Decision Support for the Optimal Coordination of Spontaneous Volunteers in Disaster Relief

Gerhard Rauchecker
University of Regensburg, Germany
gerhard.rauchecker@ur.de

Guido Schryen
University of Regensburg, Germany
guido.schryen@ur.de

ABSTRACT

When responding to natural disasters, professional relief units are often supported by many volunteers which are not affiliated to humanitarian organizations. The effective coordination of these volunteers is crucial to leverage their capabilities and to avoid conflicts with professional relief units. In this paper, we empirically identify key requirements that professional relief units pose on this coordination. Based on these requirements, we suggest a decision model. We computationally solve a real-world instance of the model and empirically validate the computed solution in interviews with practitioners. Our results show that the suggested model allows for solving volunteer coordination tasks of realistic size near-optimally within short time, with the determined solution being well accepted by practitioners. We also describe in this article how the suggested decision support model is integrated in the volunteer coordination system which we develop in joint cooperation with a disaster management authority and a software development company.

Keywords

Coordination of spontaneous volunteers, volunteer coordination system, decision support, scheduling optimization model, linear programming

INTRODUCTION

In the light of climate change, urbanization, and global conflicts, natural and man-made disasters have increasingly severe consequences on humans and their environment (Schryen et al. 2015). This leads to a rising importance of an effective disaster relief. In the wake of a disaster, professional relief units are usually supported by volunteers affiliated with humanitarian organizations, e.g., the International Federation of Red Cross and Red Crescent Societies (IFRC 2015). In addition to these affiliated volunteers, cooperativeness of citizens results in large amounts of spontaneous (i.e., not affiliated to any humanitarian organization) volunteers offering their help. As many as two million citizens assisted others in the aftermath of the 1985 earthquake in Mexico City (Dynes et al. 1990) and after 9/11, more than 15,000 people gathered at Ground Zero offering their services (Lowe and Fothergill 2003). The effective usage of these additional capabilities provides an invaluable resource that must not be neglected. However, spontaneous volunteers are present on disaster sites regardless of a request for their assistance and their mass movements (also known as *convergence*) are highly unpredictable (Fernandez et al. 2006; Lodree and Davis 2016). This makes their coordination a complicated task and often a burden. In summary, the coordination of spontaneous volunteers is a challenging yet very important part of disaster relief (FEMA 2013; Mayorga et al. 2017).

The unpredictable behavior of spontaneous volunteers results in an ad-hoc on-site coordination when it comes to real-world disasters. However, this distracts professional responders from their primary duties (Fernandez et al. 2006). This serious issue calls for the deployment of volunteer coordination systems that enable an off-site coordination of spontaneous volunteers, i.e., before they arrive at the disaster site. Such a system must enable a live interaction between professional responders and off-site volunteers during the disaster situation via modern information and communication technology. Furthermore, appropriate mathematical models and algorithms for solving problem instances are key to supporting decisions on how to exploit the capabilities of volunteers. The literature on quantitative approaches for volunteer coordination is discussed in the following section. However, we found out that all approaches miss at least one key requirement for coordinating spontaneous volunteers that we have identified in interviews with professionals at the Disaster Management Authority in Halle (Saale), Germany, which is referred to as *DMA Halle (Saale)* in the remainder of the paper). This issue leads to the following research question that we address in our paper: *How can the problem of coordinating spontaneous volunteers be mathematically modeled and solved in order to improve the coordination of spontaneous volunteers and to relieve professional responders in real-world disasters?*

This research question is addressed in our paper by developing and testing a novel approach that automatizes the coordination of spontaneous volunteers. Based on key requirements for coordinating spontaneous volunteers, which we have identified in interviews with professionals at the DMA Halle (Saale), we present an optimization model that can be solved by off-the-shelf-solvers. We computationally solved a real-world instance of the suggested model and empirically validated the determined solution of the coordination task. Our results show that a problem instance of realistic size can be solved near-optimally within short time with the determined solution being well accepted by practitioners. Professionals at the DMA Halle (Saale) confirmed that such complex coordination decisions can hardly be provided manually by professional responders in comparable quality - especially when they are under severe time pressure. Working closely together with the DMA Halle (Saale) and the software development company *esri Germany*, our optimization model is integrated in a holistic volunteer coordination system called *KUBAS*, which guarantees the practical applicability of our approach (see KUBAS (2018) for further information). Thus, our contributions can be used in practice to improve the coordination of spontaneous volunteers and to relieve professional responders at the same time.

The remainder of the paper is structured as follows: In the following section, we discuss relevant literature. Then, we provide a detailed overview on the KUBAS system. After that, we give a description of our volunteer scheduling problem as well as a mathematical formulation as a linear optimization model. Subsequently, we use an off-the-shelf solver to solve case study instances that represent a real-world flood scenario. Finally, we discuss our results and present our conclusions.

RELATED WORK

Volunteerism in general and volunteer management in particular are vitally discussed in the field of disaster and emergency management, see reviews by Alexander (2010) and Whittaker et al. (2015) for an overview. In this regard, especially the use of information systems has gained increasing attention in recent years, see Auferbauer et al. (2016), Van Gorp (2014), Havlik et al. (2016), Horita and De Albuquerque (2013), Lindner et al. (2018), and Siemen et al. (2017), for example. In our paper, we focus on the coordination of spontaneous volunteers in disaster relief. As unfolded in the introduction, this demands for automatized decision support and therefore quantitative approaches. In the following, we discuss relevant literature.

Mayorga et al. (2017) present a queuing system to model the uncertain arrival and departure of spontaneous volunteers. They present four heuristics to assign volunteers to queues. This is the only article that explicitly deals with coordinating spontaneous volunteers. Liu and Wang (2013) develop an integer linear program (ILP) to coordinate entire volunteer teams and minimize their travel costs. Pielorz and Lampert (2015) also present an ILP to coordinate the volunteer organization *Team Austria*. They seek an assignment that requires a minimal amount of volunteers to meet all task demands. The approaches in both articles assume that task demand can always be fulfilled, i.e., that there is no shortage of volunteers in any task.

Falasca, Zobel, and Fetter (2009) and Falasca and Zobel (2012) formulate volunteer scheduling as a bi-objective optimization model. They minimize the number of unmet task demands as well as the number of volunteer assignments to undesired tasks or time slots. Furthermore, Falasca, Zobel, and Ragsdale (2011) introduce a spreadsheet modeling approach and apply it to a South American development organization. They use a single objective function representing a weighted sum of the percentage of undesired assignments, the percentage of unmet task demands, and the percentage of budget used. Lassiter, Alwahishie, et al. (2014) and Lassiter, Khademi, et al. (2015) optimize volunteer scheduling on a semi-professional level involving long-term training. They seek to minimize several costs associated with the volunteer coordination. Another approach was presented by Aman et al. (2012) who use a goal programming model to schedule volunteer organizations after a volcano eruption. Their objective is to minimize deviations from task demands as well as from the desired number of shifts expressed by each volunteer.

