

A modified ant colony optimization algorithm to increase the speed of the road network recovery process after disasters

Rostislav Vodák^{a,b}, Michal Bíl^{a,*}, Zuzana Křivánská^a

^a CDV – Transport Research Centre, Líšeňská 33a, Brno 636 00, Czech Republic

^b DMA and AM, Faculty of Science, Palacky University, 17. listopadu 1192/12, Olomouc 771 46, Czech Republic



ARTICLE INFO

Keywords:

Road network
Ant colony optimization
Resilience
Connectivity
Recovery
Disaster

ABSTRACT

When a disaster strikes many roads are blocked and the affected network may break up into a number of isolated parts. The reconnection of the network is therefore necessary for both relief distribution and planning of construction work. Shortening the time during which the road network is separated into isolated parts helps decrease indirect losses from disasters. The obstacles usually faced during the process of reconstruction include both the large number of blocked links and extensive affected areas (road networks).

A reduction of the network into a much smaller complete graph and metaheuristic based on an ant colony optimization has been introduced to overcome this issue. We demonstrate that, for small networks, the metaheuristic produces the same results as other deterministic algorithms. We further show that the method is still a viable approach for large networks (723 nodes and 974 links, where we artificially blocked 46 links) when the NP-hard nature of this problem began to affect the computational time of the deterministic algorithms.

We demonstrate how the various scenarios can be included into the algorithm. We finally introduce a new ranking of feasible solutions which enables the algorithm to minimize the time of reconstructions for all repair units. Reasonable results were obtained after five minutes of computation. There is nevertheless an up-to-38% improvement of the initial solution. The algorithm can also be used for both relief distribution, when no roads were damaged, and for planning of construction work when damaged roads occur.

1. Introduction

Transportation networks are particularly vulnerable to extreme events with a large spatial extent (usually natural disasters) after which many links might remain closed. These links can be not only blocked by, e.g., falling trees or temporal flooding, but also destroyed as a result of landsliding, fluvial erosion or earthquakes. These simultaneous road closures often result in network disintegration into a number of mutually isolated parts (components). Reestablishing network connectivity is among the most important tasks after such events because of relief distribution and the minimization of economic losses. The duration of this process is to a high degree dependent on the sequence in which the blocked links are (albeit provisionally) reopened. An optimal sequence of repair works will significantly shorten the time needed for network recovery. Shortening the time needed for reconnection of a road network damaged by a disaster ranks among the measures which not only reduce the impact of the disaster but also the disaster risk understood as the probability of the disaster multiplied by its consequences.

The problem of the optimal reconstruction is becoming increasingly

important due to limited resources [22,63] and the importance of the disaster recovery was highlighted in Altay and Green [4] or Ergun et al. [22].

The phase of recovery of a network damaged by an event is an important part of the resilience of the network. Two essential definitions of resilience exist. The first one defines resilience as the capability of a system to recover from a large disruption [34], while the other one adds the ability of the system to withstand the disruption with a low impact on its functioning [17,46]. An example of the network resilience, in the form of the time from the event to the recovery, can be seen in Fig. 1.

The figure also demonstrates the effect of the improved resilience. Notwithstanding the definition, one of the main tasks is thus to analyze and improve the resilience of the network (see for instance [64,71], as the most recent examples). The improvement and analysis of the resilience can, however, face many difficulties. One of the problems is how to evaluate the resilience of the network under the numerous uncertainties caused by a disaster. It led to the development of uncertainty based models for road network reliability in [54], which can be

* Corresponding author.

E-mail address: michal.bil@cdv.cz (M. Bíl).

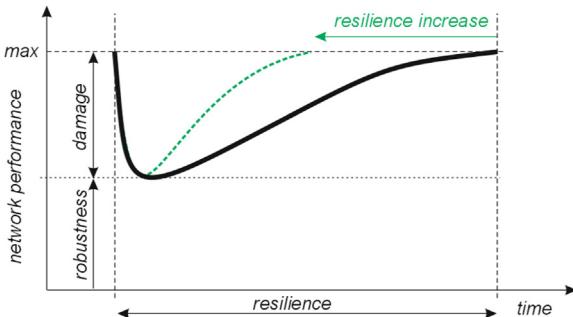


Fig. 1. An example of the network resilience.

understood as **an important feature of the resilience**. The main evaluating criteria in the paper are the total travel time, flow and consumer surplus. We also refer the reader to [55], where the post-disaster uncertainties are included in physical road capacity, parameters of link travel time function, travel demand and mode choice behavior. The authors studied the influence of several parameters on the final reliability of the network.

In this paper we work on an actual road network which developed its internal structure over time where every change in topology, e.g., construction of new road links, would be an extremely expensive process. The same is valid for a number of existing road networks around the world. The next issue is that many of the disruptions are event-dependent, which means that the ability of the system to restore its functioning can differ significantly under various scenarios (e.g., if only flooding or also landsliding is taking place). We thus pay attention only to the recovery phase. Enormous potential exists in the methods of optimization of the reconstruction process (after disruptive events) which could save resources and shorten the time to the network recovery.

The optimization problem is now discussed more thoroughly. One can assume that we have **a system represented by a road network which was affected by an extreme event**. The result is usually **a large number of concurrently blocked links** [12,7,8]. Common traffic patterns are changed significantly during such events [29,32,33,37]. The resilience of the affected network is therefore closely related to the process of reconstruction of the blocked links. This raises several important questions. Are all blocked links equally important during the reconstruction process? If not, what criteria determine their importance? Is it possible or desirable to reconstruct the most important links among the first ones? What sequence of repair works shall an administrator set, if only a limited number of repair units are available? How shall the administrator place the resources (as heavy machinery) in order to ensure optimal reconstruction? And what does the optimal reconstruction actually mean? The answers to these questions are not trivial and all of them lead to optimization problems. **The main task is thus to find such sequences of blocked links whose reconstruction is in some sense preferable over other existing sequences.**

There are two basic approaches to this problem. The first one focuses on ranking the elements of the network which should be repaired, but does not consider the routing of the repair units [10,35,9]. The other one draws attention to the scheduling and routing of the repair units under various assumptions and constraints. **It can take into account the maximum time needed to reconnect the network and minimize it** [1,36]. The optimization problem can be, however, made more complicated. In [13], **an asymmetric traffic assignment model** was incorporated into their algorithm. [25], analyzed an early stage of the repair activities with the aim of maximizing the performance of the emergency road repair activities, maximizing the number of people that benefit from it and minimizing the risk for repair units. **The problem can also be formulated as an integer network flow problem** [66,68]. Both approaches can be further combined with distribution relief

[45,51,65,67], limited resources [39,43,44], time constraints and other related operating constraints [69]. These extensions of the model lead to various types of loss functions which can also involve the total weighted earliness of all the cleared paths [3], accessibility [47], travel cost [42] and the total prize gained by reconnecting the network [2,36]. A number of models attempt to cope with incomplete information concerning the debris along the roads and try to find an optimal sequence of links for each period of reconstruction [11] and with stochastic factors during the operational stage [70]. Additional relevant papers dealing with the reconstruction process include [26,27,5,52,62].

The primary problem of all the approaches is that the optimal solution has to be found in a large state space. Let us assume, for instance, 40 concurrently blocked links. There are 8×10^{47} possible sequences of blocked links which should be evaluated. It is apparent that it is impossible to evaluate all the sequences in a reasonable time. **The number above can be significantly reduced if we know the position (the base) of the repair units**. Despite this simplification, the problem unfortunately still remains highly nontrivial and ranks among the so-called NP hard problems. This means that the time, we need to find its solution, depends strongly on the size and structure of the network and cannot be found in a reasonable time when a large network is investigated. Despite this fact, the deterministic and stochastic algorithms are developed to cope with larger networks. **The problems of network reconstruction were modeled using mixed integer programming and solved with a fuzzy genetic algorithm** [13], GRASP and VNS metaheuristics [43], dynamic programming and an iterated greedy-randomized constructive procedure [45], a rule-based constructive heuristic [47], a Markov decision process reconstruction [11], a greedy algorithm using critical links [42], a heuristic algorithm based on problem decomposition and variable fixing techniques [69,70], an ant colony optimization algorithm [65,68], simulated annealing [27], tabu search [27] and genetic algorithms [5,62]. The results produced by the above algorithms are often incomparable due to the use of different loss functions. However, in [65], one finds a comparison among an ant-colony based heuristic algorithm with an original algorithm Cplex 12.5 of a deterministic nature on a small network. The heuristic algorithm produced results which seem to be close enough to the best solution in a much shorter time. A similar approach for one repair crew can be found in [45], where two algorithms were developed and analyzed. The first one is based on dynamic programing and is able to find exact solutions on small networks (up to 41 nodes). The other iterated greedy-randomized constructive procedure is then tested on the small networks as well and further on medium and large networks (up to 401 nodes). In [27], the authors compare two stochastic algorithms (simulated annealing and hill-climbing procedure) and a tabu search algorithm. In [43], two metaheuristics (GRASP and VNS) are analyzed using small networks with known optimal solutions and then applied to large networks (216 nodes) without known optimal solutions. In [47], the authors presented four rule based heuristics and an analysis of the averages of their results and their variability. Optimality of the solutions was not, however, discussed. The study of the resilience can be more complex if we include other phases in the whole process including pre-event and post-event resilience. The three phase process stochastic model, based on evaluating possible scenarios combined with the user equilibria-traffic assignment problem, was introduced in [24].

The size of this problem can be reduced by a ranking of the blocked links under various criteria (see [53,6,61]). This can be seen, for instance, in [53,61] where the Network robustness index and Network trip robustness were used to evaluate the importance of links after their interruption or decreased capacity. The disadvantage of the method is that the ranking can change under various events when more links are blocked. A link may, for example, exist which is not very important for the functioning of the network. If several other links are blocked, however, and the link is the last one preventing the network from disintegration, the importance of the link dramatically increases. The next problem is that the repair of the links, according to their ranking,

can lead to a zig-zag movement of the repair units in the network which can be costlier and have undesirable side effects in the relief distribution or other phases of the reconstruction process. Another interesting approach can be found in [56]), where the authors focused their attention on bridges as key parts of the infrastructure. They optimized pre-positioning of recovery centers based upon clustering of the bridges, operational cost and system reliability. The formulation of the problem leads to integer programming with two contradictory objective functions. The problem was resolved by the Pareto front and additional evaluation of the results.

This paper has the following structure. We provide a formulation of the problem in Section 2 and introduce the algorithm for construction of its solution. We also introduce possible modifications and extensions of the algorithm which cover the different optimality criteria, optimality constraints and multi-objectivity. We show the ability of the algorithm to find the optimal solution for a problem on the Sioux-Falls network in Section 3. We then introduce several examples together with their results which demonstrate the flexibility of the proposed method with larger networks. We also tune the parameters used in the algorithm. Section 4 focuses on the results and Section 5 contains the conclusion.

2. Model and methods

The problem of a network reconnection resembles problems such as travelling salesman problems (TSP) and vehicle routing problems (VRP) (see for instance [16]. Compared to the TSP problem, we assume more than one repair unit and we do not require visiting each component exactly once. The VRP problem with its fleet of vehicles is much closer to the present problem. It does not contain, however, sets of equivalent nodes (customers) when it is sufficient to visit only one of them (see the sets of boundary nodes below).

