

Optimizing Emergency Supply for Mass Events

Dennis Güttinger¹ Eicke Godehardt¹ Andreas Zinnen²

¹SAP AG, SAP Research

²University of Luxembourg

¹{dennis.guettinger, eicke.godehardt}@sap.com

²andreas.zinnen@uni.lu

ABSTRACT

In this work we provide an approach for simulating and optimizing the emergency supply in a run-up to a mass event. For a given set of hospitals, transport vehicles and injured suffering from common injuries, our algorithm simulates the workload of provided transport and medication capacities, e.g., doctors. In addition to standard methods, our algorithm considers how a patient's individual waiting time until medication impacts the corresponding course of disease. We use Simulated Annealing with transition probabilities favoring a balanced workload of vehicles and doctors as optimization strategy. We show that using this strategy speeds up convergence and leads to better results compared to Greedy and standard Simulated Annealing with an underlying equal transition probability. Finally, we briefly discuss how different initialization strategies affect the performance of the provided algorithm.

Keywords

Public Security, Health Care, Simulated Annealing

1. INTRODUCTION

In recent years, the number and frequency of major events such as concerts, demonstrations and sport events have considerably increased. Additionally the average number of attendees is still growing (e.g., the audience of football games [9] or rock concerts). Apparently, the high number of attendants carries a tremendous risk. Apart from the danger of terrorist attacks, factors that at first glance appear trivial can cause mass panic with many injured people. Today, units involved in the treatment of injured such as police, fire brigade or doctors are frequently pushed to their limits (e.g., Love Parade disaster in 2010 with 21 victims [19], mass panic in the Heysel Arena [2] in 1985 with 39 victims and earthquake in Turkey in 1999 with 12000 victims [21]). Clearly, the preparation and planning of safety measures for mass events has gained in importance. In order to medicate all casualties as good and quickly as possible, the transport



Figure 1: Care of injured and crowd at music event. The high number of injured and difficult access to them poses new challenges for involved units.

and waiting times have to be minimized. In addition, the casualties need medical care by qualified personnel. One key factor of safety measures is the allocation of transport vehicles as well as capacities in hospitals. A determination of adequate demands before a specific mass event is non-trivial, and often the allocated capacities turn out to be not sufficient. The reasons for these deficits are twofold. First, organizers aim at a high profit. Thereby, they risk a poor treatment in case of unlikely accidents. Legal restrictions and simulations can help to address this problem. Second, the assignment of casualties to transport vehicles and hospitals is not optimized these days leading to an overall unbalanced workload. Nowadays, the selection of a hospital per injured is usually taken ad-hoc, sometimes even after the transport has already been started. For that purpose, the injured are categorized into injury groups. Patients with severe injury will be preferred. However, discussions with the chief officers of a health department at the city council have motivated the need for a more detailed classification. Often, a longer waiting time will affect the patients' state of health leading to a falsified prioritization. To improve the process, we propose the definition and consideration of functions over time for all injuries reflecting the corresponding course of disease (cf. section 3.1 for a detailed explanation of the groups and functions). The first main contribution of the paper is a new approach to efficiently simulate and optimize the emergency supply in a run-up to a mass event. The simulation tool optimizes the assignment of casualties to transport vehicles and physicians in hospitals considering the total waiting time until medication. Note that casualties are assigned only to those physicians who are able to medicate the specific type of injury. We apply Simulated Annealing along the line of Job Shop Scheduling problems (JSS) with transition probabilities favoring a balanced workload of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIMUTOOLS 2011, March 21-25, Barcelona, Spain

Copyright © 2011 ICST 978-1-936968-00-8

DOI 10.4108/icst.simutools.2011.245537

vehicles and doctors as optimization strategy. Subsequently, the results can be both used for identification of bottlenecks in patient care and construction of rough assignment plans. The second main contribution is the introduction of functions reflecting the course of disease for each injury. These functions prove to be a crucial factor during the optimization process. The third main contribution is an evaluation of our method simulating a practical example in the context of a football match at the arena of Frankfurt/Main during the FIFA soccer world cup 2006 in Germany. For the given example, we compare our method with a Greedy assignment strategy as applied nowadays and classic Simulated Annealing. The last contribution is a discussion how different initialization and transition probabilities effect the performance of Simulated Annealing. The remainder of this work is structured as follows. In Section 2 we will review related work. Section 3 first introduces the technique of triage and subsequently explains the concept of penalty functions. Thereafter the sources of data are described which we have used for our simulations. The transformation into a mathematical model and optimization algorithms are introduced in Section 4. Section 5 contains results of our empirical simulation. Section 6 concludes the paper and discusses future research.