Two earlier works on volunteer coordination have been developed in application areas different from disaster management. Sampson (2006) presents a goal programming model minimizing deviations in both task demands and volunteer preferences. They apply their model to reviewer assignment for a scientific conference. In addition, they elaborate crucial differences between volunteer assignment and traditional labor assignment; the most important of

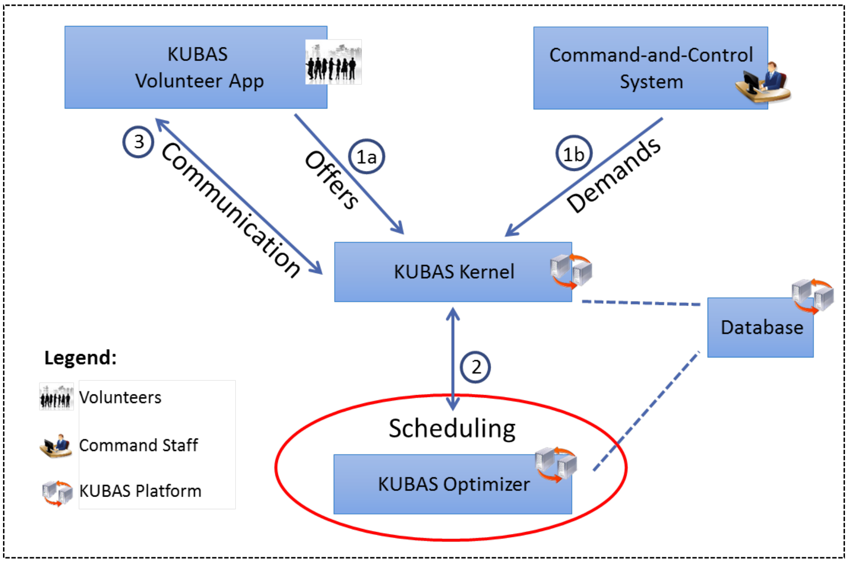

Figure 1. Volunteer coordination system KUBAS

which are that the variable costs of labor are usually trivial for volunteers and that there is a possibility that task demands cannot be fully covered. This prevents using algorithms from the vitally studied field of traditional labor assignment (see Van den Bergh et al. (2013) for an overview) to the volunteers setting. Gordon and Erkut (2004) present an optimization model to automatize the volunteer shift assignment process for the Edmonton Folk Festival. They assume that enough volunteers are available and maximize volunteers' shift preferences subject to covering all shifts.

With the exception of the work of Mayorga et al. (2017), who suggest an approach for on-site coordination, all other approaches address off-site coordination, which is the focus of our work. However, none of the latter approaches considers a key requirement that we identified in interviews with professionals at the DMA Halle (Saale): When there is volunteer shortage (which may occur, for example, at night or when there is bad weather), unmet task demands should be kept to a minimum while the unmet demands should be distributed among tasks in a balanced way. For example, when there are two tasks that both demand for ten volunteers of the same capability and there are only ten eligible volunteers available, then it is better to assign five volunteers to each task than to assign ten volunteers to one task and no volunteer to the other task. In order to develop a decision support approach for off-site coordination of spontaneous volunteers that accounts for requirements occurring in real-word scenarios, including the previously mentioned one, we (1) conducted interviews with professionals at the DMA Halle (Saale) to identify those requirements and (2) propose and (computationally and empirically) validate an optimization model that considers the identified requirements. A full list of requirements is presented in the *Problem Description* section.

VOLUNTEER COORDINATION SYSTEM KUBAS

In this section, we explain the volunteer coordination system KUBAS for which the optimization model presented in this article is the key decision support methodology. The KUBAS system is developed in close cooperation with the DMA Halle (Saale) and esri Germany. As demanded by the former, it is designed to enable an optimal coordination of spontaneous volunteers *before* they arrive at the disaster site in an uncontrolled way. To achieve this, the KUBAS system takes advantage of both mathematical methods, which are developed in our paper, and modern information and communication technology. A graphical overview of the KUBAS system is given in Figure 1, which is explained in the following. Using the term *volunteer*, we refer to a spontaneous volunteer in the remainder of the paper.

Volunteers can register at any time via smartphones using the *KUBAS volunteer app* and submit offers for their help to an administrative component called *KUBAS kernel*. The command staff is connected to the KUBAS kernel via an interface to their command-and-control system, which allows them to submit and prioritize demands for volunteers. The submission of offers and demands is labeled as steps 1a and 1b, respectively, in the figure. After collecting this information, the KUBAS kernel calls the *KUBAS optimizer*, which uses an optimization model to coordinate the registered volunteers. The KUBAS optimizer matches offers and demands to schedule volunteers in a way that satisfies preferences of both the command staff and the volunteers. This is marked as step 2 and is the focus of this paper. Finally, relevant information about the scheduling decision is communicated to the individual

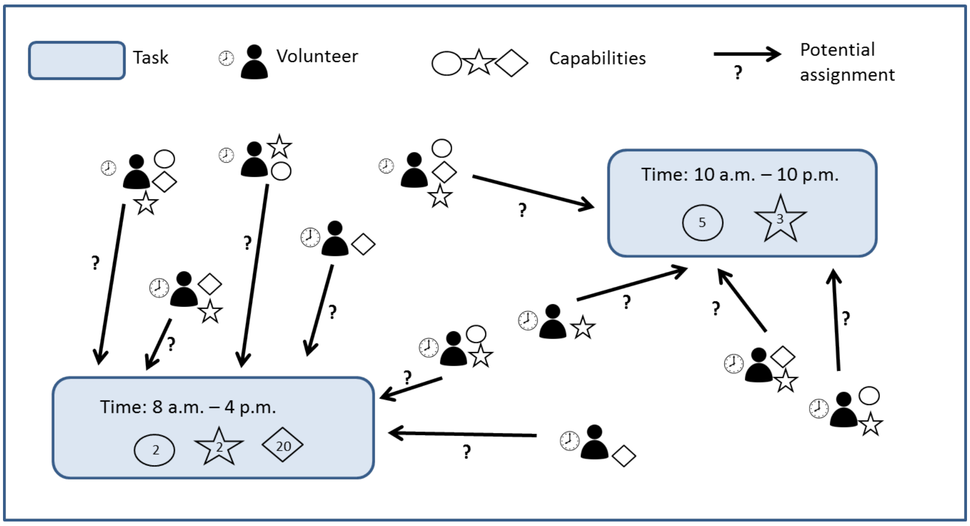

Figure 2. Sample situation for VSP with two tasks and ten volunteers

volunteers, which are prompted to confirm their assignments (step 3). We stress that our interview partners at the DMA Halle (Saale) pointed out that due to time pressure, it is more convenient for the command staff to not confirm or adjust the schedules calculated by the KUBAS optimizer. Therefore, the scheduling decisions are immediately broadcast to the volunteers. Note that it may be convenient to also communicate the scheduling decision to the command-and-control system for visualization purposes. However, this it not essential for the workflow presented in Figure 1. A description of a basic implementation of the KUBAS system – including a graphical user interface as proxy for the command-and-control system – can be obtained from Betke (2018).

The ad-hoc nature of volunteers implies that they may register in any number at any time. In addition, the command staff may submit new demands or reset their priorities at any time due to the unpredictability of changes in disaster relief situations. The KUBAS system explicitly takes into account this dynamic setting. The KUBAS kernel starts the KUBAS optimizer whenever the situation has changed substantially, e.g., when new demands occur, when priorities change, or when a number of new volunteers register. Each time when the KUBAS optimizer is started, it schedules the registered volunteers based on current information, i.e., a sequence of scheduling problem instances is solved over time to reflect the dynamic setting. Technically spoken, steps 2 and 3 are repeated automatically as soon as new submissions in steps 1a and 1b change the situation substantially.