We impose four basic requirements on our model and algorithm:

1. working with large networks which represent geographical regions
2. ensuring smooth access to all parts of the network, i.e., reconnecting the network
3. ensuring in some sense the optimal use of the repair units
4. replacing some constraints on solutions by recommendations

The first requirement seems natural because large number of events does not usually concentrate in a small area. Larger networks underline, however, the NP-hard nature of the problems and restrict the applicability of many deterministic algorithms. In this paper we thus provide an example of the network which is much larger than the networks in the above-mentioned papers. The second requirement is also logical and we show how to significantly reduce the number of nodes which have to be visited. To meet the third requirement, there is a need to define the optimality criteria which are often represented by a loss function. Two main disadvantages exist for any loss function. The first one is that it is unable to cover the optimal behavior of all repair units together. The second one is that it can contain several different quantities. It thus blurs the results because they are often optimal only in a sense of the weighted combination of the quantities. The problem is related to the fourth requirement. The constraints are, namely, expressed as inequalities or incorporated into a loss function. They can often be based upon imprecise estimates and may prevent the algorithm from finding a better solution by their small violation. It would seem to be interesting to replace the constraints with the recommendations, which can partially influence the process of the construction of solutions, but which are not incorporated into the loss function. It enables us to keep the optimality criteria clear and simple, while at the same time adding greater variability to the constructed solutions. The approach can produce an interesting alternative to a commonly used multi-criteria optimization (see Section 2.4 for a more extensive discussion). We refer the reader to [49], for another approach based upon

the replacement of the multi-criteria optimization by a set of objectives which should be met.

The above discussed requirements lead us to the metaheuristic known as the Ant Colony Optimization (ACO) (see [18], and Section 2.2 in this paper) which seems to be well-suited for the optimization process. The application of the ACO algorithm for emergency roadway repair is not new and was first introduced in [68]. In their paper the algorithm searched for an optimal solution to the time-space network flow problem which corresponds to the minimum of the maximum of times needed for individual repair units to reopen blocked links in a limited time. Individual ant searches for a route of a randomly chosen repair team until it reaches a given time limit. It then switches to another randomly chosen team. The links which are not repaired are finally assigned to the nearest repair units. In this paper we demonstrate a different model and application of the ACO algorithm and indicate how to add additional criteria (recommendations) to the optimization process. Unlike Yan and Shih [68], the algorithm we introduce is closer to the algorithms solving VRP and our optimal solution takes into account not only the maximal time a repair unit spends on the reconstruction process but also the times of other repair units. We also assign an ant to a repair unit and allow the ants to construct routes concurrently and not successively as in Yan and Shih [68]. To ensure a proper search of the neighborhood of a found solution, we use several colonies of ants. It enables us to apply parallelization of the problem for large networks, if necessary. We further require the repair units to return to their starting location, although the assumption can be easily relaxed. The similarity with the VRP enables the use of a 3-opt as a local search method which further improves the quality of the found solutions.

To cope with the above requirements the process of the construction of a respective model and algorithm can be decomposed into the following parts:

1. Define new optimality criteria using a special ranking.
2. Identify the nodes lying on the boundaries of the components.
3. Use the nodes and the shortest routes among them to create a complete graph and thus reduce the computational complexity.
4. Modify the ACO to cope with recommendations.
5. Use a suitable local search method.

The first point enables us to ensure the optimal result for all repair units. In the second point we use the fact that there is a need to reconnect the components for the reconnection of the network. It is thus sufficient to only visit their boundary nodes. This idea allows us to reduce the complexity of the problem by the construction of a complete graph from them. The construction of the complete graph also enables us to omit repeated computation of routes among nodes. In point 4 we exploit the potential hidden in pheromones and heuristics of the ACO algorithm used for the construction of the optimal solution. The fifth point is a standard part of many stochastic algorithms.

We apply the algorithm under three scenarios:

1. The repair units must visit all the components at least one time. The repair units can overcome the blocked links without loss of time.
2. The repair units must visit all components at least one time. The repair units can overcome the blocked links without loss of time, although the repair unit which as the first one meets the blocked link must repair it. The time needed to repair the blocked links is generally nonzero.
3. The repair units must visit all components at least one time. The repair units can overcome the blocked links without loss of time. The optimization process is influenced by other criteria which represent the importance of the components.

The requirement of zero time needed to overcome the blocked links is not illogical in our part of Europe since natural disasters often only

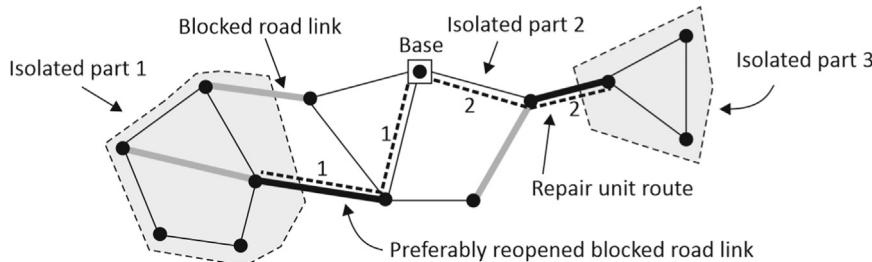


Fig. 2. An example of a disintegrated road network into three components with blocked links. The base contains two repair units which only have to reopen the two originally blocked links to ensure network connectivity.

partially damage the roads. They are thus not suitable for common traffic but can be used and without loss of time bypassed by repair units. The first scenario can also be used in the case of optimization of routes for heavy machinery when the time needed for the reconstruction is not important because roads are partially passable. In the second scenario we want to optimize the routing for repair units which includes the time to repair the blocked links. The time needed to repair a blocked link is usually much longer than the time to get to the link. We believe that the time the repair units spend on unblocked links cannot be neglected because it covers secondary costs, such as for petrol, etc. Under the third scenario we study the case where the recommendations during the construction of routes for repair units need not necessarily correspond with the optimality criteria. We also demonstrate how this algorithm can be modified and extended in Section 2.4.

2.1. Formulation of the problem and adjustment of the respective graph

In this section we introduce basic notation and show how the road network is represented by a graph and then reduced to a complete graph. We also provide a mathematical formulation of the problem we solve. We introduce here minimization of time of repair units which ranks among the important criteria [1,36]. It also enables the reader to visually verify the results.

Let us assume that m repair units are available. Any road network can be expressed as an undirected connected graph $G = (N, L)$, where N is a set of nodes and L is a set of links among them. A link between nodes i and j is denoted as (i, j) , c_{ij} is the time needed to travel the link. Let us further assume that the network is damaged after an event and disintegrated. This means that B , a set of links, which are blocked, exists. Let us denote as r_{ij} the time needed to repair the link $(i, j) \in B$. If the link (i, j) does not belong to B , then $r_{ij} = 0$. Set $C = \{C_1, \dots, C_n\}$, where C_i is the i^{th} component and consists of the nodes from N for which there is no route to the nodes in the other components and n is the number of components. Without a loss of generality we assume that the blocked links from B lie among the components from C and not inside any component C_i . Due to technical reasons, we assume that the starting node of the repair units forms one of the components and r_{ij} for the artificially blocked links leaving the starting node are equal to zero.

There is consequently a need to reconnect the components from C in order to reconnect the disintegrated network once again. To reconnect the network, it is only sufficient to visit the boundary nodes of the particular components. Let us denote as BC_i the set of boundary nodes of C_i . We argue that the node k is the boundary node of the i^{th} component if $k \in C_i$ and if there exists $l \in C_j$ such that $i \neq j$ and $(k, l) \in L$. To reduce the size of the problem and consequently the computational time, we can replace the graph G with a complete graph

$$CG = (\cup_{i=1}^n BC_i, SP) \quad (1)$$

where $\cup_{i=1}^n BC_i$ is the union of all the sets with the boundary nodes and SP is a set of links among all the nodes from $\cup_{i=1}^n BC_i$ whose lengths correspond to the lengths of the shortest paths in (undamaged) G among the nodes. To preserve the needed information we add two

attributes to the links from SP . We assume to have a link $(k, l) \in SP$. The first attribute is the sequence $sp_{kl} = [(k, i_1), (i_1, i_2), \dots, (i_s, l)]$ of links from L which form the shortest path between nodes k and l while the other one is the time needed to traverse the shortest path. In this way, the graph size is much smaller than the original network and thus computations are easier and quicker.

$$t_{kl} = c_{ki_1} + \sum_{j=2}^s c_{i_{j-1}i_j} + c_{is_l} \quad (2)$$

We now derive a mathematical model corresponding to our problem. To express the problem verbally we can state:

Let us have a network with a large number of blocked links. Let us further assume that the blocked links can be overcome by repair units but are not suitable for common transportation. What is the optimal sequence of remedial works to ensure the reconnection of the particular components and the return to the base in a minimal time? (see Fig. 2 for a simple example).

Let us begin with the quantity which should be minimized. All solutions can be interpreted as a descending sequence of times

$$T = [T_1, \dots, T_m], \quad T_i \geq T_j \text{ for } i < j, \quad (3)$$

where m is the number of repair units. Times T_i represent the time the repair units have to spend on their routes and on repairs before they return to the base. The fact that T is a descending sequence enables us to define the ranking (lexicographical order)

$$A < B \Leftrightarrow \exists i : A_j = B_j \text{ for } j < i \text{ and } A_i < B_i. \quad (4)$$

We argue that m -tuple T is acceptable if there are sequences of links from SP for all the repair units whose repairs lead to the recovery of the network. Let us denote the set of the sequences with AD . We then look for using a ranking from (4).

$$\operatorname{argmin}_{AD} T \quad (5)$$

Having formulated the optimization problem, we take a look at the computation of times $T_i, i = 1, \dots, m$, under the three scenarios described above. The scenarios can be covered by the following definition:

$$T_i = TS_i + TR_i \quad (6)$$

where TS_i is the shortest time needed to visit the respective components and return to the base in an undamaged graph, while TR_i is the total time the repair unit must spend by repairing the blocked links lying on the shortest path with the length TS_i . To make the above description more detailed, we describe each time quantity with respective mathematical relations. We assume to have the i^{th} repair unit which reconnects k components and returns to the base. Then

$$TS_i = \sum_{j=0}^k t_{s_js_{j+1}}^i \quad (7)$$

where $s_0 = s_{k+1}$ is the base and $t_{s_js_{j+1}}^i$ is the time defined by (2) for nodes $s_j \in C_{s_j}$ and $s_{j+1} \in C_{s_{j+1}}$. The upper index i corresponds to the shortest path of the i^{th} repair unit. The second quantity TR_i is defined as follows

$$TR_i = \sum_{(l,m) \in L_i} \eta_m \quad (8)$$

where L_i is a set of blocked links on the route from (7) where the repair unit comes first and thus all the links must be repaired. η_m is the time to repair the blocked link between nodes l and m .

To evaluate the sequence of values (for instance the number of people) assigned to components reconnected by each unit, we use the following sum:

$$P(V) = \sum_{i=1}^n (n - i + 1) \cdot v_i, \text{ for } V = [v_1, \dots, v_n], \quad (9)$$

where v_i is the value assigned to the i^{th} component. It is apparent that the higher numbers v_i are situated at the beginning of the sequence V , the higher the number P . We call the number P the progress.