2. RELATED WORK

Public security has gained a lot of interest in recent years. [15] use genetic algorithms and a vehicle assignment heuristic to solve a logistics model for delivery of critical items in a disaster relief operation. The public funded research projects SOGRO [17] (Instant rescue at big accident with masses of casualties) or SoKNOS [18] (Service-Oriented Architectures Supporting Networks of Public Security) address the optimized care of injured in disasters. While SOGRO focuses on techniques to shorten the first chaotic phase immediately after a major incident and before assignment of casualties to transport vehicles and hospitals, SoKNOS mainly deals with the adaptation of service oriented architectures, semantics and human adapted workplaces in public security. Information processing is also a major topic within SoKNOS, but especially with a focus on visualization. Several works deal with the special case of optimization and scheduling techniques in emergency and disaster scenarios. Most of them are special cases of network flow problems and none of them can be transformed into (coupled) Job Shop Scheduling problems as our problem [1, 3]. Simulated Annealing (SA) is an optimization heuristic first applied in thermodynamics [7] and later on several other combinatorial problems. [5] compute an approximately optimal solution to the traveling salesman problem. [20] gives a detailed overview of SA applications in general whereas [14] focus on OR problems in particular. The Job Shop Scheduling problem (JSS) was firstly presented by Graham [11] in 1966. For the special case of two machines and an arbitrary number of jobs this problem can efficiently be solved with Johnson's algorithm [13]. Garey [10] provided a proof in 1976 that JSS is NP-complete for more than two machines. For this general case many different approximation techniques for JSS have been suggested. These can roughly be divided into priority dispatch rules, bottleneck based heuristics, artificial intelligence, local search methods and meta-heuristics, see [12]. Simulated Annealing also belongs to the last group of local search methods. Van Laarhoven et al. were the first

ones who applied this technique on JSS [22]. There exist several works about emergency logistics planning in common. For example [24] have developed a time-space network model with the objective of minimizing the length of time needed for emergency repair, subject to related operating constraints. This leads to an integer network flow problem with side constraints and the authors have developed a heuristic algorithm to solve it. [3] deal with the problem of assigning units to tasks, but in a more general context. They have constructed a real-time decision support system which uses optimization methods, simulation and the judgement of the decision maker. Özdamar has extensively studied emergency logistics planning. In his first study about emergency logistics he has developed a hierarchical multi-criteria methodology for helicopter planning during a disaster relief operation in [1]. Some years later he has focused on the problem of dispatching commodities to distribution centres in emergency situations. The used mathematical model is a hybrid of a multi-commodity network flow problem and a vehicle routing problem. The authors solved it by coupling the sub-models with relaxed arc capacity constraints using Lagrangian relaxation.

Another work that deals with a vehicle assignment problem as we do is [16]. The authors present an approximate dynamic programming approach for making ambulance redeployment decisions in an emergency medical service system. However, while their primary decision variable is the place where idle ambulances should be redeployed to, in our scenario idle ambulances return back to disaster area by default.

Beyond that we are not aware of approaches similar to our that improve care of the injured in disasters through a mapping to Job Shop Scheduling problems. Thus, we have developed our own optimization framework to solve this problem. It will be introduced in detail in the next sections.

3. EXPERIMENTAL SETUP

In Section 1 we explain that a more sophisticated optimization of resources during an emergency is reasonable. In addition, we motivate that a classification into triage groups as used today is often not sufficient. An introduction of functions that reflect the course of injury was found useful in consultation with the chief officers of the health department at Frankfurt city council. Section 3.1 first summarizes the triage groups and how they are applied nowadays for categorization of injured. Next, we introduce 23 functions that will be considered while optimizing. In Section 3.2 we describe the training and test data sets used for evaluation of our method (cf. Section 4).

3.1 Triage Groups and Penalty Functions

If nowadays a high number of casualties have to be medicated at once, the injured are first divided into different categories (*triage categories*) to ensure an adequate supply. Patients with a higher priority are served first. These groups differ only slightly from country to country. [23] give a detailed overview about those differences. For the purpose of this paper, we use the definition of [6]:

T1: Urgent vital thread, e.g., respiratory insufficiency with asphyxia, massive (external) bleeding, heavy shock, serious thorax or abdomen trauma, face and/or respiratory system

burnings, tension pneumothorax.
 → immediate treatment necessary

T2: Seriously injured, e.g., craniocerebral injury, serious eye and/or face injury, instable luxation, exposed fracture/joint.
 → postponed treatment urgency, supervision

T3: Slightly injured, e.g., uncomplicated fracture, repositionable luxation, contusion, distortion, graze.
 → posterior treatment (ambulant, if applicable)

T4: Without any viability chance, dying, e.g., open craniocerebral injury with cerebral mass discharge, severest and extended burnings.
 → supervising, death awaiting treatment, terminal care

Often, a longer waiting time will affect the patients' state of health because the severity of many injuries depends on the therapy time. A tension pneumothorax, for example, is non-hazardous as far as treated immediately. But already after a short period of time without treatment, a tension pneumothorax can get life-threatening. Note that applying the concept of triage categories as described above fixes a priority for each casualty not considering the waiting time. Therefore assignment switches are not allowed and often leading to a falsified prioritization.

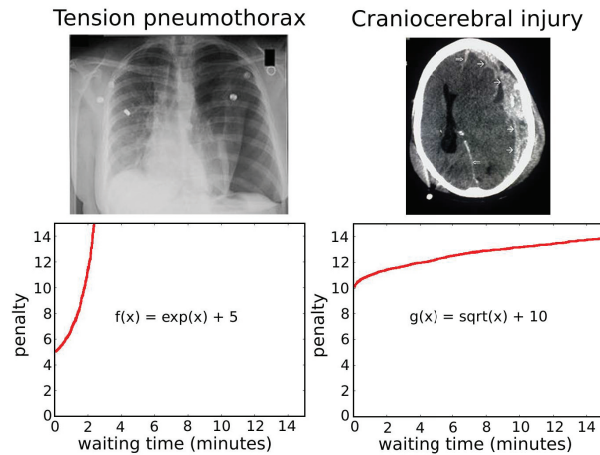


Figure 2: Example pictures and functions illustrating the course of disease in time