PROBLEM DESCRIPTION AND DECISION SUPPORT MODEL

Problem Description

In the following, we describe the problem which is solved by the KUBAS optimizer. To construct a realistic setting, we have developed the problem in close cooperation with the DMA Halle (Saale). It is important that volunteers can be coordinated with high flexibility. Our interview partners therefore prefer a *scheduling* problem type instead of a pure assignment problem. [1] Consequently, we refer to our problem as *volunteer scheduling problem* (VSP). We present and explain VSP using a fictitious sample situation, which is illustrated in Figure 2.

Each VSP instance consists of a set of volunteers who offer their help. Each volunteer expresses his/her preferences in terms of available times and personal capabilities. For example, a volunteer may be available from 8 a.m. to 1 p.m. and again from 6 p.m. to 10 p.m. offering the capabilities *physical labor* and *writing*.

Furthermore, there is a set of *tasks* which volunteers are required to work on. Each task has a fixed location and a specified time when it takes place. Each task consists of several activities, each of which is characterized by demanding a certain number of volunteers with a specific capability. An example for a task during a flood would be *working at the sandbag supply station* taking place from 8 a.m. to 4 p.m. and consisting of the following activities: *filling sandbags* with a demand for 10 volunteers capable of physical labor; *carrying sandbags* with a demand for 10 volunteers capable of physical labor; *disposing meals* with a demand for 2 volunteers capable of light physical labor; and *documenting* with a demand for 2 volunteers capable of writing. These activities require in total 20

[1]Note that a scheduling problem type is possible because we coordinate off-site volunteers. On-site volunteers do not allow for this flexibility since they cannot be sent back home (to be assigned at a later time) without negative consequences on their motivation.

volunteers with the capability *physical labor*, 2 volunteers with capability to perform *light physical labor*, and 2 volunteers with *writing* capabilities.

Finally, each VSP instance addresses a discrete planning horizon with a positive number of equal-length time slots. The aim of solving the VSP is to schedule the volunteers to process all *task activities* (e.g., activity *filling sandbags* of task *working at the sandbag supply station*) during the entire planning horizon such that real-world requirements are fulfilled in an optimal way. We have identified those requirements in personal interviews with professionals at the DMA Halle (Saale) and present them in the following:

(i) The crucial metric for all task activities at any time is the ratio of assigned to requested volunteers. These ratios are called *workloads* and become especially important when there is a shortage of volunteers. The overall goal is to maximize the aggregated workload over all task activities and all time slots under the restriction that workloads must not exceed 100%; i.e., there must not be assigned more volunteers than requested to any task activity in any time slot.

(ii) There are several priority levels for tasks. Unmet demands for volunteers are more severe for activities of a higher-prioritized task than for activities of a lower-prioritized task. When there is volunteer shortage, workloads (i.e., assignment ratios) should be higher in activities of higher-priority tasks.

(iii) Unmet demands for volunteers are more severe in earlier time slots than in later time slots, since unmet demands in early time slots are more likely to be lost for good than demands in later time slots (which might be fulfilled when new volunteers register).

(iv) Workloads of task activities that have the same priority levels and require the same capability should be identical within each time slot.

(v) When task activities with different priorities demand for volunteers with the same capability at the same time, there must be a compensation between these task activities; i.e., activities of lower-priority tasks may be filled to a certain level even if activities of higher-priority tasks cannot be filled to 100%. [2]

(vi) A volunteer cannot work on more than one task activity at the same time.

(vii) A volunteer can only be scheduled to process a task activity at a certain time slot when s/he has the appropriate capability and availability.

(viii) When a volunteer is assigned to a task activity, there must be a lower bound on the time s/he works consecutively on this task activity because short work blocks would lead to chaotic situations.

(ix) There must be an upper bound on the total working hours during the entire planning horizon that must not be exceeded by any volunteer.

(x) Volunteers require a break (setup time) when switching from one task activity to another. Factors contributing to setup times are, for example, traveling between locations of activities, changing wet clothes, consuming food, or getting briefed about the next activity.

While requirements (i)-(vi) mainly ensure that a VSP solution is regarded as optimal by the command staff, requirements (vii)-(x) aim at achieving a high level of volunteer satisfaction, which is important to maintain volunteer motivation – in the current disaster as well as future disaster situations.

As outlined in the previous section, the KUBAS optimizer is started automatically whenever the situation changes substantially. Then, a new VSP instance is generated and solved based on current information, which includes the scheduling decisions from the solution of the previous VSP instance. Since scheduling decisions are broadcast to volunteers immediately after solving a VSP instance (see step 3 in Figure 1), the following requirement aims at avoiding confusion among volunteers:

(xi) When a volunteer has assignments from the solution of the previous VSP instance, these assignments must not be changed by the solution of the current VSP instance.

This requirement implies that the feasible solutions of a VSP instance are also affected by the optimal solution of the previously solved VSP instance, the scheduling decisions of which must not be changed. Note that all presented requirements hold for arbitrary off-site volunteers, not only for spontaneous volunteers. Although the intention of the KUBAS system is to coordinate (otherwise hardly controllable) spontaneous volunteers, it can be used to coordinate any off-site volunteers – as long as they register via the KUBAS volunteer app.

[2]Note that assigning volunteers to activities of a lower-priority task leads to smaller workloads in activities of higher-priority tasks.

Table 1. Notation for optimization model

Notation	Description
$t = 0, \ldots, T$	Time slots (planning horizon)
$i = 1, \ldots, V$	Volunteers
$c = 1, \ldots, C$	Capabilities
$j = 1, \ldots, A$	Task activities
$p = 1, \ldots, P$	Priorities
$\mathcal{A}_{pct}$	Set of task activities that have priority p and demand for volunteers with capability c at time t
X_{ijt}	Binary decision variable (volunteer i is assigned to task activity j at time t or not)
L_{pct}	Continuous variable (minimum workload among all task activities $j \in \mathcal{A}_{pct}$)
pr_j	Priority of task activity j (depends only on the underlying task)
r_{jt}	Binary indicator showing whether time slot t is within the time frame of task activity j or not (depends only on the underlying task)
req_{jc}	Binary indicator (task activity j requires capability c or not)
n_j	Number of volunteers requested by task activity j
w_{pt}	Weight parameter which indicates the severity of unmet demands for volunteers in time slot t at task activities with priority p
d_{pt}	Scaling parameter which represents the fraction of demands of the task activities with priority p and demands of all task activities at time t
$\sigma_{p-1,p}$	Compensation factor which indicates the relative importance of adjacent priorities $p-1$ and p
o_{ijt}	Binary indicator (volunteer i has an assignment to task activity j in time slot t from the solution of the previous VSP instance or not)
a_{it}	Binary indicator (volunteer i is available in time slot t or not)
cap_{ic}	Binary indicator (volunteer i offers capability c or not)
τ_1	Number of slots that a volunteer must work consecutively on the same task activity
τ_2	Number of slots that each volunteer is allowed to work in total during the planning horizon
τ_3	Number of slots for setup time between working on different task activities

Decision Support Model

Based on the problem description, we present a linear optimization model for the VSP. All notation that we use and introduce in this subsection is summarized in Table 1. The table lists all relevant indices, sets, variables, and parameters – in this order and separated by horizontal lines. A problem instance comprises a discrete planning horizon of equal-length time slots $t = 0, \ldots, T$ in which volunteers $i = 1, \ldots, V$ with certain capabilities $c = 1, \ldots, C$ are scheduled to work on task activities $j = 1, \ldots, A$ which have task-specific priorities $p = 1, \ldots, P$.