We have introduced the graph representation of a road network and all the notations and functions we use afterwards. Our approach is based on minimizing the time the repair units spend on the road and repairing the blocked links. With relation (4) we have also defined a special ranking which enables a comparison of particular results according to the times.

2.2. Ant colony optimization

In this section we present the reader with a basic version of the ant colony optimization (ACO) we further modified. The basic idea of this algorithm was introduced by [18]. One of the primary issues for every stochastic algorithm is its convergence to the optimal solution. We refer the reader to [20,30,31,60], for the theory related to this problem. The reason why we choose this particular algorithm is that it was successfully utilized in solving related problems, namely TSP [18,19] and VRP [28,50]. The algorithm was further successfully applied in a number of other fields. We refer the reader to [20,21], for additional applications of the algorithm. The algorithm is based on the construction of a solution using recommendations via pheromones and heuristics. The basic version of the ACO algorithm is described below and is a slightly modified version of the algorithm from [21].

Algorithm 2.1. ACO algorithm

```

input: The road network represented by graph ( $G$ ), the number of colonies ( $numb\_col$ )
best_solution ← FindInitialSolution( $G$ )
 $P \leftarrow InitializePheromoneValues(G, best\_solution)$ 
 $H \leftarrow InitializeHeuristicInformation(G)$ 
while termination conditions not met
  for  $i = 1: numb\_col$ 
    new_solution ← ConstructSolution( $P, H, G$ )
    if new_solution is a valid solution
      new_solution ← LocalSearch(new_solution,  $G$ ) #optional
      if new_solution is better than best_solution
        best_solution ← new_solution
        PheromoneUpdate( $P, best\_solution, new\_solution$ )
output: best found solution ( $best\_solution$ )

```

The main advantage of the ACO algorithm is that the optimization process, despite its random nature, is under control due to pheromones and heuristics. The pheromones represent the cumulated experience of the ant colony, i.e., serve as a memory for the best so-far-found solution and ensure that the neighborhood of the solution will be thoroughly searched. The pheromones are updated after every iteration when the algorithm constructs new solutions. In contrast, heuristics represent the attractiveness of the next step during the process of construction of a solution independent from the found solutions. In the heuristics we can thus incorporate the additional information we want the algorithm to take into account. This information can accelerate the search for the solution or influence its direction using additional criteria. It also

provides the algorithm with larger variability and can be used for the transformation of constraints into recommendations.

An important part for any stochastic algorithm is how to prevent it from being stuck in local minima. Three measures were used here to resolve this issue. The first one is *pheromone evaporation* which results in the best found solution disappearing gradually. The second one is based upon *pheromone update* which can be done with a nonzero probability for a newly found solution, instead of for the best so-far-found solution. The last one is the process of construction of a new route based upon the probability:

$$prob_{ij} = \frac{p_j^\alpha h_{ij}^\beta}{\sum_j p_j^\alpha h_{ij}^\beta} \quad (10)$$

where i is a fixed node and j represents consecutive nodes which are connected with i , p_j and h_{ij} are the values of pheromone and heuristics, respectively, which correspond to the link (i, j) . α and β represent weight powers which enable us to highlight or suppress the influence of respective pheromones or heuristics. To prevent the algorithm from being stuck in a local minimum we also use a special version of ACO known as the Max-Min Ant System (MMAS) (see [57–59]). The algorithm in this version does not permit the pheromone value to leave a given interval during the evaporation and updates and further enhances the chances that it will not be stuck in a local minimum.

We have thus introduced a basic version of the ACO algorithm using a pseudocode to provide the reader with an impression of how it works. Ants construct a solution, moving along the edges, exploiting information provided by the heuristics and the pheromone left by other ants, and deposit a certain amount of pheromone based on the quality of the solution found.

2.3. The optimization process

We describe in more detail the entire optimization process to ensure the reproducibility of our results. The version of the ACO algorithm described in Section 2.2 forms the core of the proposed algorithm. This process can be broken down into the following parts:

1. Road network representation and its adjustment
2. Initialization
3. Construction of a solution
4. The update phase

2.3.1. Road network representation and its adjustment

The most common representation of a road network is a graph with nodes and links which can provide further information. The crucial step, however, is the transformation of the original graph G into the complete graph CG using the boundary nodes of the components which appear after disintegration. The transformation enables us to significantly reduce the number of nodes and links and thus the computational time. The next advantage of the transformation is that the movement in the complete graph facilitates the direct movement among the components. Using the notation from Section 2.1, we can break down the entire process into these steps:

1. Transform the undamaged road network into graph G
2. Interrupt (remove) the links which were blocked after a particular event
3. Find all the components of the subgraph of G
4. Create an artificial component from the node, where the repair units are located, by interrupting the links which leave it.
5. Find the boundary nodes for any component of the subgraph.
6. Use the undamaged graph G to compute the shortest paths among all pairs of boundary nodes.
7. Create a complete graph CG using the nodes from step 5 and the links representing the shortest paths from step 6.

Table 1

Attributes of links in the complete graph.

Attributes of links of CG	Description of the attributes
The shortest path among the boundary nodes in graph G	Sequence of links from L .
Cost of the link	Time t_{kl} needed to overcome the corresponding shortest path.
Time of the repair	The total time needed to repair interrupted links on the shortest path (sum of corresponding r_{lm}).
Pheromones	Attractiveness p_{kl} of the links affected by their presence in the best found solution.
Heuristics	Attractiveness h_{kl} of the links under additional assumptions.

The individual steps can be performed using elementary graph algorithms (see for instance, [23]). Each link in the complete graph CG represents the shortest path in the original graph G and apart from its length it can carry other information about the route, the pheromones and heuristics as its attributes (see Table 1).

2.3.2. Initialization

In this subsection we demonstrate how to construct the initial routes for the repair units and establish the values of pheromones. These routes are then further optimized.

The initial routes should be constructed using a rapid procedure which provides a reasonable first solution T but does not compete with the quality of the solution obtained by the following optimization. We use two initializing procedures depending on whether we want to use special recommendations or not.

- Without recommendations, we use the so called savings algorithm (see [14,48]). We remind the reader that we work with the complete graph constructed in the previous section. The savings algorithm works as follows

- Take the costs of the links between the node with the repair units (0) and other nodes and multiply them by factor 2, because the routes must cover the return of the units.
- Take the shortest path $(0 - i - 0)$. Find such node j that the difference of the cost of the path $(0 - i - 0 - j - 0)$ and of the cost of the path $(0 - i - j - 0)$ is maximal.
- Replace the routes $(0 - i - 0)$ and $(0 - j - 0)$ with $(0 - i - j - 0)$ and regard the path $(i - j)$ as one node k .
- Repeat the entire procedure until the number of routes is reduced to the given number of repair units (m).

- With recommendations. In this case we assume that each component is in some sense evaluated by a number. The number can represent the number of people living in the component, the importance of the component, etc. The process of the construction of an initial solution is:

- Define m ants with the repair units based at a starting node. Take an ant.
- Order the nodes according to their value given by the number assigned to the component and, in the case of the same values, use the distance from the latest node on the route of the ant as a secondary criterion. Add the node with the largest value (and the shortest distance) to the route of the ant.
- Repeat step 2.2 for m ants.
- Repeat steps 2.2 and 2.3 until the ants visit at least one node in all the components.
- As the last node add the base of the repair units.

We therefore do not consider the time to repair in any of the constructions of the initial solution. We only add the time if necessary.

Now we demonstrate how pheromones are initialized. Using (3) and (6) we put

$$\text{lengthpath} = \sum_{i=1}^m T_i \quad (11)$$

Since our algorithm is based upon the MMAS version of ACO, we

must further prescribe the values

p_{\min} minimal possible value of the pheromones

p_{\max} maximal possible value of the pheromones

In the end we identify the links lying on the constructed routes and put the values of their pheromones

$$p_{ij} = \max\left(p_{\min}, \min\left(p_{\max}, \frac{1}{\text{length_path}}\right)\right) \quad (12)$$

The reciprocal value of the total length of the routes ensures that the shorter routes are more attractive for further optimization than the longer ones.

In the initialization of heuristics, we also use two procedures which distinguish between optimization without and with recommendations.

- Without recommendations we simply put

$$h_{kl} = \frac{1}{t_{kl}}$$

where t_{kl} is defined by (2). This means that during the construction of a route, the nearer components are more attractive for an ant than the more distant ones.

- With recommendations we take a link (k, l) between the components and the values v_k, v_l which correspond to the values assigned to the particular components. Without loss of generality we assume that the values are positive. Then we put

$$h_{kl} = \sqrt{v_k v_l}$$

Using the geometric mean, we ensure that the ants prefer components with higher values.

2.3.3. Construction of a solution

In this section we demonstrate the construction of a solution which consists of routes for repair units. In the construction we use the terminology from the ACO. This means we speak about ants and colonies in the optimization process. It enables us to keep the process of optimization separate from the results which represent the routes for the repair units. The number of ants corresponds to the number of repair units and the ants are used in the construction of routes for the repair units. Each colony has the same number of ants. The colonies enable us to construct routes (solutions) under the same setting (the same value of pheromones p_{ij} and heuristics h_{ij}) which are further compared, and the best of them can be used in a pheromone update (see Section 2.3.4). The approach is particularly suitable for the parallelization and enables us to search the neighborhoods of the routes encoded into pheromones more thoroughly which can lead to a better solution. In the following Algorithm 2.2, we describe the construction of the routes.

Algorithm 2.2. Construction of routes

input: nodes in graph (nodes_in_graph), starting node with repair units (start_node), colony of ants (colony), matrix with probabilities (prob_{ij}) based upon pheromones and heuristics given by (10)
forbidden_nodes = { start_node }

```

shift all ants in colony to start_node
while nodes_in_graph – forbidden_nodes is not empty
    a ← 0
    for ant in colony
        new_node ← GenerateNewNode(current_ant's_position, (probij))
        if new_node doesn't exist
            the ant returns to start_node
    else
        length_of_path ← length of path between current ant's position and
        new_node
        char[a] ← 1/(length_of_path + time_to_repair +
        length_of_ant's_path)
        a ← a + 1
    chosen_new_node ← GenerateNewNode(char)
    add chosen_new_node and other boundary nodes in its component to
    forbidden_nodes
    shift the corresponding ant to chosen_new_node and recompute its path
output: ants with their routes

```

As can be seen in [Algorithm 2.2](#), we begin the construction of a solution with all the ants located in the starting node. We then construct the routes using individual ants until at least one node in any component is visited by an ant. This means that all the components were reconnected to the starting node. The individual routes are generated by ants node by node. The process proceeds as follows: let the ant a be in the node i and the cost of the route, it has constructed so far, is s_{ai} . Using [\(10\)](#) we randomly choose the node j as the next node on its route and put

$$\text{char}[a] = \frac{1}{(s_{ai} + t_{ij} + rt_{ij})}. \quad (13)$$

If it is impossible to generate a next node for the ant we put

$$\text{char}[a] = 0. \quad (14)$$

After normalization of char we can understand its values as probabilities. Using the probabilities, we select the ant b which can add the chosen node to its route. At the end we once again compute the value s_{bj} . After adding the selected node to the corresponding route, we know that its component is reconnected. We thus add all the boundary nodes of the component to the set $\text{forbidden}_\text{nodes}$ which consists of all the reconnected boundary nodes. In cases when there is not any node where the ant can move, the ant then returns to the starting node. We ensure that every ant colony generates one solution using [Algorithm 2.2](#). To find a better route in the neighborhood of the found solution, we implement a local search method based on 3-opt (see [\[40,41\]](#)). We can consequently generate several solutions whose number corresponds to the number of ant colonies. The best solution is selected in view of [\(3\)](#) and [\(4\)](#).