To improve the process, our optimization method considers functions (*penalty functions*) over time for all injuries describing the corresponding course of injury. The penalty functions f_y assign a penalty value to each injury $y \in Y$ (set of injuries) depending on the waiting time t until medication. Figure 2 illustrates the course of injury for a tension pneumothorax (left figure) and a craniocerebral injury (right figure). Obviously, these two diseases have a different initial priority and course. A long waiting time thus has a different impact. In case of pneumothorax, the damage for quick treatment is very low, but increases exponentially in time. Indeed the waiting time for a craniocerebral injury also influences the damage, but has nowhere the effect as for a tension pneumothorax. Overall, we define 23 different

sample functions for injuries such as luxation, craniocerebral injury, bleeding or serious dermal burn. These differ both in their initial priority ($t = 0$) and in the positive functional course (e.g., $f(x) = \ln(x + 1)$, $f(x) = x^2$, $f(x) = e^x - 1$, $f(x) = \sqrt{x}$). Clearly, a longer waiting time causes a higher penalty value for all functions. For later evaluation, we assign those 23 functions to the triage groups as defined before considering the value for $t = 0$ similar to initial priorities. 11 out of 23 functions are assigned to T1, 6 to T2 and the remaining 6 to T3. We do not consider T4 since those subjects do not have an impact on the overall result. We once more point out that an introduction of such functions is useful and feasible after consultation with experts. The chosen functions are, however, elected exemplary to show the power of our approach and need to be adapted by experts in realistic future scenarios.

3.2 Training and Test Data

This section introduces the training and test data that will later be used to show the feasibility of our method for optimization of emergency supply for mass events (cf. Sections 4 and 5). The factors affecting the optimization problem are manifold. Following values have to be specified both for training and for testing the algorithm: number of casualties and corresponding injuries, duration for medication for each casualty, number of hospitals including number and type of physicians, number of vehicles and a matrix specifying the transportation time between the place of catastrophe and the hospitals. Since we design our algorithm to work in arbitrary settings, the training will consider artificial data that is randomly drawn. For testing, we evaluate a disaster during a football game in Frankfurt, Germany.

3.2.1 Training Data

For training the parameters of Simulated Annealing (cf. Section 4), we have generated 50 instances of a random training set as follows. For each of the factors mentioned above (e.g., number of casualties, injuries), we randomly draw a value from a uniform distribution within reasonable intervals, see table 1. Because the training set is generated randomly and has a representative amount of training instances, the method should generalize to many problems in this domain. A consideration of all combinations over the factors is not realistic. The number of degrees of freedom is certainly too high. Later, we will see that a higher number of instances will not lead to improved results (cf. Section 5.2.2). Thus, a set of 50 different training instances is sufficient.

variable	left bound	right bound
number of casualties	200	1600
medication duration	5 minutes	2 hours
initial priority	10	1000
number of hospitals	3	20
number of beds	100	1000
number of physicians	50	500
number of vehicles	5	25
vehicle capacity	1	4
journey length to hospital	5 minutes	40 minutes

Table 1: Interval bounds for randomly generated input variables in the training set

100% (1000 casualties)			
T1	T2	T3	T4
20%	40%	20%	20%
(200 casualties)	(400 casualties)	(200 casualties)	(200 casualties)

Table 2: Distribution of 1000 casualties to triage groups.

3.2.2 Test Data

For testing, we use the arena disaster scenario at the football world cup 2006 as described in [8]. The number of casualties is fixed to 1000. The distribution between different triage groups (T1 to T4) has been assumed as 20%–40%–20%–20% as illustrated in Table 2. This leads to 400 casualties assigned to triage group T2 and 200 casualties to all remaining groups. Altogether, 800 casualties (ignore T4) with a significant chance for viability have to be served and considered in the optimization problem. In Section 3.1, we explore the affiliation between the 23 injuries and the triage groups. We are thus able to draw injuries for the casualties for each group randomly, leading to 800 casualties distributed to T1–T3 as specified by [8]. To obtain realistic values for the number of hospitals including the number and type of physicians and the number of available vehicles, we have collected data from 20 quality reports from different hospitals [4] around Frankfurt/Main, Germany. Finally, we estimate the transportation time matrix between the arena and the hospitals using a route planning tool. The medication duration for the randomly generated casualty test set is chosen in the same range as for the training set (between 5 minutes and 2 hours).

4. ALGORITHMS FOR OPTIMIZATION

In this section, we first define formally the optimization problem (cf. section 4.1). Then we describe two different algorithms that we apply to address the problem (cf. section 4.2).

4.1 Optimization Problem and Criteria

Figure 3 illustrates the previously described optimization problem including relationships between involved entities. Intuitively, our aim is to find an optimal assignment between casualties and transport vehicles and physicians such that all casualties can be medicated as good and quickly as possible with respect to their individual injuries. In the following we formally introduce the involved factors. Furthermore, we define the optimization function, which calculates the total damage over all patients. Clearly, the goal is to minimize this function.