Each task activity (e.g., activity *filling sandbags* of task *working at the sandbag supply station*) comes with a priority pr_j (where higher numbers represent a higher priority) and a time frame (indicated by binary parameters r_{jt}) in which volunteer help is needed. Priority and time frame of a task activity depend only on the underlying task, i.e., they are the same among all activities of the same task. A task activity j is further characterized by a single capability that the underlying task requires (indicated by binary parameters req_{jc}) and the corresponding number n_j of requested volunteers.

We use binary decision variables X_{ijt} to indicate whether volunteer i is assigned to work on task activity j in time slot t ($X_{ijt} = 1$) or not ($X_{ijt} = 0$). Furthermore, we define

$$\mathcal{A}_{pct} := \{j = 1, \ldots, A | pr_j = p \text{ and } r_{jt} \cdot req_{jc} = 1\}$$

as the set of task activities that have priority p and request for volunteers with capability c in time slot t. This allows us to introduce variables

$$L_{pct} = \min_{j \in \mathcal{A}_{pct}} \frac{1}{n_j} \sum_{i=1}^{V} X_{ijt},$$

which represent the minimum workload (i.e., assignment ratio) among all task activities $j \in \mathcal{A}_{pct}$. Note that these minimum workloads are to be maximized in the objective function to take into account requirements (i) and (iv). To show how the variables L_{pct} help to balance workloads and therefore to fulfill requirement (iv), we present a small example. Consider a scenario where we have two task activities, both with task priority $p = 2$. Each task activity demands for 40 volunteers with capability $c = 4$. We fix a time slot $t = 6$ and assume that there are exactly 40 eligible volunteers. Now, we compare L_{246} for three different possible solutions in Figure 3. In the first solution,

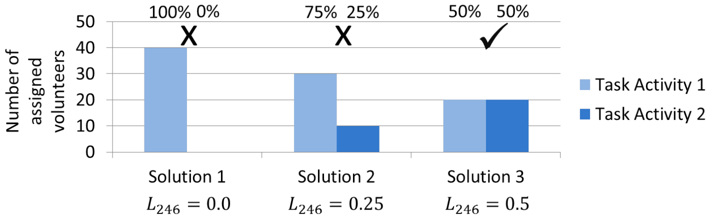

Figure 3. Comparison of L_{246} for three different feasible solutions

the workload of task activity 1 is 100% while the workload of task activity 2 is 0%. L_{246} is the minimum of both workloads and hence $L_{246} = 0.0$. The values of $L_{246} = 0.25$ in solution 2 and $L_{246} = 0.5$ in solution 3 can be calculated analogously. In summary, solution 3 has the highest minimum workload L_{246} among the presented solutions.

According to requirements (ii) and (iii), we introduce weights $w_{pt} > 0$, which indicate the severity of unmet demands for volunteers in time slot t at task activities with priority p. However, maximizing (a sum of) weighted workloads $w_{pt} \cdot L_{pct}$ is not sufficient to fulfill all requirements (i)-(iv) as we illustrate again in a small example. We fix a time slot t and assume $w_{1t} = 1$ and $w_{2t} = 5$. In particular, priority $p = 2$ is more important than priority $p = 1$. Consider an example where two task activities demand for the same capability c. Task activity 1 has priority $p = 1$ and demands for 5 volunteers. Task activity 2 has priority $p = 2$ and demands for 50 volunteers. Further assume that there are exactly 5 eligible volunteers. Now, the maximum value for $w_{1t} \cdot L_{1ct} + w_{2t} \cdot L_{2ct}$ is achieved by assigning all volunteers to task activity 1 resulting in $1 \cdot 1 + 5 \cdot 0 = 1$. However, assigning all volunteers to the task activity with the lower priority and no volunteer to the task activity with the higher priority contradicts requirement (ii). To address this issue, we define for all priorities p and time slots t a scaling parameter

$$d_{pt} = \frac{\sum_{c=1}^{C} \sum_{j \in \mathcal{A}_{pct}} n_j}{\sum_{j=1}^{A} r_{jt} \cdot n_j}$$

which represents the fraction of the demands of task activities j with priority $pr_j = p$ at time t and the demands of all task activities at time t. In the example, we have $d_{1t} = \frac{1}{11}$ and $d_{2t} = \frac{10}{11}$. Evaluating $w_{1t} \cdot d_{1t} \cdot L_{1ct} + w_{2t} \cdot d_{2t} \cdot L_{2ct}$ we see that the sum now achieves its maximum when all five volunteers are assigned to the higher-priority task activity. Using the introduced notation, we formulate the following optimization model:

$$\max \quad \sum_{p=1}^{P} \sum_{c=1}^{C} \sum_{t=1}^{T} w_{pt} \cdot d_{pt} \cdot L_{pct} \qquad \text{(Max-WL)}$$

$$L_{pct} \leq \frac{1}{n_j} \cdot \sum_{i=1}^{V} X_{ijt} \qquad \forall p = 1, \ldots, P; c = 1, \ldots, C; t = 1, \ldots, T; j \in \mathcal{A}_{pct} \qquad (1)$$

$$\sum_{i=1}^{V} X_{ijt} \leq n_j \cdot r_{jt} \qquad \forall j = 1, \ldots, A; t = 1, \ldots, T \qquad (2)$$

$$0 \leq L_{pct} \qquad \forall p = 1, \ldots, P; c = 1, \ldots, C; t = 1, \ldots, T \qquad (3)$$

$$X_{ijt} \in \{0, 1\} \qquad \forall i = 1, \ldots, V; j = 1, \ldots, A; t = 1, \ldots, T \qquad (4)$$

A mapping of requirements from the previous subsection and their equivalents in the model is provided in Table 2. The objective function (Max-WL) maximizes a weighted (by parameter w_{pt}) and scaled (by parameters d_{pt}) sum of workload variables L_{pct}, which are specified by constraint (1). Constraint (2) ensures that the number of assigned volunteers does not exceed the number of requested volunteers at any task activity in any time slot. The workload variables L_{pct} also balance workloads among all task activities $j \in \mathcal{A}_{pct}$ as presented in Figure 3. Constraints (3) and (4) define the domain of definition of the variables L_{pct} and X_{ijt}. [3] To sum up, the objective function (Max-WL) and constraints (1)-(4) ensure requirements (i)-(iv).

[3]Note that constraint (2) also implies $L_{pct} \leq 1$ for all priorities p, capabilities c, and time slots t.

Table 2. Mapping of requirements vs. constraints

Requirements	Constraints
(i), (ii), (iii), (iv)	(Max-WL), (1), (2), (3),(4)
(v)	(5)
(vi)	(6)
(vii)	(7)
(viii)	(8), (9)
(ix)	(10)
(x)	(11)
(xi)	(12)

In order to fulfill the compensatory requirement (v) for task activities of different priorities, we introduce a compensation factor $\sigma_{p-1,p} \in [0,1)$ for all priorities $p = 2, \dots, P$ which indicates the relative importance of priority $p-1$ in comparison to p in all time slots and formulate the following constraint: [4]

$$L_{(p-1)ct} \geq \sigma_{p-1,p} \cdot L_{pct} \quad \forall p = 2, \dots, P; c = 1, \dots, C; t = 1, \dots, T \tag{5}$$

This constraint ensures that for any priority $p \geq 2$, any capability c, and any time slot t, the workload $L_{(p-1)ct}$ must be at least as high as the workload L_{pct} multiplied by the compensation factor (e.g., 30% which corresponds to $\sigma_{p-1,p} = 0.3$). When $\sigma_{p-1,p} = 0$, we do not have any compensation between the priorities $p-1$ and p. Thereby, constraint (5) takes into account requirement (v).