2.3.4. The update phase and termination of the computation

The primary aim of the update phase is to mark the best route, while at the same time prevent the algorithm from being stuck in a local minimum. This is the reason why this phase consists of two steps:

1. Pheromone reduction
2. Marking a chosen path

The pheromone reduction is realized by the relation

$$p_{ij} \leftarrow \max(r^* p_{ij}, p_{min}), r \in (0, 1). \quad (15)$$

During the process of marking, it is often unsuitable to mark the best so-far-found routes (see [\[20\]](#) because it can limit the probability that the algorithm finds an even better solution. We thus take the best so-far-found routes and the best routes found in the current iteration and randomly choose one of them. To update pheromones we then use relation [\(12\)](#). This ensures that the links which have not thus far

participated in the process of the construction of a solution still have reasonable probabilities of being chosen in the next construction. We therefore prevent the algorithm from being stuck in a local minimum. We refer the reader to [\[20\]](#) for a detailed description of the MMAS method and the further discussion there.

Although the termination of the computation can be carried out in several ways, we decided to terminate the algorithm after a given number of iterations.

To summarize this section, we have shown in much more detail how the original road network is reduced to a complete graph, how the initial solutions can be constructed and how the solution is further improved. The main idea of the algorithm is to go through the neighborhood of the initial solution and search for improvement. It is important that the described technique prevents the algorithm from being stuck in local minima and the new solutions can significantly diverge from the initial ones with nonzero probability.

2.4. Possible extensions and modifications of the method

There are a number of possibilities as to how the suggested algorithm can be applied in the field of reduction of impacts of disasters. We draw attention here to the extensions and modifications of the algorithm which cover various optimization criteria, optimization constraints and multi-objective criteria.

One of the natural modifications of the proposed algorithm is to repair in the first stage such links which are the most important for the network performance. The importance can be given for instance by their capacity or another form representing their criticality. Let us assume for simplicity that the higher numbers assigned to the links represent more critical links. Problem [\(5\)](#) can be then reformulated as

$$\underset{AD}{\operatorname{argmax}} T, T = [T_1, \dots, T_m], \quad T_i \leq T_j \text{ for } i < j \quad (16)$$

where the coordinates of T represent link criticality. The ordering [\(4\)](#) can be reversed

$$A > B \Leftrightarrow \exists i : A_j = B_j \text{ for } j < i \text{ and } A_i > B_i. \quad (17)$$

Using the relation

$$\underset{AD}{\operatorname{argmax}} T = \underset{AD}{\operatorname{argmin}} (-T + C)$$

where C is a universal constant added to all coordinates of T to ensure its positivity, we can transform problem [\(16\)](#) back to problem [\(3\) – \(5\)](#) and solve it analogously. The lengths of the shortest routes used in the construction of the complete graph can be replaced by the maximal criticality of the blocked links lying on them.

The usual optimization constraint is a budget constraint which requires repairs to the network under limited resources. Let us assume that the time given by [\(3\) – \(5\)](#) is an optimization criterion. In the first stage we can modify the savings algorithm to construct an initial feasible solution repairing the links with the lowest costs. Due to the constructive nature of our algorithm, we can simply incorporate budget constraints. We allow the ants to construct the routes adding single links and, in case the budget is exceeded, we can stop and restart the construction. In this way we can also add any number of additional constraints without restricting the performance of the algorithm.

The last topic discussed is the multi-objective optimization. The typical optimization involves criteria which contradict one other. To resolve the problem the Pareto front was developed in [\[15\]](#). The Pareto front represents a set of solutions for many convex combinations of two criteria. The optimal solution is then selected from the set using additional criteria. Let us assume we now have a case where the initial solution is constructed with regard to the people living in components (see [Section 2.3.2](#) or another algorithm can be used) but the minimization problem is given by [\(5\)](#). In view of [\(5\)](#), the algorithm begins to prefer time to people and starts to converge to the solution of [\(5\)](#). Due to the constructive nature of the algorithm, we can follow and evaluate

the constructed solutions between the two stages and thus simulate the Pareto front. Since we can only assure the local optimality of the solutions, we can understand the process as a weaker but faster version of the Pareto form.

3. Results

In this section we applied the algorithm from [Section 2](#) to two networks. We used a computer with Linux, 10 CPUs Intel E5-2670 2.6 GHz, RAM: 10 GB. Parallel computing was only used if we needed to repeat the entire computation several times. The algorithm was programmed in Python 3.6.

The main problem of all stochastic algorithms is how “optimal” a found solution is. There are several standard processes which can be used to provide basic information about the results. Due to the nature of our problem we decided to test our algorithm on a small network where the best solution is easily verifiable. For this purpose we chose a modified Sioux-Falls network [\[38,72\]](#). We further tested the algorithm on a larger network representing a region in the Czech Republic.

3.1. Sioux-Falls network

We used a modified version of the Sioux-Falls network (see [\[38,72\]](#)). The modified network consists of 24 nodes and 38 bidirectional links and each of the links has an attribute which corresponds to time in seconds needed to overcome the link. We simulate an artificial event which leads to disintegration of the network (see [Fig. 3](#)). For the sake of simplicity, we assume that the blocked links can be overcome without loss of time to distribute relief and that the repair units need not return to their base.

The basic setting is:

α, β
numb_of_colonies

1, 2
30

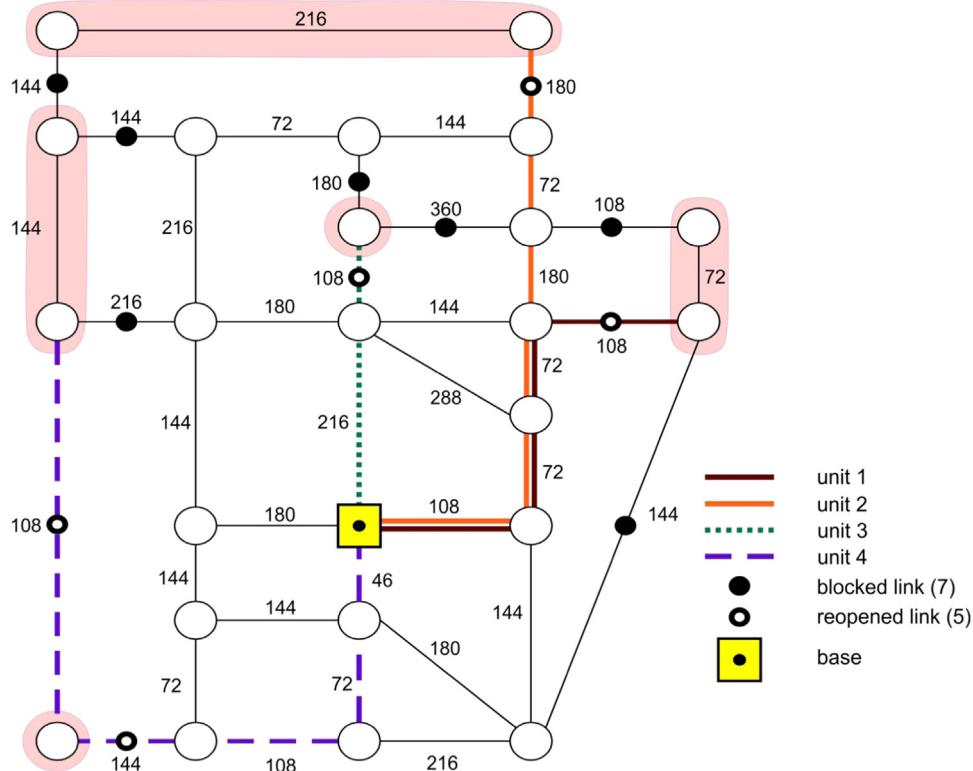


Fig. 3. The best routes for 4 repair units. All 5 components were reconnected using 5 reopened links.

Table 2
The best found times for the repair units.

Number of repair units	Best found results [s]
4	[684, 540, 360, 324]

numb_of_ants	4
numb_of_iterations	10
p_min, p_max	0, 1
r	0.8

The best result of the algorithm is summarized in [Table 2](#) and [Fig. 3](#).

Under the above setting, we carried out 10,000 computations to obtain information about the stability of the results. The time needed for one of the computation is 0.5 s. The results are summarized in [Table 3](#).

In addition, more than 94% of the 10,000 computations found the best results. The NaN in the last two columns indicates that the fourth vehicle remains at the base in the initial solution and thus the length of its route is equal to zero. It is apparent that in the case the respective values are infinite.

3.2. Zlín network

We used the complete road network of the Zlín region in the Czech Republic as a large network which has 723 nodes and 974 links. We artificially interrupted 46 links which leads to the disintegration of the network into 16 components. This setting simulates a disastrous event which can be caused, in this particular region, by a combination of flooding and landsliding (e.g., [\[7\]](#)). We subsequently introduced the following three scenarios which can be based on the decision of a road administrator (see also [Section 2](#)):

- Minimize the time for all repair units needed to visit any component at least one time under the assumption that the repair units alone

Table 3
Stability results.

Number of repair units	Averages of the best results [s]	Relative standard deviations of the best results [%]	Average percentage change to the initial state [%]	Standard deviation of the percentage change [%]
4	[684, 542.3, 362.7, 324]	[0, 2.9, 3.7, 0]	[-1.4, -7.1, -8.9, NaN]	[3.6, 9.6, 13.5, NaN]

- are able to overcome the blocked links without loss of time.
2. Minimize the time for all repair units needed to repair and reconnect the network under the assumption that the repair units alone are able to overcome the blocked links without loss of time.
 3. Minimize the time for all repair units needed to visit any component at least one time under the assumption that the components can be somehow evaluated and the components with higher values are preferred. We once again assume that the blocked links can be overcome without loss of time.

In the following subsections we discuss particular scenarios more thoroughly.

In contrast to Section 3.1, we do not know the optimal solution for the larger network. We thus display the best found solution to enable the reader to visually verify the meaningfulness of the results. We also demonstrate how some of the parameters were tuned, namely we show

1. Dependence of the results on coefficients α, β from (10).
2. Dependence of the results on the number of iterations.
3. Mean values and standard deviations of the results after repeated computation.

3.2.1. Scenario 1

In this subsection we analyze the situation when the repair units are able to overcome the interrupted links without any delays but at the same time the links are not suitable for a common traffic. The scenario can be used for relief distribution if the damages of the links are not serious or for optimization of routes for heavy machinery which repair the links without taking into account the time of repair. The optimization process based on (3) – (5) ensures minimization of the time the repair units have to spend on the distribution of aid. We make use of the savings algorithm and the first definition of heuristics (see Section 2.3.2). In order to provide a justification for the choice of parameters, we also present their impacts on the results.