Let V , P and H be a set of vehicles, physicians and hospitals, respectively. Furthermore let M be a set of assignment tuples (φ, ψ) , where φ maps each casualty $x \in X$ to a physician $p \in P$ and ψ maps each casualty to a vehicle $v \in V$, respectively. We define following optimization function:

$$F := \sum_{x \in X} f_x(w(x)), \quad (1)$$

with

$$w : X \rightarrow \mathbb{R}_{\geq 0}, w(x) := \gamma(x) + \delta(x) + \epsilon(x), \forall x \in X$$

The function first considers the total waiting time $w(x)$ until medication for casualty x . Subsequently, this waiting time serves as a parameter for the penalty function f_x describing the course of injury for the injury that casualty $x \in X$ is suffering from. The overall damage is the sum over all casualties' single damage $f_x(w(x))$. The waiting time itself in turn is the sum of:

- $\gamma : X \rightarrow \mathbb{R}_{\geq 0}$: Waiting time until transport
- $\delta : X \rightarrow \mathbb{R}_{\geq 0}$: Transport duration for each casualty
- $\epsilon : X \rightarrow \mathbb{R}_{\geq 0}$: Medication waiting time in hospital

The objective is to find an optimal mapping of casualties to physicians and vehicles. Note that $(\varphi, \psi) \in M$ only assigns casualties to physicians and vehicles, but does not define an order of treatment. Therefore, we additionally define the functions $(ord_p)_{p \in P}$, $(ord_v)_{v \in V}$ with

$$\begin{aligned} ord_p : \varphi^{-1}(p) &\rightarrow \{1, 2, \dots, |\varphi^{-1}(p)|\}, \\ ord_v : \psi^{-1}(v) &\rightarrow \{1, 2, \dots, |\psi^{-1}(v)|\}. \end{aligned}$$

The functions define for all physicians p and vehicles v a sequence of treatment for patients that are either assigned to the specific doctor ($\varphi^{-1}(p)$) or vehicle ($\psi^{-1}(v)$). The mappings (φ, ψ) and orderings $(ord_p)_{p \in P}$ and $(ord_v)_{v \in V}$ should be chosen such that they minimize F and coevally satisfy the following two conditions:

- C1: The total number of medicated casualties $\lambda(h)$ in a hospital does not exceed its number of available beds $\beta(h)$ for all hospitals $h \in H$.
- C2: A casualty x is assigned to a qualified physician $\varphi(x) : \varphi(x) \in \pi(\iota(x)), \forall x \in X$, with classification function $\iota : X \rightarrow Y$ mapping a casualty to an injury and $\pi : Y \rightarrow \mathcal{P}(P)$ assigning injuries to the set of qualified doctors.

Note that the ordering sequences $(ord_p)_{p \in P}$ and $(ord_v)_{v \in V}$ are bijective, since each casualty has exactly one position in assigned transport and medication queue. It can be easily seen that in the worst case (each physician $p \in P$ can medicate each type of injury and number of available beds is greater than total number of casualties in each hospital) there are $|P|^n \cdot |V|^n$ different combinations of feasible maps φ and ψ and at most $n! \cdot n!$ possible bijective orderings for each $(\varphi, \psi) \in M$. Thus, altogether we obtain $\mathcal{O}(n^n)$ different combinations. In addition, the optimization problem is non-linear since the objective function is non-linear: Obviously equation (1) can contain non-linear penalty functions f_x and thus using linear optimization algorithms like Simplex or interior point for solving our optimization problem is impossible. Therefore we have decided to use Greedy and Simulated Annealing as optimization techniques which are completely independent from the objective function characteristics.

4.2 Optimization Algorithms

Two different algorithms are used to address the introduced optimization problem. Greedy is the algorithm that is used

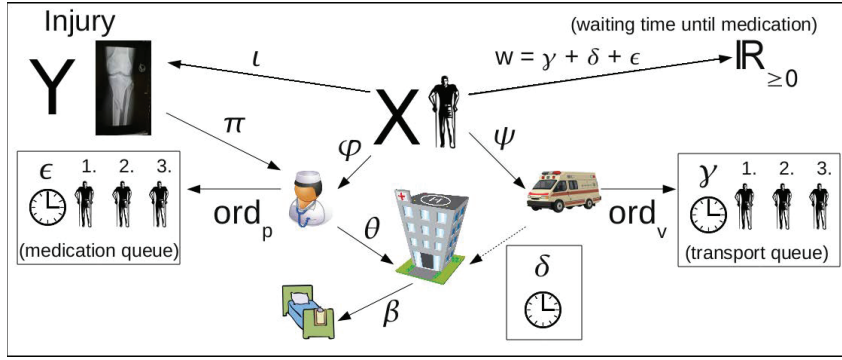


Figure 3: Interrelations between involved entities in emergency supply for mass events. Core entities are casualties (X) with injury Y mapped to a vehicle and a doctor, that in turn belongs to a specific hospital. The optimization is performed subject to waiting time w , doctors’ qualifications and hospitals’ capacity β

today. The casualties are classified into triage groups considering the injury. Subsequently, patients in poor condition are preferred. Once more, the triage groups only consider the initial priority of the patients’ health. In addition, Simulated Annealing takes the course of disease into account, as illustrated in section 3.1. The following sections give a short introduction to the algorithms.

4.2.1 Greedy

Greedy takes the following steps for optimizing the motivated problem:

In the first step casualties are sorted descending by their initial priority value (casualties with highest initial priority first). The initial priority is given by the correlated penalty function. Second, the algorithm maps the current patient to the most readily available vehicle. After that, the patient is taken to the hospital where it is treated fastest. Note that the hospital must not be fully loaded (C1) and needs qualified and available doctors who can treat the patient (C2). The resulting mapping and ordering can be used straightforward to calculate the resulting value for our optimization function defined by Equation (1).