In the following, we introduce additional constraints that are necessary to fulfill requirements (vi)-(xi). A binary parameter a_{it} indicates availabilities and equals 1 if and only if volunteer i is available in time slot t. Another binary parameter cap_{ic} is 1 if and only if volunteer i has capability c. Furthermore, let the binary parameter o_{ijt} be 1 if and only if volunteer i has an assignment to task activity j in time slot t from the solution of the previous VSP instance. [5] Let τ_1 be the minimum number of time slots that a volunteer must work consecutively on the same task activity. Let τ_2 be the maximum number of time slots that a volunteer is allowed to work in total during the entire planning horizon. Finally, let τ_3 be the number of time slots that represents the setup time between working on two different task activities. [6] This notation allows for formulating the following constraints:

$$\sum_{j=1}^{A} X_{ijt} \leq 1 \quad \forall i = 1, \dots, V; t = 1, \dots, T \tag{6}$$

$$X_{ijt} \leq a_{it} \cdot \sum_{c=1}^{C} cap_{ic} \cdot req_{jc} \quad \forall i = 1, \dots, V; j = 1, \dots, A; t = 1, \dots, T \tag{7}$$

$$X_{ij0} = o_{ij0} \quad \forall i = 1, \dots, V; j = 1, \dots, A \tag{8}$$

$$\sum_{t'=1}^{\min\{\tau_1, T-t+1\}} X_{ij(t+t'-1)} + \tau_1 \cdot \left(X_{ij(t-1)} - X_{ijt} + 1\right) \geq \tau_1 \quad \forall i = 1, \dots, V; j = 1, \dots, A; t = 1, \dots, T \tag{9}$$

$$\sum_{j=1}^{A} \sum_{t=1}^{T} X_{ijt} \leq \tau_2 \quad \forall i = 1, \dots, V \tag{10}$$

$$\sum_{j \neq j'=1}^{A} \sum_{t'=1}^{\min\{\tau_3, T-t\}} X_{ij'(t+t')} \leq \tau_3 \cdot (1 - X_{ijt}) \quad \forall i = 1, \dots, V; j = 1, \dots, A; t = 1, \dots, T \tag{11}$$

$$X_{ijt} \geq o_{ijt} \quad \forall i = 1, \dots, V; j = 1, \dots, A; t = 1, \dots, T \tag{12}$$

[4] We assume that priority levels are ordered such that priority $p = 1$ represents the lowest priority level and $p = P$ represents the highest priority level.

[5] Note that we solve a sequence of dependent VSP instances over time as explained at the end of the previous subsection.

[6] We assume a constant setup time of τ_3 time slots since it is impossible to exactly predict sequence- and volunteer-dependent setup times in the chaotic setting of disaster relief. Nevertheless, the concept of setup times is an important feature of real-world scheduling and consequently is included in the model.

Constraint (6) assures that a volunteer is not assigned to multiple task activities simultaneously, see requirement (vi). Constraint (7) ensures that a volunteer is only assigned to task activities where s/he has the matching capability and in time slots where s/he is available, see requirement (vii). Constraint (8) includes the assignments that volunteers have at time of decision making $t = 0$. [7] We have $X_{ij0} = o_{ij0} = 1$ if and only if volunteer i works on task activity j at time of decision making due to an assignment from the solution of the previous VSP instance. This is important for constraint (9) where the index $t - 1$ can be 0. In constraint (9), we have $X_{ij(t-1)} - X_{ijt} + 1 = 0$ if and only if $X_{ijt} = 1$ and $X_{ij(t-1)} = 0$, i.e., when volunteer i begins working on task activity j at the beginning of time slot t. [8] In this case, constraint (9) assures that volunteer i keeps working on task activity j for at least τ_1 time slots. The combination of constraints (8) and (9) therefore ensures requirement (viii). Constraint (10) assures that each volunteer does not work more than τ_2 time slots in total during the entire planning horizon, see requirement (ix). In constraint (11), we have $1 - X_{ijt} = 0$ if and only if $X_{ijt} = 1$. In this case, it is assured that volunteer i has no assignment to any task activity different from j during the τ_3 time slots after t. [9] Therefore, constraint (11) accounts for the setup times from requirement (x). Constraint (12) guarantees that if a volunteer i has an assignment to task activity j in time slot t from the solution of the previous VSP instance (i.e., $o_{ijt} = 1$) then X_{ijt} must be 1. Consequently, requirement (xi) is fulfilled. Finally, all features of VSP as introduced in the previous subsection are incorporated in the presented model with objective function (Max-WL) and constraints (1)-(12).

CASE STUDY: 2013 SAALE FLOODING

We evaluate the presented model using tasks from the Saale flooding, which was part of the major floods across Central Europe in early summer 2013. This flooding was characterized by the highest Saale water level ever recorded and a high number of uncoordinated volunteers according to interviews with practitioners. We begin with a description of our case study instances before we explain our parameter setting for solving the corresponding optimization models. Finally, we present and discuss our computational results. All information and data in this section is conducted from an internal flood report of the DMA Halle (Saale) and personal interviews with professionals at the agency.

Case Study Instances

In our case study, we distinguish four capabilities, namely *hard physical labor* ($c = 1$), *medium physical labor* ($c = 2$), *light physical labor* ($c = 3$), and *writing* ($c = 4$). Furthermore, we distinguish three different priorities, namely *low* ($p = 1$), *medium* ($p = 2$), and *high* ($p = 3$) priority. We demonstrate the applicability of our model in a dynamic setting by considering a sequence of two dependent VSP instances that represent real instances during the Saale flooding 2013:

Instance 1 *The initial instance consists of the four tasks flood control 1, flood control 2, flood control 3, and sandbag supply. Flood control tasks consist of the three activities carrying sandbags, disposing meals, and documentation which require the capabilities hard physical labor, light physical labor, and writing, respectively. A sandbag supply task consists of the three mentioned activities plus the activity filling sandbags which requires the capability medium physical labor. This leads to a total of 13 task activities. The demands and priorities of all tasks are presented in Figure 4. All four tasks demand for volunteers during the entire planning horizon. We further assume that there are 400 volunteers registered in the KUBAS system.*

Instance 2 *Two hours later, a new task occurs and we assume that 300 new volunteers have registered. The new instance consists of the five tasks flood control 1, flood control 2, flood control 3, flood control 4, and sandbag supply. Priorities and demands of all tasks can again be obtained from Figure 4. All five tasks demand for volunteers during the entire planning horizon. The 300 newly registered volunteers - in addition to the 400 volunteers from instance 1 - give us an instance with 700 volunteers and 16 task activities.*

Since there is no empirical data about volunteer behavior in the 2013 Saale flooding, we have to generate volunteer availabilities and capabilities randomly. We investigate a 24-hour planning horizon and divide it into six four-hour segments. For each of the segments, the probability for each volunteer to be available in the segment is set to $\frac{1}{3}$. This allows for multi-hour consecutive volunteer availabilities and leads to an average volunteer available time

[7] Note that $t = 0$ is an auxiliary time slot that represents the time of decision making where no new assignments are made. In particular, X_{ij0} is fully determined by o_{ij0}.

[8] In all other cases, we have $X_{ij(t-1)} - X_{ijt} + 1 \geq 1$ and therefore, constraint (9) is trivially fulfilled.

[9] Note that the left hand side in constraint (11) is always $\leq \tau_3$ because of constraint (6). Therefore, constraint (11) is trivially fulfilled in the other case where $X_{ijt} = 0$.

