

A method to optimize evacuation instructions

Olga L. Huibregtse · Serge P. Hoogendoorn ·
Andreas Hegyi · Michiel C. J. Bliemer

Published online: 30 March 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract In this paper, an optimization method is presented for instructions to evacuate by car the population of a region threatened by a hazard. By giving the optimized instructions to the evacuees, traffic conditions and, therefore, the evacuation efficiency are optimized. The instructions, containing a departure time, a safe destination, and a route, are created using an optimization method consisting of two phases: the generation of the search space and the algorithm AES⁺ evacuation, a version of ant colony optimization. The main contributions of the proposed optimization method are the unique approach to generate the search space in which network degeneration is taken into consideration, the possibility to optimize instructions under the assumption of both full and partial compliance of the evacuees with the instructions, and the flexibility in the sense that the user of the method can define his or her own objective function and choose a suitable traffic simulation model. The paper contains a comprehensive case study. The case study shows that the effectiveness of the optimized instructions is more than doubled when compared with the effectiveness of instructions set up by straightforward rules (like evacuating to the nearest destination using the shortest route). Further, the case study shows that the number of arrivals under optimized,

O. L. Huibregtse (✉) · S. P. Hoogendoorn · A. Hegyi · M. C. J. Bliemer
Department of Transport and Planning, Faculty of Civil Engineering and Geosciences,
Delft University of Technology, Delft, The Netherlands
e-mail: O.L.Huibregtse@tudelft.nl

S. P. Hoogendoorn
e-mail: S.P.Hoogendoorn@tudelft.nl

A. Hegyi
e-mail: A.Hegyi@tudelft.nl

M. C. J. Bliemer
e-mail: M.C.J.Bliemer@tudelft.nl

but possibly sub-optimal instructions is equal to at least 90% of the theoretical upper bound on this number of arrivals.

Keywords Evacuation · Instructions · Optimization · Ant Elitist System⁺

1 Introduction

Authorities have to be prepared for disasters like bush fires and floods. This includes creating a plan to evacuate people from a threatened region by car, to avoid as many casualties as possible. People can make their own decisions about where to go and when to leave, but these decisions will most probably not be system-optimal for the following reasons: (1) people have a lack of information about, and a lack of experience with, the extreme situation (i.e., the demand is high as compared to the daily situation and the hazard may cause inaccessible roads) and (2) people act out of a user- instead of system-optimal thinking. If people act out of a user-optimal thinking, each evacuee acts such that it is the best for himself, for example by minimizing his own travel time. This will probably be not optimal for the system. If they act out of a system-optimal thinking, each evacuee acts such that it is the best for the system, meaning that an evacuee accepts a higher travel time than his minimal travel time, to achieve a better system performance (e.g., less time needed to evacuate all the people). Giving optimized departure time, route and destination instructions to the people, can solve the lack of information and can steer the evacuees in the direction of a system-optimal traffic assignment. Therefore, giving optimized instructions to the evacuees will most probably lead to a more effective evacuation from a systems perspective (i.e., shorter evacuation times, fewer casualties).

Several methods exist to optimize evacuation instructions under the assumption of full compliance of the evacuees with the instructions. These methods can be distinguished based on the kind of instructions that are optimized (departure time, route, and/or destination instructions). An overview of these methods is given here. In some of these methods, the traffic assignment or the flows are optimized instead of the instructions. However, under the assumption of full compliance, optimal instructions are the same as the optimal traffic assignment and will result in the same traffic flows. Therefore, also methods that optimize traffic assignments or flows are included in the overview (as if they optimize instructions). In, for example, [Miller-Hooks and Sorrel \(2008\)](#) and [Stepanov and Smith \(2009\)](#), the route instructions are optimized, while in [Saadatseresht et al. \(2009\)](#), the destination instructions are optimized. Both the route and the destination instructions are optimized in [Liu et al. \(2006\)](#) and [Ben-Tal et al. \(2009\)](#). For all these problems holds that the non-optimized components (out of the departure time, route and destination instructions) are not included in the problem or are assumed to be fixed (for example, all evacuees are instructed to follow the shortest route). Departure time, destination and route instructions are considered simultaneously in [Sbayti and Mahmassani \(2006\)](#), [Afshar and Haghani \(2008\)](#), and [Dixit and Radwan \(2009\)](#).