4.2.2 Simulated Annealing

Simulated Annealing is a common technique for minimization of functions whose exact analytical characteristics are unknown due to high complexity. Each step of the Simulated Annealing algorithm replaces the current solution by a random nearby solution, chosen with a probability that depends both on the difference between the corresponding function values and also on a global parameter T (called the temperature), that is gradually decreased (degression coefficient) during the process. The dependency is such that the current solution changes almost randomly when T is large, but increasingly downhill as T goes to zero. The allowance for uphill moves saves the method from becoming stuck at local optima. In our setting a solution S is an arbitrary assignment of casualties to vehicles and physicians. A nearby solution is one with the same assignments except that one casualty is assigned to a different vehicle and/or physician or another queuing position. The performance of Simulated Annealing is carried mainly by the initial temperature T , the degression coefficient and the number of runs for each

temperature. Section 5.1 will summarize how to train adequate values for those parameters. Two factors have high impact on an efficient calculation of a good approximation to the global optimum. First, a good initialization is crucial for a fast convergence. Second, a sophisticated generation of nearby solutions will strongly effect the speed of convergence.

Initialization.

In this section, we will introduce two different ways for initializing Simulated Annealing. First, the assignment can be randomly. In each step one arbitrary casualty is chosen and a vehicle as well as a physician is assigned to this casualty, such that constraints C1 and C2 are satisfied. If a casualty is assigned to a vehicle (physician) that already transports (medicates) other casualties, the new casualty is enqueued at the end of the transport (medication) list of this vehicle (physician). Second, the assignment can be done applying Greedy. Obviously, a random initialization will often not lead to good assignments to start the algorithm. In our specific scenario, we expect an optimal solution that does not differ to a high degree from assignments calculated by Greedy (cf. 4.2.1). Intuitively, an initialization by Greedy should therefore allow for faster convergence since the algorithm starts closer to the optimal solution. Random initialization can be done in linear time with respect to number of casualties (that is $\mathcal{O}(n)$), whereas Greedy initialization needs $\mathcal{O}(n^2)$ steps. This difference does not have an impact on computation time for small number of casualties (e.g., $n < 100$). For a size of $n = 800$ casualties as predefined in our example scenario [8] a random initialization on a centrino processor runs less than one second and a Greedy-Initialization takes about 5 seconds. Since this additional effort on computation time seems to be reasonable, Greedy is the preferred choice for initialization.

Generation of Nearby Solutions.

As mentioned before, a good generation of nearby solutions will strongly effect the algorithm’s speed of convergence. The current solution S of the algorithm can be defined as a quadruple

$$(\varphi, \psi, (ord_p)_{p \in P}, (ord_v)_{v \in V})$$

with mapping of casualties to vehicles and doctors $(\varphi, \psi) \in M$ and the corresponding orders $(ord_p)_{p \in P} \in O_\varphi$ and $(ord_v)_{v \in V} \in O_\psi$. A nearby solution is one with the same assignments except that one casualty is assigned to a different vehicle and/or physician or another queuing position. In this paper, we use two approaches to generate nearby solutions. The first approach chooses nearby solutions with equal probability, namely one divided by the total number of nearby solutions (random transition). The second approach is based on following observations:

- A reassignment of casualties with higher damage will conjecturally have a greater effect on the global optimization function F and thus speed up convergence of Simulated Annealing. Therefore we define the probability for reassigning $x \in X$ in a specific Simulated Annealing step by the ratio of her current damage $f_x(w(x))$ and the overall damage $F(S)$ for the current solution S . Remember w as the function for waiting time and f_x as the function for the corresponding course of disease for casualty x .

$$\mathbb{P}(x) = \frac{f_x(w(x))}{F(S)}$$

- The probabilities for assignment of a chosen casualty x to a specific vehicle result from similar considerations. It seems plausible that an assignment of x to a vehicle $v \in V$ with low workload will improve the optimization. In order to estimate the workload of a specific vehicle v , we sum up the damage of all patients $\psi^{-1}(v)$ that are assigned to the specific vehicle (negative term of the numerator). By subtracting this damage from the overall damage $F(S)$, we obtain an anti-proportional dependency leading to low values if the vehicle is fully loaded. Finally, we normalize by the sum of the corresponding numerators for all vehicles.

$$\mathbb{P}(v) = \frac{F(S) - \sum_{x \in \psi^{-1}(v)} f_x(w(x))}{\sum_{\hat{v} \in V} (F(S) - \sum_{x \in \psi^{-1}(\hat{v})} f_x(w(x)))} \quad (2)$$

- Similarly, we define probabilities for the reassignment of the chosen casualty x to qualified doctors. Once more, an assignment of x to a physician p with low workload will probably improve the optimization. Let $P_x = \pi(\iota(x))$ be the set of physicians that are qualified to medicate the injury of x .

$$D_x = \sum_{\hat{p} \in P_x} \sum_{\hat{x} \in \pi^{-1}(\hat{p})} f_x(w(\hat{x}))$$

gives the overall damage of all casualties \hat{x} that are assigned to qualified doctors for x 's injury. The probability for assigning a casualty x to a physician p is calculated as follows:

$$\mathbb{P}(p | x) = \begin{cases} 0 & \text{if } p \notin P_x \\ \frac{D_x - \sum_{\hat{x} \in \varphi^{-1}(p)} f_x(w(\hat{x}))}{\sum_{\hat{p} \in P_x} (D_x - \sum_{\hat{x} \in \varphi^{-1}(\hat{p})} f_x(w(\hat{x})))} & \text{if } p \in P_x \end{cases}$$

If $p \notin P_x$, an assignment of x to p is not feasible because physician p has not the necessary qualification. If p has the qualification, the probability is high in case of a low workload for p . Once more, the negative term of the numerator specifies the damage of all casualties

that are mapped to the specific physician. By subtracting this value from the overall damage of patients medicated by qualified physicians, the numerator will be higher for doctors with a low workload. The denominator normalizes the specific values by summing up the corresponding values for all qualified physicians $\hat{p} \in P_x$.