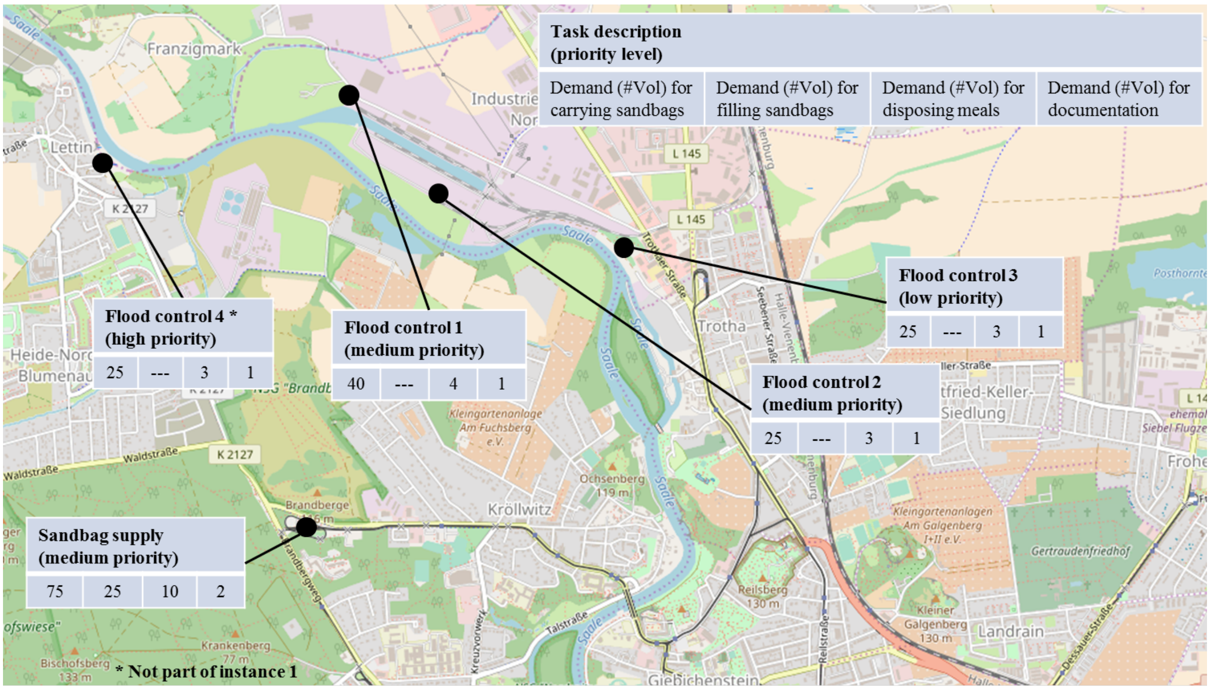

Figure 4. Task details for case study instances

Table 3. Setting of instance-independent parameters

Description	Value/Range/Distribution
Number of time slots ...	
... in entire planning horizon	$T = 24$ (representing 24 hours)
... that a volunteer must work consecutively on a task activity	$\tau_1 = 2$
... that a volunteer is allowed to work during planning horizon	$\tau_2 = 8$
... for setup/travel	$\tau_3 = 1$
Weight parameters which indicate the severity of unmet task demands	w_{pt} defined in equation (13)
Compensation factor which indicates the relative importance ...	
... between priorities *low* ($p = 1$) and *medium* ($p = 2$)	$\sigma_{12} = \frac{1}{3}$
... between priorities *medium* ($p = 2$) and *high* ($p = 3$)	$\sigma_{23} = 0$

of eight hours, which corresponds to a standard work day. We finally set the probability for a volunteer to offer capability *hard physical labor*, *medium physical labor*, *light physical labor*, and *writing* to 0.5, 0.7, 0.9, and 0.95, respectively. Note that these settings lead to an expected number of approximately 67 (for instance 1) and 117 (for instance 2) volunteers with capability *hard physical labor* in each four-hour segment. This implies a shortage of volunteers for the carrying sandbag activities, which have a cumulated demand for 165 (instance 1) and 190 (instance 2) volunteers.

Parameter Setting

In order to solve the presented instances using our optimization model, we specify all instance-independent parameters from Table 1 in this subsection. A summary is given in Table 3. The 24-hour planning horizon is represented by 24 one-hour time slots, i.e., $T = 24$. We set the minimum number of time slots that a volunteer must work consecutively on the same task activity to $\tau_1 = 2$. Furthermore, we set the maximum number of time slots that a volunteer is allowed to work in total during the entire planning horizon to $\tau_2 = 8$. Finally, we set the number of time slots that represent the setup/travel time between working on two different task activities to $\tau_3 = 1$.

Next, we set the weight factors w_{pt} that indicate the severity of unmet demands for volunteers in time slot t at task activities with priority p. According to requirement (ii), we have $w_{1t} < w_{2t} < w_{3t}$ for all time slots $t = 1, \ldots, 24$. Requirement (iii) implies $w_{pt} > w_{p,t+1}$ for all priorities $p = 1, 2, 3$ and time slots $t = 1, \ldots, 23$. Furthermore, high workloads in activities of low-priority tasks in earlier time slots are more important than high workloads in activities of medium-priority tasks in later time slots. We quantify this by setting $w_{1t} = w_{2,t+4}$ for all time slots $t = 1, \ldots, 20$.

[10] In addition, high workloads of activities of high-priority tasks are more important (even in the last time slot $t = 24$) than high workloads in any other task activities in any time slot $t = 1, \ldots, 24$. This means that $w_{3t} > w_{2t'}$ for all time slots $t, t' = 1, \ldots, 24$. The presented relations can be guaranteed by setting

$$w_{3t} = 25 \cdot \left(1 - \frac{t-1}{24}\right), \quad w_{2t} = 1 \cdot \left(1 - \frac{t-1}{24}\right), \quad w_{1t} = \frac{20}{24} \cdot \left(1 - \frac{t-1}{24}\right) \tag{13}$$

for all time slots $t = 1, \ldots, 24$.

Finally, we specify the compensation factors $\sigma_{p-1,p}$, which indicate the relative importance between adjacent priorities $p-1$ and p. The compensation factor between the priorities *low* and *medium* is $\sigma_{12} = \frac{1}{3}$. This means that in each time slot t and for each capability c, the workload L_{1ct} of activities at low-priority tasks must be at least one-third of the workload L_{2ct} of activities at medium-priority tasks. Furthermore, there is no such lower bound on the workloads L_{2ct} of activities of medium-priority tasks. In particular, there is no compensation between the priorities *medium* and *high*, i.e., $\sigma_{23} = 0$. This – in combination with the weight setting from equation (13) – implies that in each time slot, high-priority activities have to be filled with volunteers as good as possible before volunteers can be assigned to any other activities.

Computational Results and Discussion

Having outlined our parameter setting in the previous subsection, we solve both presented case study instances by calculating a near-optimal solution of their corresponding model instances. To achieve this, we use the off-the-shelf solver GUROBI 7.5 on a laptop with an Intel Core i5-3340M CPU @2.70GHz and 4 GiB RAM. The results are presented and discussed in the following.