The previous paragraph showed that methods exist to optimize evacuation instructions, among which methods to optimize departure time, route, and destination

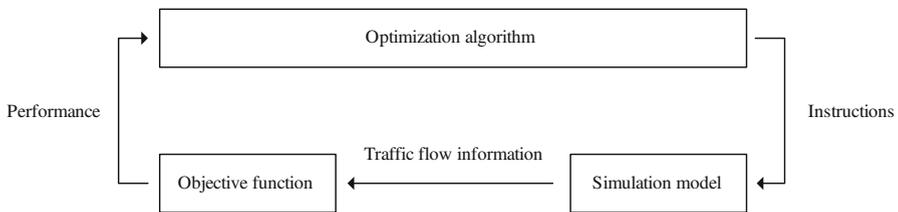


Fig. 1 Simulation optimization approach

instructions. In [Dixit and Radwan \(2009\)](#), a kind of non-compliance is taken into consideration in the sense that the resulting departure times are a deterministic function of the instructed departure time and the time of the day. All other methods mentioned in the previous paragraph can only be used to optimize instructions under the assumption of full compliance of the evacuees with the departure time, route, and destination instructions, a highly unrealistic assumption. As far as the authors are aware, no method exists to optimize departure time, route, and destination instructions under partial compliance.

The method presented in this paper contributes to the research in the field because it can be used to optimize instructions not only under the assumption of full compliance, but also under the assumption of partial compliance. Under the assumption of partial compliance, the preferences of the evacuees for their departure times, routes, and destinations are taken into consideration. In the developed method, departure time, route, and destination instructions are simultaneously optimized using a simulation optimization approach. An iteration of this approach is illustrated in [Fig. 1](#): instruction sets constructed in the optimization algorithm are evaluated by a simulation model and an objective function, resulting in performance values of the instruction sets as input for the construction of new instruction sets. The result of the method is an instruction set which performance is most probably not equal to the performance of the optimal instruction set, but approaches this optimal performance.

A strength of the method is that the structure is such that the objective function and the simulation model can be chosen by the user of the method. This has the advantage that advanced models and functions can be used, by which it is for example possible to deal with uncertainty (by the choice of the objective function). A requirement for the simulation model is that the model can deal with departure time, destination, and route instructions under partial compliance. Further, the required output of the simulation model depends on the choice of the objective function. Another strength of the method is that the method can be applied to an arbitrary region and hazard. The only condition for the hazard is that there has to be a possibility to instruct the evacuees. Thus, the hazard has to be predictable or there has to be enough time during the development of the disaster to instruct the people.

The second contribution of the paper is the approach to generate the search space whereby network degeneration is taken into consideration. A search space, consisting of a selection of all possible combinations of departure times, routes, and destinations, has to be generated before the optimization algorithm can be applied. This search space influences both the feasibility of the optimized instructions and the applicability of the

optimization method. A complicated factor in the generation of the search space for the evacuation problem is the network degeneration caused by the hazard. An example of the influence of the network degeneration on the search space is a route that can be instructed in combination with early departure times, but not in combination with late departure times, because part of the route is flooded. In this paper, an approach is developed to generate the search space whereby network degeneration is taken into account and such an approach has not been developed before, as far as the authors are aware.

The third contribution of the paper is a comprehensive case study, wherein the results of an application of the optimization method are analyzed. The objectives of the analysis are to show the performance of the method, the need to give instructions to the evacuees, the need to optimize these instructions, patterns in the optimized departure time, route, and destination instructions, the near-optimality of the performance of the optimized instructions, and the influence of the parameter settings.

To summarize, the contributions of the paper are

- the development of an optimization method for evacuation instructions (departure times, routes, and destinations) that can deal with the assumption of partial compliance of the evacuees with the instructions, whereby the objective function and the simulation model can be chosen by the analyst, and that can be applied to an arbitrary region and hazard;
- the development of an approach to generate the search space whereby network degeneration is taken into consideration;
- a comprehensive case study.