5. EXPERIMENTS AND RESULTS

In the following we make two different experiments to evaluate the presented methods. The first experiment compares Greedy with Simulated Annealing. As initialization for Simulated Annealing we use the Greedy results. To calculate nearby solutions, we use the strategy that takes into account the workload of doctors and vehicles. The second experiment discussed how different parameters influence the performance of Simulated Annealing.

5.1 Training and Testing

Greedy (cf. Section 4.2.1) requires no training phase. We can apply the algorithm directly to find a solution for the test data described in Section 3.2.2.

We apply Simulated Annealing with Greedy initialization and transition probabilities considering the vehicles' and physicians' workload. Section 4.2.2 explains Simulated Annealing in detail.

5.1.1 Training

The performance of Simulated Annealing depends strongly on the choice of initial temperature, coefficient of scale, and number of runs for each temperature. In order to find adequate parameters for our scenario, we continuously generate 1000 different parameter triples randomly and calculate for each triple and each instance of the training data (cf. Section 3.2.1) an assignment of casualties to vehicles and hospitals with Simulated Annealing. The triple leading to the minimum standardized average damage is saved and used later in the test. Note that results for different number of casualties cannot directly be compared. Therefore, we normalize with respect to number of casualties. In Section 5.2.2, we will show that 100 triples are already sufficient because a consideration of up to 1000 does not significantly improve the results. The procedure of generating and evaluating new triples is stopped as soon as no significant improvement can be observed.

5.1.2 Testing

In the test phase, we apply Simulated Annealing to the test data from Section 3.2.2 using the parameter triple trained in the previous section.

5.1.3 Results

The Greedy solution can be improved significantly by an additional optimization with Simulated Annealing. Figure 4 illustrates the performance (F) of both algorithms over time (mind the logarithmic scale of the F -axis). Greedy elapses after 5 seconds. Already after additional 1–2 seconds of computation, Simulated Annealing converges thereby decreasing the optimization value significantly by over 99% in comparison to Greedy (compare values in Table 4). The main reason for this result is the fact that Greedy only considers initial

technique	F after init	F finally	number of iterations	elapsed time
Random Complete	1877888798004211200.00	187370.598831	15080	3 seconds
Systematic Complete	16225142.25	53306.985539	183551	40 seconds
Hybrid I	1877888798004211200.00	52913.271735	165388	37 seconds
Hybrid II	16225142.25	61652.378285	237088	40 seconds

Table 3: Optimization results on test set for all four configuration of SA

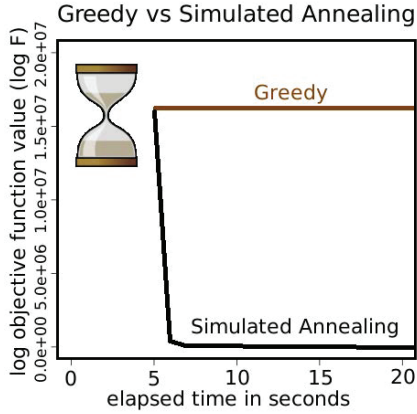


Figure 4: Overall damage over time when applying Greedy and Simulated Annealing on test set.

priorities of casualties when computing an assignment to transport vehicles and physicians. The temporal course of disease that is influenced by the waiting time until medication is completely neglected.

technique	F finally	elapsed time
Greedy	16225142.251736	5 seconds
Simulated Annealing	53306.985539	7 seconds

Table 4: Optimization results when applying Greedy and Simulated Annealing on the test data set.

Simulated Annealing, in contrast, takes the course of disease into account. Although the time with the second approach is higher, Simulated Annealing achieves excellent results in less than 10 seconds (passable time frame for scheduling). As shown later, the good performance results mainly from the sophisticated generation of nearby solutions.

5.2 Discussion of Simulated Annealing

The following discussion of Simulated Annealing is twofold. First, we will compare four optimization strategies based on Simulated Annealing for assignment of casualties to vehicles and physicians (cf. Section 5.2.1). Second, we will discuss how the number of parameter triples during training and the number of training instances influence the performance of Simulated Annealing (cf. Section 5.2.2).

5.2.1 Initialization and Transition Matrix

In Section 4.2.2 we introduce Simulated Annealing. In this section, we discuss the effect of two kinds of initializations

(random and Greedy) and two ways for generate nearby solutions (random and considering workload, cf. Section 4.2.2). The training and testing are similar to the description in Section 5.1. In the following, we compare the performance of the following configurations of Simulated Annealing on the test and training data:

1. Random initialization/Random transition (“Random Complete”)
2. Greedy initialization/Workload-aware transition (“Systematic Complete”)
3. Random initialization/Workload-aware transition (“Hybrid I”)
4. Greedy initialization/Random transition (“Hybrid II”)

To evaluate performance of the four SA configurations in our scenario we first apply SA on the training instances. The results are shown in Figure 5. The average damage with respect to number of training instances is displayed on the ordinate axis.