We abort calculations for the first instance after 30 minutes where the provable gap (returned by the GUROBI solver) between the objective function value of the obtained solution and the optimal objective value was at most 0.08%. The best known solution was found after approximately 13 minutes, which is sufficient in practical disasters according to our interviews with the DMA Halle (Saale). The activities filling sandbags, disposing meals, and documentation are filled to 100% during the entire planning horizon $t = 1, \ldots, 24$. The reason is that they demand for a relatively low numbers of volunteers. The bottleneck in all tasks is the activity of carrying sandbags (capability $c = 1$). Therefore, we present only the workloads L_{p1t} for capability $c = 1$ in the upper half of Figure 5 and discuss them in the following. Workloads L_{31t} equal 0 in all time slots since there is no task with priority *high* ($p = 3$) in the first instance. All other workloads during the entire planning horizon $t = 1, \ldots, 24$ of the first instance are non-zero. However, they are all clearly below 1 because there are not enough volunteers to meet the entire demand in any time slot. One important trend is that the workloads are decreasing with increasing t. [11] The reason is that volunteer shortages in early time slots are more severe than in later time slots and consequently, our model prefers feasible schedules in which volunteers work on activities as early as possible. Since volunteers are not allowed to work for more than eight hours within the entire planning horizon, this leads to higher volunteer shortages in later time slots. Finally, the workloads of the sandbag filling activities of medium-priority tasks are at most three times higher than the workload of the corresponding activity of the low-priority task. This effect mirrors the compensation factor $\sigma_{12} = \frac{1}{3}$.

Two hours later, when the new task *flood control 4* occurs, we solve the second instance. Again, we abort calculation after 30 minutes where the provable gap (returned by the GUROBI solver) between the objective function value of the obtained solution and the optimal objective value was at most 0.01%. The best known solution was again found after approximately 13 minutes. As for the first instance, the only activities where there is a shortage of volunteers are carrying sandbags activities (capability $c = 1$) and consequently, the lower half of Figure 5 presents only the workloads L_{p1t} for capability $c = 1$. [12] We see that the activity of the high-priority task ($p = 3$) is filled to 100% during the entire planning horizon $t = 3, \ldots, 26$. This is forced by high weights w_{3t} for all time slots t and a zero compensation factor $\sigma_{23} = 0$ between the priorities *medium* and *high*. Note that assignments from the solution of the first instance are not changed due to constraint (12). Consequently, only volunteers that have registered after the solution of the first instance can work on the new activity in time slots $t = 3, \ldots, 24$. Furthermore, the activities of the medium- and low-priority tasks have higher workloads than in the solution of the first instance since not all newly registered volunteers are necessary to fill the high-priority activity to 100%. Again, workloads decrease with

[10] The skip from t to $t + 4$ is used because the volunteer availabilities are generated using four-hour segments.

[11] There are some exceptions, e.g. in $t = 17$, because the number of available volunteers is not constant over time due to the random generation of volunteer availabilities.

[12] Since the second instance is solved two time slots after the first instance, time of decision making $t = 0$ in the second instance is equal to $t = 2$ in the first instance. For the sake of presentation, however, we set the time of decision making in the second instance to $t = 2$ which leads to a 24-hour planning horizon of $t = 3, \ldots, 26$.

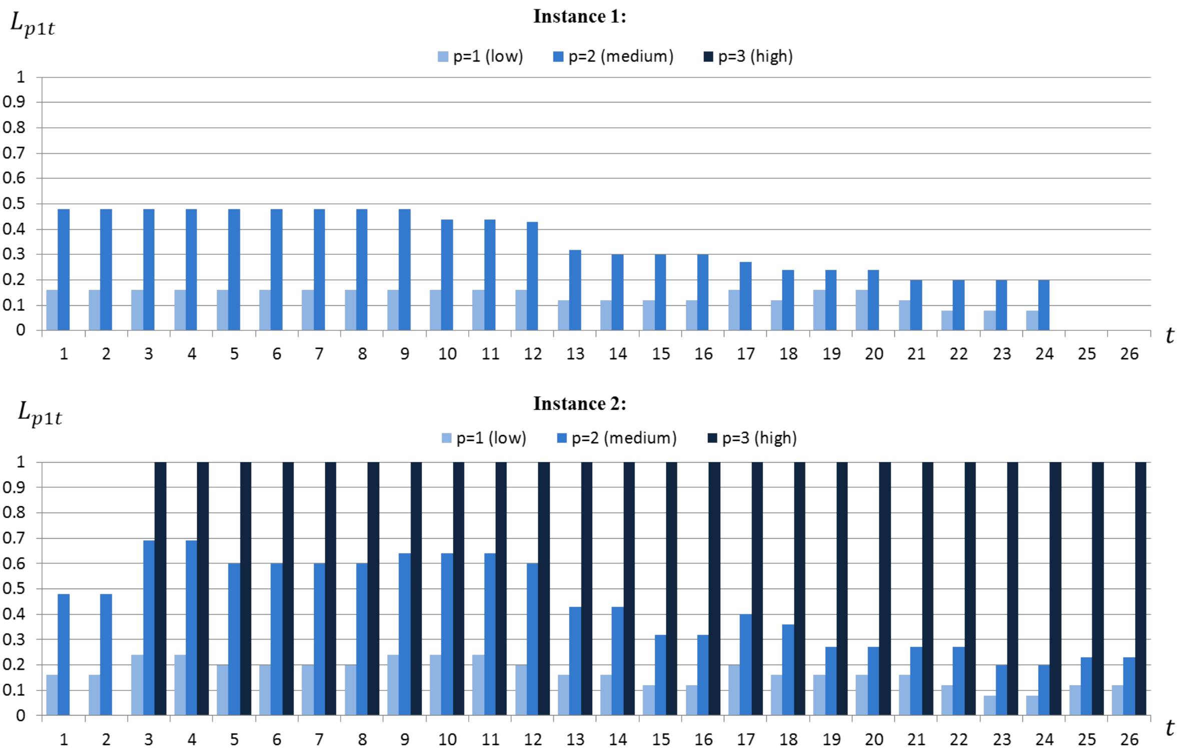

Figure 5. Workloads L_{pct} for all priorities p, capability $c = 1$, and time slots t for both instances

increasing t (with some exceptions, e.g. in $t = 17$) during the entire planning horizon $t = 3, \ldots, 26$ for the same reason as explained above for instance 1.

To sum up, our model can be used to obtain near-optimal solutions in a big real world case study scenario with 16 task activities and 700 volunteers within a few minutes. Beyond the computational evaluation in this section, we presented the solution of both instances to practitioners at the DMA Halle (Saale) for an empirical ex-post validation. They confirmed that our solutions meet their expectations on an effective coordination of volunteers and that it is hardly possible for professional responders to find better solutions manually - especially under severe time pressure during a disaster situation.

CONCLUSIONS

In this section, we present our conclusions. We sum up the findings and contributions of our paper before we provide several avenues for further research.

Summary

In disaster situations, there is often a large number of spontaneous (i.e., not affiliated to any humanitarian organization) volunteers that offer their help. These volunteers represent a precious resource that must not be neglected but their coordination is often a burden that distracts professional responders from their primary duties. In our paper, we develop and test a novel approach that uses an optimization model to automatize the coordination of spontaneous volunteers. We conducted interviews with professionals at the DMA Halle (Saale) to identify key requirements for coordinating spontaneous volunteers in real-world disasters. Based on these requirements, we develop a linear optimization model that can be used by off-the-shelf solvers to solve problem instances of realistic size near-optimal within short time. We confirm this in a case study where we use the model in combination with the GUROBI solver to calculate coordination decisions for problem instances that represent a real-world flood scenario. In addition to this computational evaluation, we presented the solutions of our case study instances to the DMA Halle (Saale) for an empirical ex-post evaluation in which our solutions were well accepted by practitioners. Our interview partners confirmed in particular that such complex coordination decisions can hardly be provided manually by professional responders in comparable quality under the severe time pressure of a disaster situation. Our optimization model is integrated in a holistic volunteer coordination system called *KUBAS*, which we develop in close cooperation with the DMA Halle (Saale) and esri Germany. This guarantees that our approach can be used to improve the coordination of spontaneous volunteers and to relieve professional responders during real disasters.