We begin with the dependence of the results on coefficients α, β from (10). Due to computational difficulty, we study the combinations of (α, β) under the following settings:

(α, β)	$[1,2,3,4,5] \times [1,2,3,4,5]$
<i>numb_of_colonies</i>	30
<i>numb_of_ants</i>	4
<i>numb_of_iterations</i>	1000
<i>numb_of_repetitions</i>	100
<i>p_min, p_max</i>	0, 1
<i>r</i>	0.8

The study of 25 combinations of numbers 1, 2, 3, 4, 5 seems to be reasonable in comparison with similar problems [18]. Other variables (*numb_of_colonies*, *p_min*, *p_max*, *r*) were again chosen according to the recommended values in comparable problems [18]. The variable *numb_of_ants* is given by the number of repair units. The choice of the value of *numb_of_iterations* is based upon the observation that the results begin to stabilize at the value and are not further significantly improved (see the second test). For any of the combinations (α, β) we obtain 100 results (*numb_of_repetitions*) which enable us to compute the mean values for all the repair units and the mean values provide better insight into the role of particular values of α and β . The results are summarized in Table 4.

Table 4
Results for (α, β) .

α	β	Average time of repair unit no. 1 [s]	Average time of repair unit no. 2 [s]	Average time of repair unit no. 3 [s]	Average time of repair unit no. 4 [s]
1	1	9922.00	9118.20	8836.43	7956.66
1	2	9922.00	9136.62	8838.10	7962.95
1	3	9922.00	9155.44	8842.37	7893.32
1	4	9922.00	9176.98	8827.27	7817.31
1	5	9922.00	9192.60	8825.90	7721.46
2	1	9922.00	9180.40	8823.43	7776.47
2	2	9922.00	9177.11	8833.20	7826.00
2	3	9922.00	9187.20	8821.53	7815.84
2	4	9922.00	9208.77	8799.99	7761.55
2	5	9922.00	9227.98	8780.82	7704.44
3	1	9922.00	9218.00	8785.83	7733.02
3	2	9922.00	9215.17	8793.48	7763.80
3	3	9922.00	9226.31	8769.49	7743.52
3	4	9922.00	9245.66	8748.31	7711.01
3	5	9922.00	9268.93	8708.16	7658.39
4	1	9922.00	9259.43	8715.94	7672.93
4	2	9922.00	9254.26	8723.93	7690.68
4	3	9922.00	9259.01	8718.05	7694.42
4	4	9922.00	9274.90	8693.03	7667.17
4	5	9922.02	9293.37	8651.26	7626.64
5	1	9922.02	9284.95	8659.41	7641.20
5	2	9922.02	9279.60	8673.78	7668.58
5	3	9922.01	9280.19	8680.22	7676.34
5	4	9922.01	9288.53	8665.87	7655.75
5	5	9922.01	9301.40	8640.14	7634.39

In view of (3) – (4), the combination $(\alpha, \beta) = (1, 1)$ seems to be the most suitable for further testing.

In the second part of the analysis we try to find the optimal number of iterations to reduce the computational intensity. The settings are

α, β	1, 1
<i>numb_of_colonies</i>	30
<i>numb_of_ants</i>	4
<i>numb_of_iterations</i>	[200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]
<i>numb_of_repetitions</i>	100
<i>p_min, p_max</i>	0, 1
<i>r</i>	0.8

The range of the variable *numb_of_iterations* is based upon the observation that 200 iterations should be enough to have an improvement of the initial state and more than 2000 iterations should not have a significant impact on the quality of the results. We once again pay attention to the mean values of the 100 results. The results are summarized in Table 5, along with the average time of one of 100 repetitions.

It is apparent that the choice of 1000 iterations in the previous analysis was meaningful.

We now analyze the results of the algorithm under the same setting as above. We only put $(\alpha, \beta) = (1, 1)$ and *numb_of_iterations* = 1000, which seem to be reasonable values due to the results in Tables 4, 5. The algorithm was repeated 100 times. In the following Table 6, we present the main characteristics of the computation: the best found solution, the average percentage improvements of the first solution derived by the savings algorithm and their relative standard deviation,

Table 5
Results for the number of iterations.

Number of iterations	Average time of one repetition [s]	Average time of repair unit no. 1 [s]	Average time of repair unit no. 2 [s]	Average time of repair unit no. 3 [s]	Average time of repair unit no. 4 [s]
200	294.29	9922.00	9215.11	8650.67	7735.23
400	589.60	9922.00	9187.82	8745.25	7880.24
600	879.73	9922.00	9171.94	8783.00	7913.13
800	1184.24	9922.00	9161.33	8795.51	7957.52
1000	1461.82	9922.00	9153.00	8797.83	7964.53
1200	1757.60	9922.00	9147.53	8801.79	7977.37
1400	2019.96	9922.00	9143.60	8802.29	7968.26
1600	2310.53	9922.00	9140.30	8800.26	7978.41
1800	2710.24	9922.00	9137.84	8800.25	7984.64
2000	2965.57	9922.00	9135.54	8797.00	7981.80

Table 6
Analysis of the results.

Best found solution [s]	9922, 9114, 8688, 7748
Average percentage improvement, relative standard deviation [% (%)]	(−5.55, 3.12), (−10.58, 2.87), (−9.7, 4.46), (−6.51, 13.91)
Average times of repair units and their relative standard deviation [(s, %)]	(9922.0, 0.0), (9125.75, 0.33), (8837.65, 1.55), (7937.1, 6.98)

the average times of the repair units and their relative standard deviations.

The best found solution is depicted in Fig. 4.

3.2.2. Scenario 2

In this subsection, we assume that the time needed for repairing the blocked links has already been estimated. Using the data we would like to minimize the time which the repair units have to spend on the

Table 7
Results for (α, β) .

α	β	Average time of repair unit no. 1 [h]	Average time of repair unit no. 2 [h]	Average time of repair unit no. 3 [h]	Average time of repair unit no. 4 [h]
1	1	30.50	28.69	26.07	21.26
1	2	31.16	29.16	26.48	21.98
1	3	32.00	30.00	26.97	22.10
2	1	32.98	30.98	28.22	23.58
2	2	32.68	30.72	28.04	23.61
2	3	32.77	30.84	28.22	23.79
3	1	33.10	31.16	28.54	24.25
3	2	32.96	31.01	28.48	24.32
3	3	32.90	30.97	28.47	24.40

reconnection of the entire network. The basic setting is the same as in the previous example, including the number of repetitions. We only include the time needed to repair the blocked links into the computation. We use the savings algorithm and the first definition of heuristics (see Section 2.3.2). We once again first test which coefficients α, β are the most suitable. The results can be found in Table 7. In a comparison with Table 4, we present only a shortened version of the table.

As in Scenario 1, it is apparent that the best results are produced by the coefficients $(\alpha, \beta) = (1, 1)$. The combination is placed into the next computation where we test the sensitivity on the number of iterations (see Table 8). We again present its shortened version.

It is apparent from the table that compared with Table 5 it is suitable to choose 2000 iterations for further computation because the improvement is significant.

We now provide the analysis for the algorithms under the above selected settings, i.e., $(\alpha, \beta) = (1, 1)$ and *numb of iterations* = 2000. We present the main characteristics of the computation in the following Table 9: the best found solution, the average percentage improvements

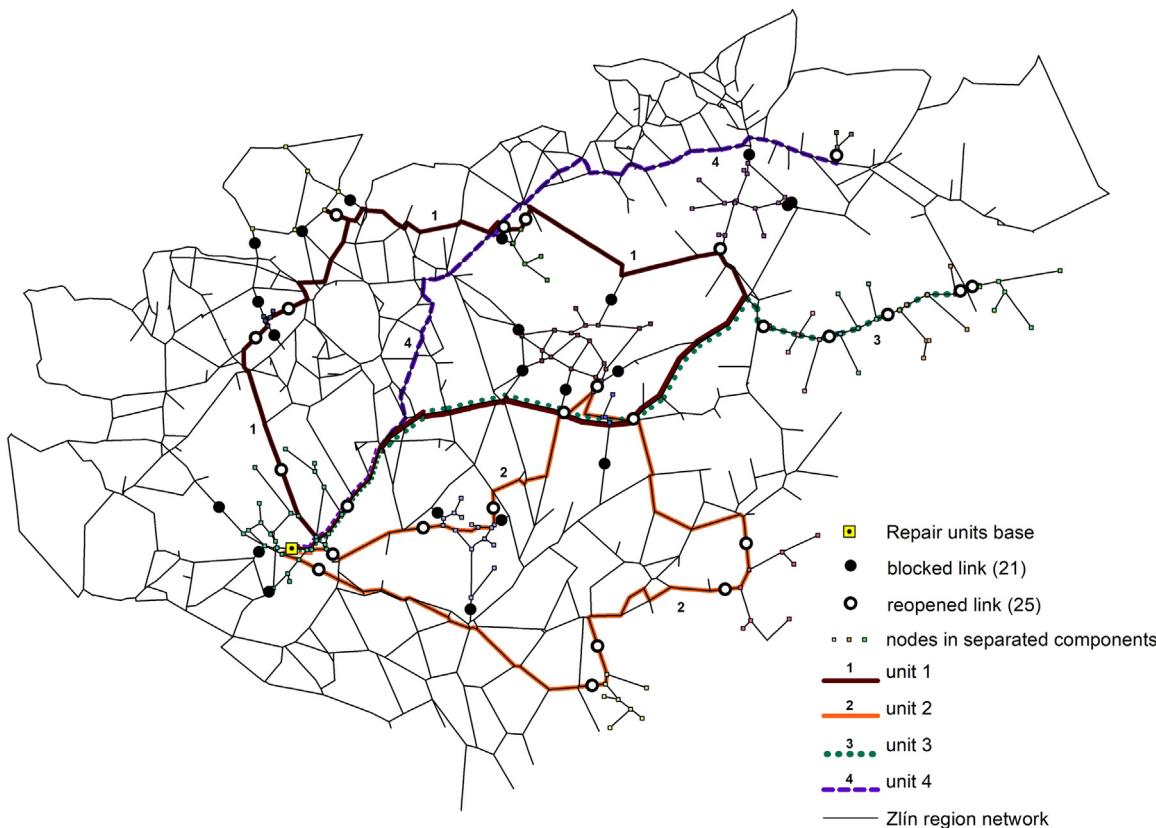


Fig. 4. The best routes for 4 repair units.

Table 8
Results for the number of iterations.

Number of iterations	Average time of a repetition [s]	Average time of repair unit no. 1 [h]	Average time of repair unit no. 2 [h]	Average time of repair unit no. 3 [h]	Average time of repair unit no. 4 [h]
1000	1796.90	31.65	29.68	26.92	22.67
1200	2160.70	31.40	29.45	26.73	22.64
1400	2522.87	31.23	29.32	26.60	22.47
1600	2891.22	31.05	29.13	26.46	22.34
1800	3311.83	30.91	28.96	26.30	22.20
2000	3631.80	30.77	28.86	26.19	22.06

Table 9
Analysis of the results.

Best found solution [h]	27.74, 24.10, 19.55, 16.60
Average percentage improvement, relative standard deviation [(% , %)]	(−37.98, 7.12), (−37.45, 11.33), (−33.10, 14.77), (−26.81, 20.42)
Average times of repair units and their relative standard deviation [(h, %)]	(29.5, 6.28), (27.58, 7.5), (25.3, 11.63), (20.81, 23.36)

of the first solution derived by the savings algorithm and their relative standard deviations and the average times of repair units and their relative standard deviations.