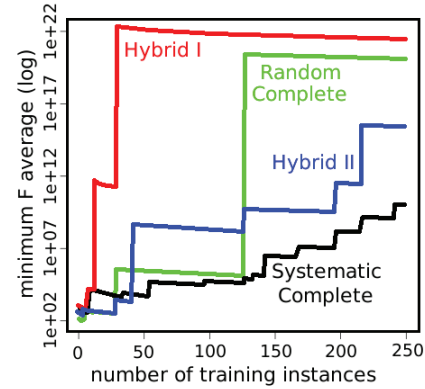


Figure 5: Average damage F for varying number of training instances and fixed optimal parameters.

On average Systematic Complete performs best on all training instances, as indicated by the low average damage. One can derive from three larger steps in the red curve that a random initialization in Hybrid I results in very bad optimization value for three training cases. Note that for the corresponding instances, the algorithm gets stuck in a bad local minimum leading to a high damage thereby escalating the average damage (cf. Figure 5). The same issue also applies

to Random Complete due to a random initialization. Hybrid II is both error-prone to difficult training instances and additionally achieves worse average results than Systematic Complete. An evaluation of these difficult instances shows that they are caused by a full workload of a specific doctor without alternatives. This fact clearly leads to high overall damage. Although such cases are exceptional they are nevertheless important and have to be taken into account. A visual inspection of the four curves emphasizes that Systematic Complete is much more robust (lower leaps). Since we aim for a method that is general enough to apply to as many settings as possible (and an unbalanced distribution of injured to doctors is realistic for disasters), Systematic Complete is the most robust choice. This choice is reinforced by the algorithms' runtime. Table 5 summarizes the average computation times for all four configurations. Obviously, the configurations considering the sophisticated and workload-aware search strategy converge much faster in less than half the time. Because of the Greedy initialization, Systematic Complete is even 3 seconds faster than Hybrid I on average. It seems that Systematic Complete, caused by the Greedy initialization and the sophisticated generation of nearby solutions, both starts closer to the optimum and converges faster.

	technique	average computation time
1	Random Complete	34.4113 seconds
2	Systematic Complete	10.0409 seconds
3	Hybrid I	13.5718 seconds
4	Hybrid II	26.0926 seconds

Table 5: Average optimization time till convergence on training set for all four configurations.

Initialization by Greedy takes 2.3 seconds on average while a random initialization is computed in less than one second (see Table 6). Due to the virtues of a Greedy initialization in combination with a workload-aware searching strategy a total computation time of less than 15 seconds is realistic and sufficient.

technique	average computation time
random initialization	0.215 seconds
Greedy initialization	2.34965 seconds

Table 6: Average computation times for random and Greedy initialization.

The results were additionally validated on the test set (see Table 3). The overall damage after initialization is best for Systematic Complete and Hybrid II, as one can expect because of the Greedy initialization. The algorithms considering the workload-aware transitions significantly outperform the remaining. Additionally, the runtime till convergence is lower for methods considering the sophisticated generation of nearby solutions (Random Complete is not considered because of poor results). Although Hybrid II performs slightly better than Systematic Complete both for duration and the optimization result on that specific test instance, we have shown before that Systematic Complete is much more robust and therefore first choice.

5.2.2 Important Factors while Training SA

In this section we investigate the changes of the average damage when increasing the number of parameter triples and training instances. Therefore we have chosen 50 different training instances and 1000 parameter triples in total for training of Simulated Annealing so far. Figure 6 shows the performance measured by the average damage of Systematic Complete with respect to the number of generated parameter triples. We can see that the average damage converges quickly already after 10 generated parameter triples. Hence, we just need to test a few different parameter triples to find a good one that performs well on all 50 training instances.

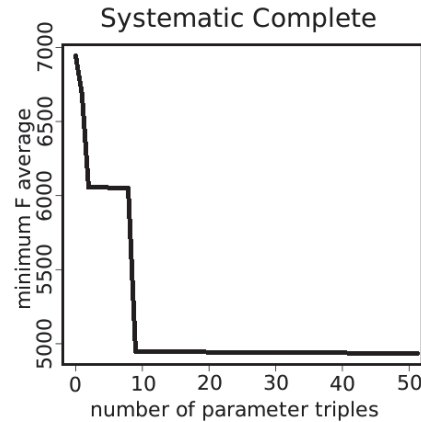


Figure 6: F-average for Systematic Complete over number of parameter triples for training.

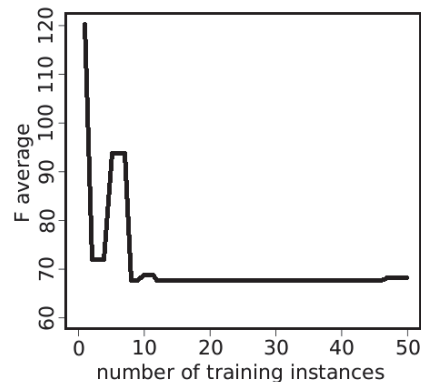


Figure 7: F-average for Systematic Complete over number of training instances.

Figure 7 shows the average damage on the test set with respect to number of randomly generated training instances. For each number of generated training instances the currently optimal parameter triple is applied on the test set and the resulting damage is displayed on the ordinate axis in Figure 7. Similar to Figure 6 we can see only minor changes of the average damage for more than 10 instances. Thus, 50 training instances are sufficient as representative set for finding a parameter triple that performs well on an arbitrary test instance.

