of assignments and schedules, therefore, we recommend adapting the number of iterations to get results faster without losing

too much of its benefits. Cutting the number of iterations to 250,000 reduces the runtime of the (200Inc|50RU) scenario to approximately one and a half hours, whereas the mean ratio of the results weakens by only 1%.

Requirement 1 can be fulfilled even more a) by further reducing the number of Monte Carlo iterations or b) by increasing computation power. We assume that high-performance processors or advanced IT infrastructure cause runtimes to diminish to a minimum even in very complex scenarios. If one makes use of this adaptability of the Monte-Carlo based heuristic in complex settings, then requirement 1 can be fulfilled.

**Table 2. Mean runtimes of the Monte-Carlo based solution heuristic (in seconds).**

t[sec]	10 Inc.	20 Inc.	50 Inc.	100 Inc.	200 Inc.
10 RU	697	934	1,993	4,170	8,987
20 RU		1,068	2,142	4,037	8,574
50 RU			2,526	4,819	9,771

## 6. Conclusion

The management of emergency response is recognized as a key issue in literature and in disaster management practice. Coordination issues in particular have been lacking attention so far.

We proposed a novel quantitative decision support model for the allocation and scheduling of rescue units based on requirements identified in the related literature and in interviews. Due to the NP-hardness, we drew on a Monte Carlo based solution heuristic and computationally demonstrated its benefits. The results show that there is a large potential to improve a currently deployed (greedy) procedure. Yet, the research invites for future work, such as the introduction of a) time windows, b) pre-emption, or c) the employment with real-time data. For example, time windows are of particular importance when humans are buried alive and need to be saved. Pre-emptive approaches become necessary when rescue units need to improvise or act more autonomously or jobs need to be switched quickly and often.

Another research stream may enhance the applicability of our optimization model, such as the integration of fatigue characteristics of rescue units. Fatigue features become apparent when rescue forces lose some of their performance abilities caused by the duration of their deployment and the constant pressure to save lives over time. Yet, addressing these issues would cause additional constraints to the model.



## Appendix A: Literature Search Procedure

We scanned the literature in the fields of NDM and IS/computer science. Regarding the former field, our search procedure included the following steps:

- We performed a title search in technological- and management-oriented literature databases, namely Business Premier Source, EconLit, and ACM Digital Library (the search string was "(response OR system OR management) AND "disaster").
- We searched the proceedings of the "International Conference on Information Systems for Crisis Response and Management" and the table of contents of the journals "International Journal of Emergency Management", "International Journal of Emergency Response", and "Disaster Prevention and Management" (since 2000).

Regarding literature on information systems and computer science-related disaster management research, our search procedure included the following steps:

- We performed a title search in technological- and management-oriented literature databases, namely ACM Digital Library, Business Premier Source, EconLit, MLA (the search string was "information AND disaster"). We also searched the literature database "Web of Science" using the same search string. Due to an unmanageable number of results we refined the search by using the following search string: "disaster AND (management OR system OR information) AND design".
- We scanned the table of contents of premier IS outlets, including "European Journal of Information Systems", "Information Systems Journal", "Information Systems Research", "Journal of the AIS", "Management Information Systems Quarterly", and "Journal of the Management of Information Systems".
- We searched the proceedings of the "International Conference on Information Systems for Crisis Response and Management" and the table of contents of the journals "International Journal of Emergency Management", "International Journal of Emergency Response", and "Disaster Prevention and Management" (since 2000).

## 7. References

- [1] Airy, G., Mullen, T., and Yen, J. 2009. Market Based Adaptive Resource Allocation for Distributed Rescue Teams. *Proceedings of the 6th International ISCRAM Conference – Gothenburg, Sweden*.
- [2] Ajami, S. and Fattahi, M. 2009. The role of earthquake information management systems (EIMSs) in reducing destruction: A comparative study of Japan, Turkey and Iran. *Disaster Prevention and Management* 18, 2, 150–161.
- [3] Allahverdi, A., Ng, C., Cheng, T., and Kovalyov, M. 2008. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research* 187, 3, 985–1032.
- [4] Altay, N. and Green III, W. G. 2006. OR/MS research in disaster operations management. *European Journal of Operational Research* 175, 1, 475–493.
- [5] Bektas, T. 2006. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega* 34, 3, 209–219.
- [6] Beroggi, G. E. G. and Wallace, W. A. 1995. Real-Time Decision Support for Emergency Management: An Integration of Advanced Computer and Communications Technology. *Journal of Contingencies and Crisis Management* 3, 18.
- [7] Bharosa, N. and Janssen, M. 2009. Reconsidering information management roles and capabilities in disaster response decision-making units. *Proceedings of the 6th International ISCRAM Conference – Gothenburg, Sweden*.
- [8] Blazewicz, J., Dror, M., and Weglarz, J. 1991. Mathematical programming formulations for machine scheduling: A survey. *Operations Management Research* 51, 3, 283–300.
- [9] Bo, J., Xiabin, T., Ping, L., and Yanru, W. 2009. *WebGIS Based Information and Decision-Making Support System for Earthquake Disaster Reduction*. IEEE Computer Society.
- [10] Chen, R., Sharman, R., Rao, H. R., and Upadhyaya, S. 2008. An Exploration of Coordination in Emergency Response Management. *Communications of the ACM* 51, 66–73.
- [11] Chen, R., Sharman, R., Rao, H., and Upadhyaya, S. 2007. Design principles for critical incident response systems. *Information Systems and E-Business Management* 5, 3, 201–227–227.
- [12] Comes, T., Conrado, C., Hiete, M., Kamermans, M., Pavlin, G., and Wijngaards, N. 2010. An intelligent decision support system for decision making under uncertainty in distributed reasoning frameworks. In *Proceedings of the 7th International ISCRAM Conference, Seattle, USA*.
- [13] Day, J. M., Junglas, I., and Silva, L. 2009. Information Flow Impediments in Disaster Relief Supply Chains. *Journal of the Association for Information Systems* 10, 8, 637–660.
- [14] Falasca, M., Zobel, C. W., and Fetter, G. M. 2009. An optimization model for humanitarian relief volunteer management. In *Proceedings of the 6th International ISCRAM Conference, Gothenburg, Sweden*.