The best result is illustrated in Fig. 5.

3.2.3. Scenario 3

In this subsection, the results similar to Scenario 1 are shown, but include the number of inhabitants living in the components into the computation as a measure of their importance. We require that the repair units spend no time with the repair of the interrupted links and want to spend the minimal time visiting each component at least one time. They are also distracted from their aim by a preference for

reconnecting the components with the higher number of inhabitants living within them. We thus use the optimization (3) – (5) and the second algorithms and relations for the creation of an initial solution. We also use the relation (9) to evaluate the sequences with the numbers of reconnected people. It is apparent that the higher number of people reconnected earlier leads to the higher number P .

The setting remains the same as in Scenario 1 and 2. The results of the sensitivity of α, β can be seen in Table 10 which was shortened.

There is now a need to choose coefficients α, β for further testing. Taking into account the above-mentioned preferences, we decided to test further computation with $(\alpha, \beta) = (1, 2)$. The combination provides not only sensible average times for the repair units, but also provides higher values of the progress for particular repair units. The results for the number of iterations are summarized in Table 11.

It follows from the table that further choice of 2000 iterations seems to be reasonable.

The following table covers the best found solution and the average results after 100 repetitions of the algorithm under the above-mentioned settings, namely $(\alpha, \beta) = (1, 2)$ and $\text{numb_of_iterations} = 2000$ (Table 12).

The best solution is in Fig. 6.

4. Discussion

Optimization of the reconstruction process of a road network damaged by an event is not a new problem. It has been solved under various constraints, loss functions and using various algorithms. The loss functions and constraints were usually introduced as classical functions and inequalities, respectively. Their aims were to minimize the maximal time needed to reconnect the network [1] and the travel cost [42], to maximize the performance of the emergency road repair activities [25] and the total weighted earliness of all the cleared paths [3], to solve an integer network flow problem [66,68], to involve

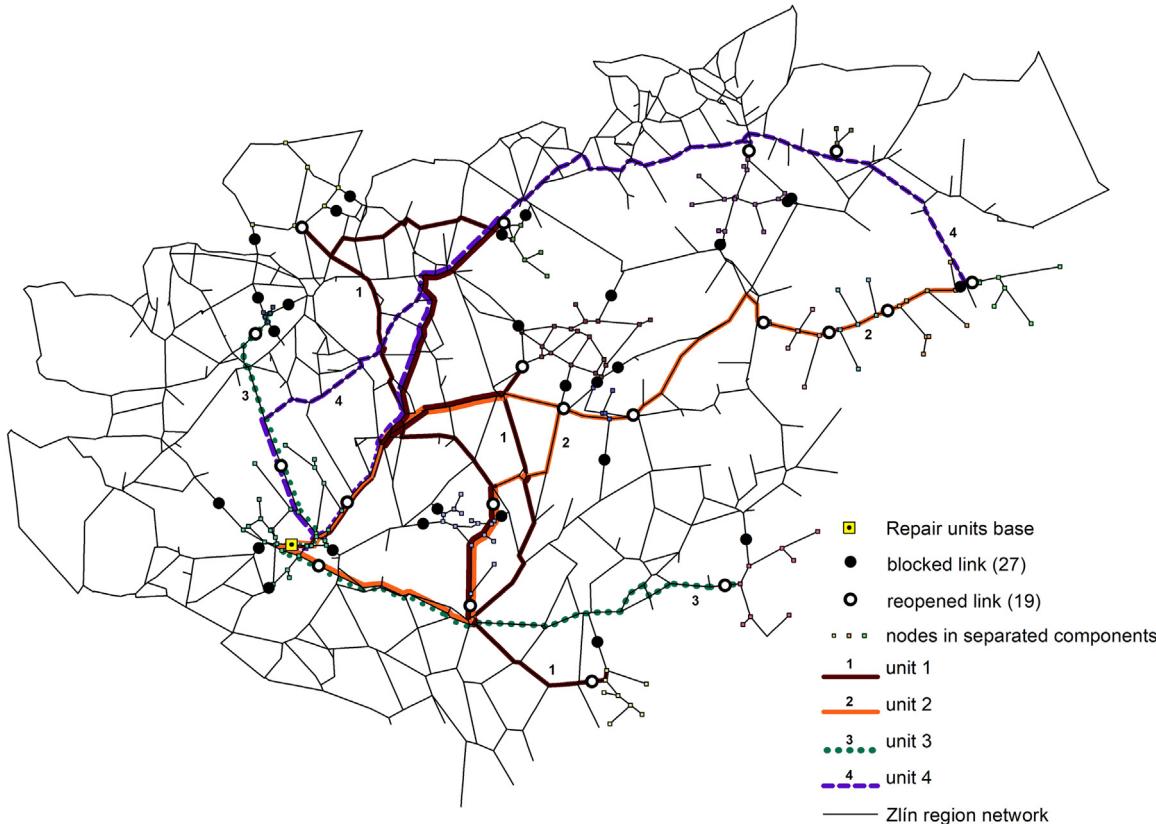


Fig. 5. The best routes for 4 repair units.

Table 10
Results for (α, β) .

α	β	Average time of repair unit no. 1 [s]	Average time of repair unit no. 2 [s]	Average time of repair unit no. 3 [s]	Average time of repair unit no. 4 [s]	Average progress of repair unit no. 1 [people]	Average progress of repair unit no. 2 [people]	Average progress of repair unit no. 3 [people]	Average progress of repair unit no. 4 [people]
1	1	9922.00	9116.63	8826.66	7972.25	3,008,178	276,561	219,705	106,323
1	2	9922.00	9116.41	8866.69	8105.25	3,214,166	274,409	218,796	104,039
1	3	9922.00	9122.33	8915.12	8233.95	3,193,298	275,411	219,071	104,431
2	1	9922.00	9139.10	8945.20	8295.04	2,990,530	274,443	219,385	110,445
2	2	9922.00	9135.66	8938.48	8301.93	3,043,808	273,014	219,309	109,339
2	3	9922.00	9133.64	8942.17	8323.10	3,088,522	272,259	218,849	108,480
3	1	9922.00	9135.42	8955.51	8314.23	3,112,510	272,883	218,699	110,021
3	2	9922.00	9133.63	8949.50	8310.77	3,130,423	272,412	218,584	109,521
3	3	9922.00	9133.01	8952.61	8323.05	3,138,889	272,796	218,399	108,911

Table 11
Results for the number of iterations.

Number of iterations	Average time of a repetition [s]	Average time of repair unit no. 1 [s]	Average time of repair unit no. 2 [s]	Average time of repair unit no. 3 [s]	Average time of repair unit no. 4 [s]	Average progress of repair unit no. 1 [people]	Average progress of repair unit no. 2 [people]	Average progress of repair unit no. 3 [people]	Average progress of repair unit no. 4 [people]
1000	1605.29	9922.00	9147.00	8944.85	8324.96	3,179,553	276,786	216,362	107,095
1200	1924.83	9922.00	9141.91	8935.21	8325.59	3,192,612	276,962	216,717	106,256
1400	2245.96	9922.00	9138.08	8930.04	8323.88	3,199,093	277,419	217,310	105,415
1600	2569.35	9922.00	9135.18	8919.92	8304.55	3,201,857	277,417	217,544	104,784
1800	2891.99	9922.00	9132.94	8912.34	8293.15	3,210,164	278,097	217,590	104,586
2000	3216.35	9922.00	9131.05	8906.66	8285.80	3,212,435	277,836	217,837	104,238

Table 12
Analysis of the results.

Best found solution [s]	[9922, 9114.3, 8836.22, 8201.73]
Average percentage improvement and relative standard deviation [(%, %)]	(−6.95, 3.14), (−10.31, 5.52), (−3.81, 9.82), (20.63, 24.18)
Average times of repair units and their relative standard deviations [(s, %)]	(9922, 0), (9114.3, 0.03), (8836.2, 1.67), (8201.7, 5.8)
Average progresses of repair units and their relative standard deviations [(people, %)]	(3,241,379, 18.03), (276,170, 16.04), (220,748, 8.54), (100,368, 0.0)

further distribution relief [45,51,65,67], limited resources [39,43,44], time constraints and other related operating constraints [69].

The approaches mentioned above involve both deterministic and stochastic algorithms, such as a fuzzy genetic algorithm [13], GRASP and VNS metaheuristics [43], dynamic programming and an iterated greedy-randomized constructive procedure [45], a rule-based constructive heuristic [47], a Markov decision process reconstruction [11], a greedy algorithm using critical links [42], a heuristic algorithm based on problem decomposition and variable fixing techniques [69,70], ant colony optimization algorithm [65,68], simulated annealing [27], tabu search [27] and genetic algorithms [5,62]. The deterministic algorithms usually suffer from the complexity of the types of problems which often results in high computational times. The stochastic algorithms, in contrast, are not usually able to provide the best results.

A number of the papers mentioned above drew attention to the events which result from natural disasters and which required not only road network reconstruction but also immediate distribution of relief with limited resources. The network can, however, disintegrate under various events which can be less catastrophic and which provide more freedom for repair units (for instance the interrupted links can only be overcome by repair units without any repair works). In this paper we therefore decided to study the situation wherein the repair units can overcome blocked links without loss of time, but at the same time the links are not suitable for common traffic. To determine optimal routes for the units, we introduced the stochastic algorithm based on ACO. The main issue of any stochastic optimization algorithm is that we can only prove its convergence to the optimal solution, but we cannot ensure

that the solution is found in finite time.

Due to this fact, and due to the introduced optimization problem based upon the special ranking (3) – (5), we decided to test the algorithm on a small network. We chose the Sioux-Falls network [38,72] whose data are freely available. We tested the algorithm for four repair units. Section 3 demonstrated that the results were very satisfactory because the algorithm identified the best solution, which can be visually verified, in 94% cases with a very low relative standard deviation which means that the solutions which were not optimal were also very close to the optimal solution. It was also apparent that the improvement of the longest route, compared with the initial solution, was not large (1.4%), but that the improvement in the second and third longest routes began to be significant (7–9%). In contrast, we can also argue that the savings algorithm, which constructs the initial solution, provides a relatively good first approximation of the optimal solution. Thanks to a low computational time, (0.5 s per computation) it is not, however, necessary to restrict the entire computation only to the construction of the initial solution.

We further applied the algorithm to the Zlín network which was approximately two times larger than the largest networks in the above-mentioned papers. We also applied various scenarios at the same time. The computational intensity was therefore high. We focused primarily on an analysis of the algorithm and thus do not use a parallel computation for one repetition. We further used 3-opt for the local search which is a very good method but also computationally intensive. Despite this fact the average times for one computation ranged from less than 5 min up to 1 h (see Table 5). As can be seen in Table 5, the results were already satisfactory after 5 min of computation. We are of the opinion that parallelization of the construction of the routes can further improve the speed of the algorithm.

4.1. Scenario 1

In Section 3.2.1 we studied the first scenario and introduced possible applications in optimization of distribution relief or in routes for heavy machinery. We also provided an analysis which enabled us to choose the most suitable values of (α, β) and the number of iterations.