In Sect. 2, the evacuation problem is formulated (Sect. 2.1) and the complexity of the problem is discussed (Sect. 2.2). The optimization method is presented in Sect. 3. The method consists of the generation of the search space (Sect. 3.1) and the algorithm EAS⁺ evacuation (Sect. 3.2). Not only the objective function and the simulation model can be chosen by the analyst, also some input of the optimization method, the so-called *heuristic information*, can be chosen by the analyst. The objective function (Sect. 4.1), the heuristic information (Sect. 4.2), and the simulation model (Sect. 4.3) used for the case study presented in this paper are discussed. The case study follows in Sect. 5. The paper ends with conclusions and future research directions in Sect. 6.

2 Problem formulation and complexity

This section starts with the description of the evacuation problem (Sect. 2.1). Then, the complexity of the problem is discussed (Sect. 2.2): it will be showed that the evacuation problem is a NP hard problem. Because of this NP hardness, it holds by definition that it is unlikely that someone will ever find an efficient algorithm leading to the exact solution. Knowledge of the proof given in Sect. 2.2 is not required for the understanding of the rest of the paper.

2.1 The evacuation problem

Let B denote the total number of people to be evacuated in a threatened region. This region contains origins $r \in R$, with corresponding number of residents, B_r . These residents have to be evacuated via a network which is assumed to be totally accessible before the hazard strikes the network and to become partly accessible when the hazard has struck parts of the network.

The evacuees are assumed to be divided into groups $g \in \{1, \dots, G\}$, where all evacuees in a group get the same evacuation instruction e_g . Each group belongs to a specific origin, and the number of evacuees in group g is denoted by B_r^g . The number of evacuees in each group is equal to a constant group size, B^{constant} , except a possible rest group for each origin when B_r is no multiple of B^{constant} . By varying the group size, the complexity of the problem, and the feasibility of the evacuation instructions following on the application (related to the number of different instructions) are influenced. By setting B^{constant} equal to 1, each group consists of one evacuee and solving the problem would result in individual instructions.

The problem is to create an optimal set of evacuation instructions, E , containing the evacuation instructions $e_g \in E$. Each instruction contains a departure time, route and destination, which are simultaneously optimized. Because the destination instruction is implicitly included in the route instruction, each instruction only contains a departure time, k_{e_g} , and a route p_{e_g} . Therefore, all destinations are connected with an artificial super destination, with links having a length equal to zero and infinite capacity, as applied e.g. in Chiu et al. (2007).

The mathematical formulation of the problem is

$$\{k_{e_g}, p_{e_g} : e_g \in E\} = \arg \max_{k_{e_g} \in K^{\text{pos}}, p_{e_g} \in P^{\text{pos}}} f_E \tag{1}$$

where f_E is the fitness of the set of evacuation instructions E , K^{pos} and P^{pos} respectively are the set of all possible departure times and the set of all possible routes. This fitness value follows from the simulation model and the objective function chosen by the analyst. The fitness can for example be related to the number of arrivals over time.

2.2 Complexity of the problem

In this section, it will be showed that the evacuation problem is an *NP hard problem*, whereby by definition holds that it is unlikely that someone will ever find an efficient algorithm (for example solvable in polynomial time) leading to the exact solution. A problem is NP hard if a restricted version of the problem is NP complete. In that case, the original problem could never be easier solved than the restricted problem. If the original problem could be efficiently solved, the restricted problem could also be efficiently solved and would therefore not be NP complete.

The NP hardness of the evacuation problem is discussed by showing that an NP complete problem, namely the arc-disjoint path (ADP) problem, is a restricted version of

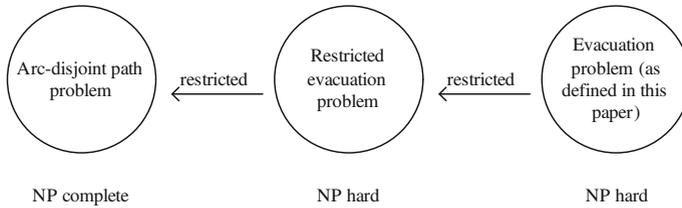


Fig. 2 NP hardness evacuation problem in relation to the NP completeness of the arc-disjoint path problem