Future Work

There are several directions for future work. First, it would be interesting to analyze how our model works on larger scenarios, e.g. the 1985 earthquake in Mexico City where up to two million volunteers offered their help (Dynes et al. 1990). Second, further case studies may also be used to measure the robustness of our optimization model against parameter changes. Third, it may be convenient to develop heuristic solution algorithms to obtain good suboptimal solutions instead of using an off-the-shelf solver to obtain optimal solutions for computationally hard scheduling instances. Fourth, we aim at further developing the presented requirements and the model itself. For example, it may be convenient to balance work among volunteers, i.e., that total working hours of volunteers do not differ more than necessary. Real GPS-based (instead of constant) setup times could also be integrated into the model. Fifth, the performance of the entire KUBAS system – and of our scheduling approach in particular – should be evaluated when applied in live disaster situations.

ACKNOWLEDGMENTS

This research has been funded by the Federal Ministry of Education and Research of Germany in the framework of KUBAS (project number 13N13942). We express our special thanks to Dr. Robert Pulz – head of the Disaster Management Authority in Halle (Saale), Germany – for giving continuous feedback in personal interviews.

REFERENCES

Alexander, D. (2010). "The Voluntary Sector in Emergency Response and Civil Protection: Review and Recommendations". In: *International Journal of Emergency Management* 7.2, pp. 151–166.

Aman, A., Bakhtiar, T., Hanum, F., and Supriyo, P. T. (2012). "OR/MS Applications in Mt. Merapi Disaster Management". In: *Journal of Mathematics and Statistics* 8.2, pp. 264–273.

Auferbauer, D., Ganhör, R., Tellioglu, H., and Pielorz, J. (2016). "Crowdtasking: Field Study on a Crowdsourcing Alternative." In: *Proceedings of the 13th International Conference on Information Systems for Crisis Response and Management (ISCRAM).*

Betke, H. (2018). "A Volunteer Coordination System Approach for Crisis Committees". In: *Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM).*

Dynes, R. R., Quarantelli, E. L., and Wenger, D. (1990). *Individual and Organizational Response to the 1985 Earthquake in Mexico City, Mexico*. Tech. rep. Disaster Research Center.

Falasca, M. and Zobel, C. (2012). "An Optimization Model for Volunteer Assignments in Humanitarian Organizations". In: *Socio-Economic Planning Sciences* 46.4, pp. 250–260.

Falasca, M., Zobel, C., and Fetter, G. M. (2009). "An Optimization Model for Humanitarian Relief Volunteer Management". In: *Proceedings of the 6th International Conference on Information Systems for Crisis Response and Management (ISCRAM).*

Falasca, M., Zobel, C., and Ragsdale, C. (2011). "Helping a Small Development Organization Manage Volunteers more Efficiently". In: *Interfaces* 41.3, pp. 254–262.

FEMA (2013). *Volunteer and Donations Management Support Annex*. URL: https://www.fema.gov/media-library-data/20130726-1914-25045-9213/nrf_support_annex_volunteer_20130505.pdf.

Fernandez, L., Barbera, J., and Van Dorp, J. (2006). "Strategies for Managing Volunteers during Incident Response: A Systems Approach". In: *Homeland Security Affairs* 2.3.

Gordon, L. and Erkut, E. (2004). "Improving Volunteer Scheduling for the Edmonton Folk Festival". In: *Interfaces* 34.5, pp. 367–376.

Havlik, D., Pielorz, J., and Widera, A. (2016). "Interaction with Citizens Experiments: From Context-Aware Alerting to Crowdtasking". In: *Proceedings of the 13th International Conference on Information Systems for Crisis Response and Management (ISCRAM).*

Horita, F. and De Albuquerque, J. P. (2013). "An Approach to Support Decision-Making in Disaster Management based on Volunteer Geographic Information (VGI) and Spatial Decision Support Systems (SDSS)." In: *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM).*

IFRC (2015). *Annual Report 2015*. URL: http://media.ifrc.org/ifrc/wp-content/uploads/sites/5/2016/11/IFRC-Annual-Report-2015-EN_LR.pdf.

KUBAS (2018). *Coordination of Voluntary Helpers to Overcome Disaster Situations*. URL: https://kubas.uni-halle.de/en.

Lassiter, K., Alwahishie, A., and Taaffe, K. (2014). “Improving Volunteer Productivity and Retention during Humanitarian Relief Efforts”. In: *International Journal of Supply Chain Management* 3.2, pp. 1–10.

Lassiter, K., Khademi, A., and Taaffe, K. (2015). “A Robust Optimization Approach to Volunteer Management in Humanitarian Crises”. In: *International Journal of Production Economics* 163, pp. 97–111.

Lindner, S., Kuehnel, S., Betke, H., and Sackmann, S. (2018). “Simulating Spontaneous Volunteers - A Conceptual Model”. In: *Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*.

Liu, J. and Wang, S. (2013). “Building a new Global Search Algorithm for Managing the Chinese Volunteer Relief Organisations”. In: *International Journal of Emergency Management* 9.3, pp. 187–204.

Lodree, E. J. and Davis, L. B. (2016). “Empirical Analysis of Volunteer Convergence Following the 2011 Tornado Disaster in Tuscaloosa, Alabama”. In: *Natural Hazards* 84.2, pp. 1109–1135.

Lowe, S. and Fothergill, A. (2003). “A Need to Help: Emergent Volunteer Behavior after September 11th”. In: *Beyond September 11th: An Account of Post-Disaster Research*, pp. 293–314.

Mayorga, M. E., Lodree, E. J., and Wolczynski, J. (2017). “The Optimal Assignment of Spontaneous Volunteers”. In: *Journal of the Operational Research Society* 68.9, pp. 1106–1116.

Pielorz, J. and Lampert, C. H. (2015). “Optimal Geospatial Allocation of Volunteers for Crisis Management”. In: *Proceedings of the 2nd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pp. 152–158.

Sampson, S. E. (2006). “Optimization of Volunteer Labor Assignments”. In: *Journal of Operations Management* 24.4, pp. 363–377.

Schryen, G., Rauchecker, G., and Comes, T. (2015). “Resource Planning in Disaster Response: Decision Support Models and Methodologies”. In: *Business & Information Systems Engineering* 7.2, pp. 1–17.

Siemen, C., Rocha, R., van den Berg, R., Hellingrath, B., and De Albuquerque, J. P. (2017). “Collaboration among Humanitarian Relief Organizations and Volunteer Technical Communities: Identifying Research Opportunities and Challenges Through a Systematic Literature Review”. In: *Proceedings of the 14th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2013). “Personnel Scheduling: A Literature Review”. In: *European Journal of Operational Research* 226.3, pp. 367–385.

Van Gorp, A. F. (2014). “Integration of Volunteer and Technical Communities into the Humanitarian Aid Sector: Barriers to Collaboration.” In: *Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*.

Whittaker, J., McLennan, B., and Handmer, J. (2015). “A Review of Informal Volunteerism in Emergencies and Disasters: Definition, Opportunities and Challenges”. In: *International Journal of Disaster Risk Reduction* 13, pp. 358–368.