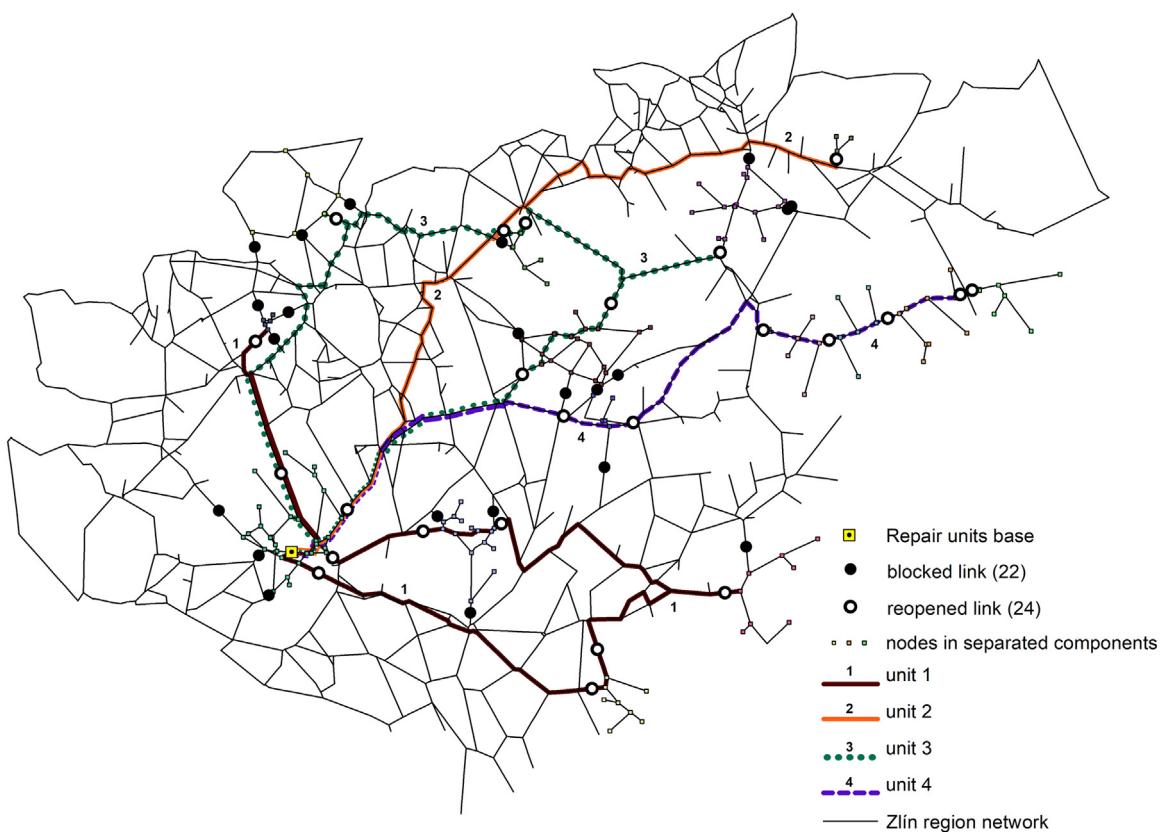


Fig. 6. The best paths for four repair units.

In Table 5 we also provided the average times for one computation together with the achieved results which can also be used by the administrator if he/she wants to influence the length of the computation. In Table 6 it is of interest to note that the longest route is found after each computation (zero relative standard deviation). The other positive fact is that the additional routes have very low relative standard deviations, which means that almost any computation found a result which is very close to the best found solution. In contrast, improvement to the initial solution is not negligible (5.5–10.6%).

4.2. Scenario 2

In the scenario we assumed that the repair units must repair the blocked links which they use as the first unit. In the scenario the times needed to repair the blocked links dominated the times to drive through other links. The times should not be neglected, however, because they can form up to 10% of the final times (compare Tables 6, 9). In Section 2.3.2 we repeated the process from Section 2.3.1 which enabled us to identify the most suitable values of (α, β) and the number of iterations. The results indicate higher relative standard deviations than in Scenario 1 (6–23.4%) which is emphasized by the best found solution which is significantly better than the average values. The next remarkable thing is the significant improvement of the initial solution (26.8–38%) which indicates that the algorithm constructing the initial solution worked much worse than in Scenario 1.

4.3. Scenario 3

In this scenario we used a different algorithm for the construction of the initial solution and recommendations (see the definition of heuristics in Section 2.3.2). This should influence the direction of the computation of the algorithm despite the same formulation of the optimization problem (3)–(5). This follows from the recommendations

which sometimes contradict the optimality criteria. The aim of the recommendations was to cause the algorithm to prefer, in a certain sense, more important components (a higher number of inhabitants). The first thing which should not be surprising is that the best found results are a bit worse than in Scenario 1, despite the large number of iterations. The standard deviations of the average results remain low, however. It contrasts with the much higher relative standard deviations for the progresses. We consider the results positive because repeated computation was able to produce more variable values of the progress under low variability of the best found solutions. This provides the administrator with the possibility to apply additional tools which are able to choose the most suitable ones among the results. It is important to mention that the greater variability is not only caused by the three components with the highest number of people. The components form 15%, 8% and 5% of the first three average progress numbers but the relative standard deviations are higher. From the results depicted in Figs. 4–6, it is also apparent that the results are far from intuitive and can be hardly achieved without any algorithm.

5. Conclusion

The primary aim of this paper was to improve current knowledge in relation to the remedial work which takes place in the first phase after a disaster strikes. We introduced a new optimization approach of network recovery. It was demonstrated with a road network where several links were blocked in order to break down the network into several components. The primary aim of this process was to identify the optimal sequences of links for every repair unit which needed to be repaired or equivalently the optimal sequences of components which had to be visited. For the optimal criterion we required an optimal exploitation of the time of all the repair units. The process of generation of the units' routes was influenced by the length of the best found route and the currently best constructed routes, by the distance from the nodes in

other components and by the importance of the components. As the criterion of importance, we chose the number of people living in the component although various other criteria can be used.

We chose the ACO approach because the problem ranks among NP-hard problems. It leads to the restricted applicability of deterministic algorithms for large events affecting larger networks. It is therefore necessary to select among metaheuristics. We decided here to use the modification of the ACO algorithm and demonstrated the viability of the approach.

The results suggested that only a portion (between 19 and 25 from 46) of the originally blocked road links had to be reopened in the first phase to ensure network recovery. The overall computation time depends on the accuracy of the result required by the administrator. One of the advantages of our algorithm is that it can be easily parallelized. Not only can any computation be assigned to a core but also each of the ant colonies can be assigned to one core. The other advantage is that the computation of the algorithm can be easily controlled by the administrator. Using various parameters and the higher variability of the results in Scenario 3 enables the administrator to use more refined tools to select the optimal solution for him/her.

To summarize the results of this paper, there are seven points which this work adds to the current knowledge about the topic:

1. Reduction of the problem size using only the boundary nodes of the components instead of the entire network.
2. Construction of a complete graph and thus significant reduction of the computational demand.
3. An introduction of a ranking which enables the ranking of n -tuples and thus minimize the time for all repair units (compare with [1,36], where only the maximal time was minimized).
4. Using a metaheuristic based on the ACO, we were able to solve the optimization problem for larger networks (723 nodes in the Zlin region versus 401 nodes in [45] and thus reduce the impact of the NP-hard nature of these problems.
5. New optimality criteria and modification of the known ACO algorithm replacing the constraints with recommendations.
6. An alternative to multi-objective optimization using the Pareto front.
7. Simple modifications and natural additions of the constraints to the algorithm without affecting the computational time.

Acknowledgements

This work was financed by the project VI20152019049 "RESILIENCE 2015: Dynamic Resilience Evaluation of Interrelated Critical Infrastructure Subsystems", supported by the Ministry of the Interior of the Czech Republic and the project of Transport Research Centre (OP R&D for Innovation No. CZ.1.05/2.1.00/03.0064).