- [15] Faraj, S. and Xiao, Y. 2006. Coordination in fast-response organizations. *Management Science* 52, 8, 1155–1169.
- [16] Fiedrich, F., Gehbauer, F., and Rickers, U. 2000. Optimized resource allocation for emergency response after earthquake disasters. *Safety Science* 35, 1-3, 41–57.
- [17] Fruhling, A. N. and Vreede, G.-J. de. 2006. Field Experiences with eXtreme Programming: Developing an Emergency Response System. *Journal of Management Information Systems* 22, 4, 39–68.
- [18] Hale, J. 1997. A layered communication architecture for the support of crisis response. *Journal of Management Information Systems* 14, 1, 235–255.
- [19] IFRC. 2010. *Disaster management - IFRC*. <http://www.ifrc.org/en/what-we-do/disaster-management/>.
- [20] Leifler, O. 2008. Combining Technical and Human-Centered Strategies for Decision Support in Command and Control: The ComPlan Approach. *Proceedings of the 5th International ISCRAM Conference - Washington, DC, USA*, 504–515.
- [21] Mendonça, D., Giampiero, B. E., and Wallace, W. A. 2001. Decision support for improvisation during emergency response operations. *International Journal of Emergency Management* 1, 1, 30–39.
- [22] New South Wales Government. 2007. *Emergency Operations*. <http://www.ambulance.nsw.gov.au/about-us/Emergency-Operations.html>. Accessed 5 March 2012.
- [23] Reijers, H. A., Jansen-Vullers, M. H., Zur Muehlen, M., and Appl, W. 2007. Workflow management systems + swarm intelligence = dynamic task assignment for emergency management applications. *Proceedings of the 5th International Conference on Business Process Management* 2007, 125–140.
- [24] Rolland, E., Patterson, R., Ward, K., and Dodin, B. 2010. Decision support for disaster management. *Operations Management Research* 3, 1, 68–79.
- [25] Schimmelpfennig, S. 2010. *Coordination after a disaster*. <http://goodintents.org/uncategorized/coordination-after-a-disaster>. Accessed 11 November 2010.
- [26] Schryen, G. and Wex, F. 2012. IS Design Thinking in Disaster Management Research. *Proceedings of the 45th Annual Hawaii International Conference on System Sciences, Maui, USA*.
- [27] Turoff, M. 2002. Past and future emergency response information systems. *Communications of the ACM* 45, 4, 29–32.
- [28] Turoff, M., Chumer, M., van de Walle, B., and Yao, X. 2003. The Design of a Dynamic Emergency Response Management Information System (DERMIS). *Journal of Information Technology Theory and Application (JITTA)* 5, 4.
- [29] van de Walle, B. and Turoff, M. 2008. Decision Support for Emergency Situations. In *Handbook on Decision Support Systems 2*, F. Burstein and C. Holsapple, Eds. International Handbooks on Information Systems. Springer Berlin Heidelberg, 39–63.
- [30] Weng, M. X., Lu, J., and Ren, H. 2001. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. *International Journal of Production Economics* 70, 3, 215–226.
- [31] Wex, F., Schryen, G., and Neumann, D. 2011. Intelligent Decision Support for Centralized Coordination during Emergency Response. *Proceedings of the 8th International ISCRAM Conference – Lisbon, Portugal*.
- [32] Wex, F., Schryen, G., and Neumann, D. 2012. Operational Emergency Response under Informational Uncertainty: A Fuzzy Optimization Model for Scheduling and Allocating Rescue Units. *Proceedings of the 9th International ISCRAM Conference – Vancouver, Canada*.