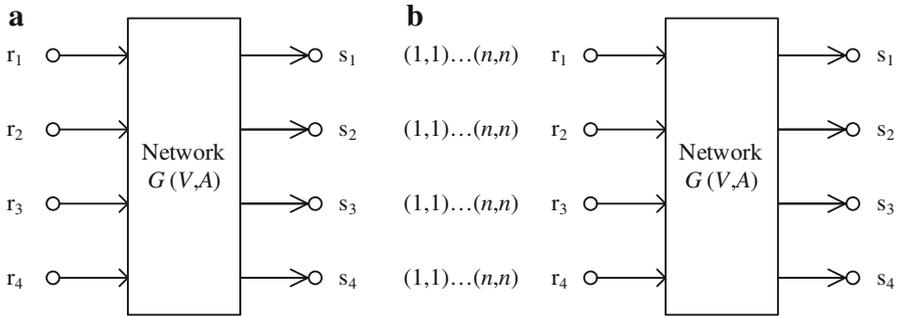


Fig. 3 **a** The ADP problem, **b** the restricted evacuation problem

the evacuation problem. In this discussion, the so-called *restricted evacuation problem* is introduced: a restricted version of the evacuation problem introduced in the previous section. The restricted evacuation deviates from the evacuation problem by the optimization of route instructions instead of the optimization of departure time, destination and route instructions, and by the specification of the objective function to the minimization of the time needed to evacuate all the people (the evacuation time). By introducing this problem, the NP hardness of the evacuation problem is discussed by showing that the ADP problem is a restricted version of the restricted evacuation problem. Figure 2 shows the relation between the three problems and their complexity.

This paragraph shows that the ADP problem is a restricted version of the restricted evacuation problem. In the ADP problem, a directed graph and a collection (r_z, s_z) of x origin (r) destination (s) pairs (where $z = 1, \dots, x$) are given. In the problem (Fig. 3a), a collection of x arc-disjoint paths D_z (paths with no arcs in common) has to be found, where D_z is an (r_z, s_z) path (Li et al. 1992), such that the total costs of the paths are minimized. The restricted evacuation problem is a general version of the ADP problem, namely an ADP problem to which evacuees and departure times are added. In the restricted evacuation problem (Fig. 3b), a number of evacuees w (between 1 and n) depart from origin r_z on departure time k (between 1 and n). The network $G(V, A)$ consists of a set of nodes V and a set of directed arcs A , for which holds that $|A| = n$. The travel time c on each arc (expressed as a function of the number of evacuees w) is 1 if the arc is used by only one evacuee and $2n + 1$ if there are more evacuees on the link: $c(1) = 1$ and $c(w) = 2n + 1, w \geq 2$. If the paths are arc-disjoint, the needed evacuation time is $T \leq 2n$; namely, the last possible departure

time is n and the maximum travel time is equal to $\sum_1^n c(1) = n$. If the paths are not arc-disjoint, the needed evacuation time is equal to $T \geq 2n + 1$, the travel time of an arc used by more evacuees. The optimal solution for the restricted evacuation problem is thus to instruct arc-disjoint paths to the evacuees. Thus, the restricted evacuation problem is a general version of the ADP problem and the evacuation problem dealt with in this research is NP hard.

3 Optimization method

Because of the NP-hardness of the evacuation problem as described in the previous section, the exact solution cannot be found in an efficient way. To solve the evacuation problem, an optimization method is developed consisting of two steps: (1) the generation of the search space and (2) the algorithm. The generated *search space* consists of so-called *elements* $u \in U$, where each element is a combination of a departure time and a route (indirectly implying an origin and a destination). After the search space is generated, this search space is used in all iterations of the algorithm. In each iteration of the algorithm, evacuation instructions are constructed by assigning each group of evacuees to an element of the search space.

The (optimization) algorithm leads iteratively to effective evacuation instructions, most probably not the optimal instructions but instructions whose performance approaches optimality. The key components of the optimization approach are the algorithm itself, a traffic simulation model and an objective function (for example related to the number of arrivals over time), see the illustration of the approach in Fig 1. In each iteration of the approach, the evacuation as consequence of each instruction set created by the algorithm is simulated using a traffic simulation model, resulting in information about the traffic flows. Based on this traffic flow information, the performance of an instruction set follows from an objective function. These performance values are input for the optimization algorithm.

As mentioned in the introduction, the choices for the simulation model and the objective function are not fixed. A requirement for the simulation model is that the model can deal with departure time, destination, and route instructions, both under full and partial compliance. The simulation model and the objective function are restricted by each other: the output of the simulation model has to contain the desired input of the objective function.