References

- [1] V. Akbari, F.S. Salman, Multi-vehicle synchronized arc routing problem to restore post-disaster network connectivity, *Eur. J. Oper. Res.* 257 (2017) 625–640.
- [2] V. Akbari, F.S. Salman, Multi-vehicle prize collecting arc routing for connectivity problem, *Comput. Oper. Res.* 82 (2017) 52–68.
- [3] D.T. Aksu, L. Ozdamar, A mathematical model for post-disaster road restoration: Enabling accessibility and evacuation, *Transp. Res. Part E: Logist. Transp. Rev.* 61 (2014) 56–67.
- [4] N. Altay, W. Green, OR/MS research in disaster operations management, *Eur. J. Oper. Res.* 175 (1) (2006) 475–493.
- [5] M. Arimura, T. Tamura, K. Saito, Application of genetic algorithms model for road investment of restoration planning, *J. East. Asia Soc. Transp. Stud.* 2 (1999) 55–69.
- [6] S.A. Bagloee, M. Asadi, Prioritizing road extension projects with interdependent benefits under time constraint, *Transp. Res. Part A: Policy Pract.* 75 (2015) 196–216.
- [7] M. Bříš, R. Vodák, J. Kubecák, M. Bílová, J. Sedoník, Evaluating road network damage caused by natural disasters in the Czech Republic between 1997 and 2010, *Transport. Res. Part A: Policy Pract.* 80 (2015) 90–103.
- [8] F. Bono, E. Gutiérrez, A network-based analysis of the impact of structural damage on urban accessibility following a disaster: the case of the seismically damaged Port Au Prince and Carrefour urban road networks, *J. Transp. Geogr.* 19 (2011) 1443–1455.
- [9] G.G. Brown, A.L. Vassiliou, Optimizing disaster relief: real-time operational and tactical decision support, *Nav. Res. Logist. (NRL)* 40 (1) (1993) 1–23.
- [10] B. Cavdaroglu, E. Hammel, J.E. Mitchell, T.C. Sharkey, W.A. Wallace, Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems, *Ann. Oper. Res.* 203 (1) (2013) 279–294.
- [11] M. Celik, O. Ergun, P. Keskinocak, The post-disaster debris clearance problem under incomplete information, *Oper. Res.* 63 (1) (2015) 65–85.
- [12] S.E. Chang, N. Nojima, Measuring post-disaster transportation system performance: the 1995 Kobe earthquake in comparative perspective, *Transp. Res. Part A* 35 (2001) 475–494.
- [13] Y. Chen, G. Tzeng, A fuzzy multi-objective model for reconstructing post-earthquake road-network by genetic algorithm, *Int. J. Fuzzy Syst.* 1 (2) (1999) 85–95.
- [14] G. Clarke, J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, *Oper. Res.* 12 (1964) 568–581.
- [15] J.L. Cohon, R.L. Church, D.P. Sheer, Generating multiobjective trade-offs: an algorithm for bicriterion problems, *Water Resour. Res.* 15 (5) (1979) 1001–1010.
- [16] G.B. Dantzig, J.H. Ramser, The truck dispatching problem, *Manag. Sci.* 6 (1) (1959) 80–91.
- [17] DfT, Department for Transport, Transport Resilience Review: A Review of the Resilience of the Transport Network To Extreme Weather Events, Department for Transport, London, 2014 (https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/335115/transport-resilience-review-web.pdf).
- [18] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of co-operating agents, *IEEE Trans. Syst. Man Cybern. Part B-Cybern.* 26 (1) (1996) 29–41.
- [19] M. Dorigo, L.M. Gambardella, Ant colonies for the travelling salesman problem, *Biosystems* 43 (2) (1997) 73–81.
- [20] M. Dorigo, T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, MA, USA, 2004.
- [21] M. Dorigo, C. Blumb, Ant colony optimization theory: a survey, *Theor. Comput. Sci.* 344 (2005) 243–278.
- [22] O. Ergun, G. Karakus, P. Keskinocak, J. Swann, M. Villarreal, Operations Research to Improve Disaster Supply Chain Management. Wiley Encyclopedia of Operations Research and Management Science, John Wiley & Sons, Inc, 2010.
- [23] S. Even, *Graph Algorithms*, 2nd ed., Cambridge University Press, New York, NY, USA, 2011.
- [24] R. Faturechi, E. Miller-Hooks, Travel time resilience of roadway networks under disaster, *Transp. Res. Part B: Methodol.* 70 (2014) 47–64.
- [25] C.-M. Feng, T.-C. Wang, Highway emergency rehabilitation scheduling in post-earthquake 72h, *J. East. Asia Soc. Transp. Stud.* 5 (2003) 3276–3285.
- [26] C.-M. Feng, T.-C. Wang, Seismic emergency rehabilitation scheduling for rural highways, *Transp. Plan. J.* 34 (2) (2005) 177–210.
- [27] F. Fiedrich, F. Gehbauer, U. Rickers, Optimized resource allocation for emergency response after earthquake disasters, *Saf. Sci.* 35 (2000) 41–57.
- [28] L.M. Gambardella, É.D. Taillard, G. Agazzi, MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows, in: D. Corne, M. Dorigo, F. Glover (Eds.), *New Ideas in Optimization*, 1999 McGraw-Hill, London, UK, 1999, pp. 63–76.
- [29] S. Gottlich, A. Klar, Modeling and optimization of scalar flows on networks, in: L. Ambrosio, A. Bressan, D. Helbing, A. Klar, E. Zuazua (Eds.), *Modelling and Optimisation of Flows on Networks*, 2002 Book Series: Lecture Notes in Mathematics, Cetraro, Italy, 2009, pp. 395–461.
- [30] W.J. Gutjahr, A graph-based ant system and its convergence, *Future Gener. Comput. Syst.* 16 (8) (2000) 873–888.
- [31] W.J. Gutjahr, ACO algorithms with guaranteed convergence to the optimal solution, *Inf. Process. Lett.* 82 (3) (2002) 145–153.
- [32] D. Helbing, Traffic and related self-driven many-particle systems, *Rev. Mod. Phys.* 73 (4) (2001) 1067–1141.
- [33] D. Helbing, Order and disorder in traffic and self-driven many-particle systems, in: S. Boccaletti, B.J. Gluckman, J. Kurths, L.M. Pecora, M.L. Spano (Eds.), *Experimental Chaos Book Series: AIP Conference Proceedings* 622, 239–250. Published: 2002, Conference: 6th Experimental Chaos Conference, 2002.
- [34] B. Immers, A. Bleukx, J. Stada, C. Tamperie, I. Yperman, Robustness and Resilience of Road Network Structures, Katholieke Universiteit Leuven, Heverlee, Belgium, 2005 (<http://www.kuleuven.be/traffic/dwn/P2004A.pdf>).
- [35] M. Karlaftis, K. Kepaptsoglou, S. Lambropoulos, Fund allocation for transportation network recovery following natural disasters, *J. Urban Plan. Dev.* 133 (1) (2007) 82–89.
- [36] M. Kasaei, F.S. Salman, Arc routing problems to restore connectivity of a road network, *Transp. Res. Part E* 95 (2016) 177–206.
- [37] S. Lammer, H. Kori, K. Peters, D. Helbing, Decentralised control of material or traffic flows in networks using phase-synchronisation, *Phys. A: Stat. Mech. Appl.* 363 (1) (2006) 39–47.
- [38] L.J. LeBlanc, E.K. Morlok, W.P. Pierskalla, An efficient approach to solving the road network equilibrium traffic assignment problem, *Transp. Res.* 9 (1975) 309–318.
- [39] F. Liberatore, M.T. Ortúñoz, G. Tirado, B. Vitoriano, M.P. Scaparra, A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in humanitarian logistics, *Comput. Oper. Res.* 42 (2014) 3–13.
- [40] S. Lin, Computer solutions of the traveling salesman problem, *Bell Syst. Tech. J.* 44 (1965) 2245–2269.
- [41] S. Lin, B.W. Kernighan, An effective heuristic algorithm for the traveling-salesman problem, *Oper. Res.* 21 (1973) 498–516.
- [42] G. Lu, Y. Xiong, C. Ding, Y. Wang, An optimal schedule for urban road network

- repair based on the greedy algorithm, PLoS One 11 (10) (2016) e0164780, <http://dx.doi.org/10.1371/journal.pone.0164780>.
- [43] P.A. Maya Duque, K. Sørensen, A grasp metaheuristic to improve accessibility after a disaster, OR Spectr. 33 (3) (2011) 525–542.
- [44] P.A. Maya Duque, S. Coene, P. Goos, K. Sørensen, F. Spieksma, The accessibility arc upgrading problem, Eur. J. Oper. Res. 224 (3) (2013) 458–465.
- [45] P.A. Maya Duque, I.S. Dolinskaya, K. Sørensen, Network repair crew scheduling and routing for emergency relief distribution problem, Eur. J. Oper. Res. 248 (2016) 272–285.
- [46] P.M. Murray-Tuite, A Comparison of Transportation Network Resilience under Simulated System Optimum and User Equilibrium Conditions. Proceedings of the 2006 Winter Simulation Conference, Monterey, California, 2006, 1398–1405, 2006.
- [47] L. Ozdamar, D.T. Aksu, B. Ergunes, Coordinating debris cleanup operations in post disaster road networks, Socio-Econ. Plan. Sci. 48 (4) (2014) 249–262.
- [48] H. Paessens, The savings algorithm for the vehicle routing problem, Eur. J. Oper. Res. 34 (1988) 336–344.
- [49] K. Ransikarbum, S.J. Mason, Goal programming-based post-disaster decision making for integrated relief distribution and early-stage network restoration, Int. J. Prod. Econ. 182 (2016) 324–341.
- [50] M. Reimann, K. Doerner, R.F. Hartl, D-ants: savings based ants divide and conquer the vehicle routing problems, Comput. Oper. Res. 31 (4) (2004) 563–591.
- [51] H. Sahin, B.Y. Kara, O. Karasan, Debris removal during disaster response: a case for Turkey, Socio-Econ. Plan. Sci. 53 (2016) 49–59, <http://dx.doi.org/10.1016/j.seps.2015.10.003>.
- [52] T. Sato, K. Ichii, Optimization of post-earthquake restoration of lifeline networks using genetic algorithms, Jpn. Soc. Civil. Eng. 537 (I–35) (1996) 245–256.
- [53] D.M. Scott, D.C. Novak, L. Aultman-Hall, F. Guo, Network robustness index: a new method for identifying critical links and evaluating the performance of transportation networks, J. Transp. Geogr. 14 (3) (2006) 215–227.
- [54] A. Soltani-Sobh, K. Heaslip, J. El Khoury, Estimation of road network reliability on resiliency: an uncertain based model, Int. J. Disaster Risk Reduct. 14 (4) (2015) 536–544.
- [55] A. Soltani-Sobh, K. Heaslip, A. Stevanovic, J. El Khoury, Z. Song, Evaluation of transportation network reliability during unexpected events with multiple uncertainties, Int. J. Disaster Risk Reduct. 17 (2016) 128–136.
- [56] A. Soltani-Sobh, K. Heaslip, P. Scarlatos, E. Kaisar, Reliability based pre-positioning of recovery centers for resilient transportation infrastructure, Int. J. Disaster Risk Reduct. 19 (2016) 324–333.
- [57] T. Stützle, H.H. Hoos, The MAX-MIN Ant System and local search for the traveling salesman problem, in: T. Back, Z. Michalewicz, X. Yao (Eds.), Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), 309–314. Piscataway, NJ, IEEE Press, 1997.
- [58] T. Stützle, Local Search Algorithms for Combinatorial Problems: analysis, Improvements, and New Applications (Dissertations in Artificial Intelligence 220), IOS Press, Amsterdam, The Netherlands, 1999<<http://iridia.ulb.ac.be/~stuetzle/publications/Thesis.ThomasStuetzle.pdf>>.
- [59] T. Stützle, H.H. Hoos, MAX-MIN Ant System, Future Gener. Comput. Syst. 16 (8) (2000) 889–914.
- [60] T. Stützle, M. Dorigo, A short convergence proof for a class of ACO algorithms, IEEE Trans. Evolut. Comput. 6 (4) (2002) 358–365.
- [61] J.L. Sullivan, D.C. Novak, L. Aultman-Hall, D.M. Scott, Identifying critical road segments and measuring system wide robustness in transportation networks with isolating links: a link-base capacity-reduction approach, Transp. Res. Part A: Policy Pract. 44 (5) (2010) 323–336.
- [62] T. Tamura, H. Sugimoto, T. Kamimae, Application of genetic algorithms to determining priority of urban road improvement, Jpn. Soc. Civ. Eng. 482 (IV–22) (1994) 37–46.
- [63] L. Van Wassenhove, New interesting POM cases from Europe, POMS Chron. 10 (2003) 19.
- [64] Y. Wang, R. Xiao, An ant colony based resilience approach to cascading failures in cluster supply network, Physica A 462 (2016) 150–166.
- [65] B. Xu, Y. Song, An Ant colony-based heuristic algorithm for joint scheduling of post-earthquake road repair and relief distribution, TELKOMNIKA 13 (2) (2015) 632–643.
- [66] S. Yan, Y.-L. Shih, A time-space network model for work team scheduling after a major disaster, J. Chin. Inst. Eng. 30 (1) (2007) 63–75.
- [67] S. Yan, Y.-L. Shih, Optimal scheduling of emergency roadway repair and subsequent relief distribution, Comput. Oper. Res. 36 (6) (2009) 2049–2065.
- [68] S. Yan, Y.-L. Shih, An ant colony system-based hybrid algorithm for an emergency roadway repair time-space network flow problem, Transportmetrica 8 (5) (2012) 361–386.
- [69] S. Yan, C.K. Lin, S.Y. Chen, Optimal scheduling of logistical support for an emergency roadway repair work schedule, Eng. Optim. 44 (9) (2012) 1035–1055.
- [70] S. Yan, C.K. Lin, S.Y. Chen, Logistical support scheduling under stochastic travel times given an emergency repair work schedule, Comput. Ind. Eng. 67 (2014) 20–35.
- [71] X. Zhang, E. Miller-Hooks, K. Denny, Assessing the role of network topology in transportation network resilience, J. Transp. Geogr. 46 (2015) 35–45.
- [72] Jianyunli/GitHub, Sioux Falls Network [Data file], Retrieved from: <<https://github.com/bstabler/TransportationNetworks/tree/master/SiouxFalls>>, 2016.