6. CONCLUSION

In this work we provide an approach based on Simulated Annealing for simulating and optimizing the emergency supply in a run-up to a mass event. For a given set of hospitals, transport vehicles and injured suffering from common injuries, our algorithm simulates the workload of provided transport and medication capacities. In addition to standard Greedy methods, our algorithm considers how a patient's individual waiting time until medication impacts the corresponding course of disease by definition of penalty functions. The simulation results can subsequently be used to adjust the capacities. We adapt the transition probabilities favoring a balanced workload of vehicles and doctors as optimization strategy. We show that using this strategy speeds up convergence and leads to better results of up to 99% compared to Greedy and 50% to standard Simulated Annealing with an underlying equal transition probability. We briefly discuss that initialization using Greedy clearly outperforms a random initialization, because with Greedy the SA iterations start from an initial state that is much more close to the optimum than for random initialization. Finally, we evaluate the performance of Simulated Annealing in different settings. The combination of a Greedy initialization and workload adapted transition probabilities turns out to be the most robust setting on many training instances whereas other settings that involve random initialization and/or equally distributed transition probabilities perform worse especially on "extreme" training instances. For future research we will try to find more realistic penalty functions in cooperation with medical expert staff. Furthermore we are going to try out other training strategies and optimization techniques like genetic approaches. Another issue will be to find an appropriate relaxation of our assignment problem that provides us a lower bound for global damage. This would allow for a better performance evaluation and comparison of different optimization techniques.

7. ACKNOWLEDGMENTS

We want to express our gratitude to the chief officers of the health department at Frankfurt city council who helped us to validate our introduction scenario and supported us with further useful information about practical process sequences in disaster scenarios.

The project SoKNOS-Dialog was funded by means of the German Federal Ministry of Education and Research under the promotional reference "01IS07009". The authors take the responsibility for the contents.

8. REFERENCES

- [1] G. Barbarosoglu, L. Özdamar, and A. Cevik. An Interactive Approach for Hierarchical Analysis of Helicopter Logistics in Disaster Relief Operations. *European J. of OR*, 140, 2002.
- [2] BBC News. The Heysel disaster, 2000.
- [3] G. Brown and A. Vassiliou. Optimizing Disaster Relief: Real-Time Operational and Tactical Decision Support. *Naval Research Logistics*, 40, 2006.
- [4] Bundesministerium der Justiz. Fünftes Sozialgesetzbuch (SGB V): §137, 1998.
- [5] V. Cerny. A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm. *J. of Optimization Theory and Applications*, 45, 1985.
- [6] U. Crespin and G. Neff. *Handbuch der Sichtung*. Stumpf & Kossendey-Verlag, Edeweicht, Germany, 2000.
- [7] A. Das and B. K. Chakrabati. *Quantum Annealing and Related Optimization Methods*, volume 679. Springer Verlag, Heidelberg, Germany, 2005.
- [8] Der Magistrat der Stadt Frankfurt am Main. Medizinisches Schutzkonzept im Rahmen der Fußballweltmeisterschaft 2006, 2006.
- [9] DFL Deutsche Fußball Liga GmbH. Bundesliga 2010: Die wirtschaftliche Situation im Lizenzfußball, 2010.
- [10] M. R. Garey. The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research*, 1:117–129, 1990.
- [11] R. Graham. Bounds for Certain Multiprocessing Anomalies. *Bell System Technical Journal*, 54:1563–1581, 1966.
- [12] A. S. Jain and S. Meeran. A State-of-the-art Review of Job-Shop Scheduling Techniques. *European Journal of Operations Research*, 113:390–434, 1999.
- [13] S. M. Johnson. Optimal Two- and Three-Stage Production Schedules with Set-Up Times Included. *Naval Research Logistics Quarterly*, 1:61–68, 1994.
- [14] C. Koulamas, S. R. Antony, and R. Jaen. A Survey of Simulated Annealing Applications to Operations Research Problems. *Omega Journal*, 22:41–56, 1994.
- [15] Y.-H. Lin, R. Batta, and P. Rogerson. A Logistics Model for Delivery of Critical Items in a Disaster Relief Operation: Heuristic Approaches. Working Paper, 2009.
- [16] M. Maxwell, M. Restrepo, and S. Henderson. Approximate Dynamic Programming for Ambulance Redeployment. *INFORMS Journal on Computing*, 22(2):266–281, 2010.
- [17] SOGRO. Sofortrettung bei Großunfall, <http://www.c-lab.de/en/research-projects/sogro/index.html>.
- [18] SoKNOS. Service-orientierte Architekturen zur Unterstützung von Netzwerken im Rahmen öffentlicher Sicherheit, <http://www.soknos.de/>, 2009.
- [19] Staatsanwaltschaft Duisburg. Loveparade – Gemeinsame Presseerklärung von Staatsanwaltschaft und Polizei, July 2010.
- [20] B. Suman and P. Kumar. A Survey of Simulated Annealing as a Tool for Single and Multiobjective Optimization. *Journal of the Operational Research Society*, 57:1143–1160, 2006.
- [21] A. Tang. Izmit (Kocaeli), Turkey, earthquake on August 17, 1999. *Technical Council of Lifeline Earthquake Engineering Monograph*, 2000.
- [22] P. J. M. van Laarhoven, E. Aarts, and J. Lenstra. Job Shop Scheduling by Simulated Annealing. *Operations Research*, 40:113–125, 1992.
- [23] J. Windle, K. Mackway-Jones, and J. Marsden. *Emergency Triage*. Blackwell Publishers, Cambridge, England, 2006.
- [24] S. Yan and Y.-L. Shih. A Time-Space Network Model for Work Team Scheduling after a Major Disaster. *Journal of the Chinese Institute of Engineers*, 30:63–75, 2007.