3.1 Generation of the search space

The search space consists of a selection of all departure times and routes. Although in Sects. 3.1.1–3.1.3 the details of the generation of this search space will be given, this generation is first illustrated by an example to explain the basic ideas of the generation. This example is given in Fig. 4: given the combinations of all departure time and routes for a specific evacuation (a), the search space is reduced by selecting part of the departure times (b), selecting part of the routes (c) and selecting only the *feasible combinations* of the remaining combinations of departure times and routes (d). A selection of departure times and routes is made to reduce the computational complexity of

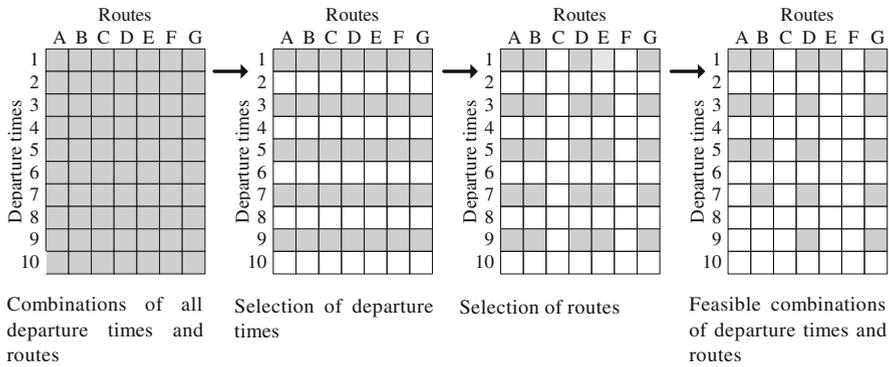


Fig. 4 Generation of the search space: all *grey blocks* together are the search space, each *grey block* separately is an element: a combination of a departure time and a route

the optimization method and to contribute to the feasibility of the resulting evacuation instructions. Another reason to select some of the routes is to avoid assigning of groups to unlikely routes (e.g., routes with relatively high free flow travel times). A so-called feasible combination (selected in Fig 4d) is a combination of a departure time and a route for which holds that it is possible to reach the destination when departing at the departure time and following the route under free flow conditions, given the network degeneration caused by the hazard.

Although the basic ideas of the generation are straightforward, the rules to generate the search space are more complicated. This complexity is primarily in the selection of routes out of all possible routes (on a real network, the number of possible routes is enormous) and in the determination of the feasibility of a combination of departure time and route (depending on the time period the hazard strikes the origin and the links on the route). In the rest of this section, these rules are explained. For the reader who is not specifically interested in the details of this generation: knowledge of the explanation (Sects. 3.1.1–3.1.3) is not needed for the understanding of the algorithm and the rest of the paper.

3.1.1 The two phases of the generation of the search space

Starting with an empty search space, the search space is generated in two phases, see Fig. 5. This division is based on the earlier mentioned network degeneration leading to the fact that routes appropriate for early departure times may not be appropriate for later departure times. In the first phase, elements are generated by constructing a route set for a totally accessible network and combining this with departure times for which it is possible to reach a destination following each route in the route set. In the second phase, the route set is adapted based on the network degeneration and combined with next departure time(s) which result in new elements. This adaptation is continued until it is no longer possible to reach any of the destinations for the next departure time. The first phase is described in Sect. 3.1.2, the second one in Sect. 3.1.3.

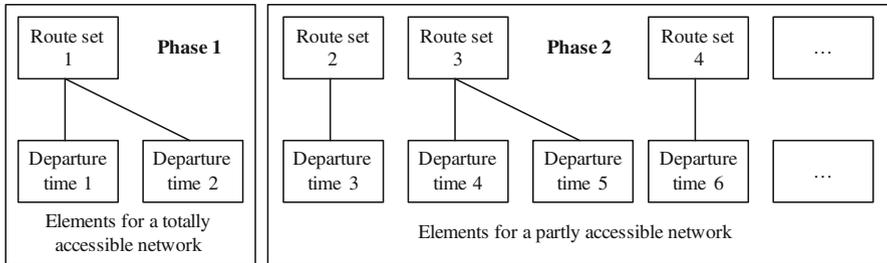


Fig. 5 The two phases of the generation of the search space

3.1.2 Phase 1: generation of elements for a totally accessible network

The generation of elements for a totally accessible network consists of the following steps:

1. Creation of candidate route sets based on network.
2. Creation of selection route sets based on overlap in routes.
3. Division of selection route sets in route set P and back-up routes.
4. Selection of departure times.

In step 1, candidate routes are generated for each origin, these are routes having relatively short free flow travel times. These routes are determined by repeatedly applying Dijkstra’s algorithm (Dijkstra 1959), where the travel times on the links are iteratively changed using Monte Carlo simulations (see Bliemer and Taale 2006). Each generated route is compared with the shortest free flow route for the origin: if the ratio of the travel time of the generated route and the travel time of the fastest free flow route is less than a given factor ξ_1 , the generated route is added to the candidate route set.

In step 2, selection route sets are created, which are subsets of the candidate route sets, based on the overlap between the routes in the candidate route sets. One by one the fastest free flow route is removed from the candidate set and the overlap between this route p^* and each route p in the (in the beginning empty) selection route set is computed:

$$\theta_{pp^*}^{route} = \frac{\sum_{a \in A_{pp^*}} tt_a^{free}}{tt_{p^*}^{free}} \tag{2}$$

where A_{pp^*} is the set of links included both in p^* and p , tt_a^{free} is the free flow travel time on link a , and $tt_{p^*}^{free}$ is the free flow travel time of p^* . If none of the overlaps for p^* is higher than the maximum allowed overlap $\theta_{p^*}^{route, max}$, p^* is added to the selection route set.

In step 3, a number of routes equal to the maximal number of routes per origin ψ_{routes}^{max} (or all the routes, when there are less then ψ_{routes}^{max} routes in the selection set) out of each selection set is added to the route set P . These are the routes having the shortest free flow travel times. All the routes in the selection sets that are not added to P are collected as back-up routes. These back-up routes might be used in the adaptation of P in the generation of elements for a partly accessible network.

All departure times which will be instructed to the evacuees are multiples of the fixed value k^* . The departure times $k \in K$ appropriate in combination with P , selected in step 4, are all multiples of the value k^* , from 0 up to and including the last departure time for which P is appropriate, k^P . This last term is equal to $\min\{k^r, k^P\} \forall r \in R^P$, $p \in P$, where R^P is the set of origins for which routes are included in P . The last departure time belonging to an origin, k^r , and the last departure time belonging to a route, k^P , are also multiples of k^* . They are equal to respectively the last departure time at which it is possible to leave the origin, and the last departure time at which it is possible to follow the route, both without being impeded by the hazard under free flow conditions. The time needed to load a group of evacuees on the network (depending on the capacity and the number of evacuees per vehicle) is taken into account. The elements, all combinations of departure times $k \in K$ and routes $p \in P$, are collected in U .

3.1.3 Phase 2: generation of elements for a partly accessible network

The elements for a partly accessible network are generated by repeating the following steps:

5. Collection of routes causing the inappropriateness of P .
6. Removal of routes from P belonging to inaccessible origins.
7. Replacement of routes in P with back-up routes.
8. Removal of routes from P belonging to origins from which destinations cannot be reached.
9. Replacement of routes in P with newly generated routes.
10. Selection of departure times.

Several adapted route sets are generated, each corresponding to a set of consecutive departure times, as also shown in Fig. 5. When it is no longer possible to leave an origin and reach a destination for the next departure time, the creation of elements stops immediately.

In step 5, the routes and origins causing the inappropriateness of P for the next departure time, \overline{P} and \overline{R} , are collected. These are the origins and routes for which respectively holds that k^r and k^P are equal to k^P , thus the last departure time at which it is possible to respectively leave the origin, and follow the route, without being impeded by the hazard under free flow conditions is equal to the last departure time belonging to the last created route set. Steps 6 up to and including 9 are followed until the sets \overline{P} and \overline{R} are empty. When they become empty, the departure times are selected in step 10.

If there are origins causing the inappropriateness because they are inaccessible ($\overline{R} \neq \emptyset$), routes belonging to the origins \overline{R} are removed from P in step 6 and when these routes are part of \overline{P} , also from \overline{P} .

If the set of back-up routes is not empty, unavailable back-up routes are removed from the back-up routes in step 7, where the availability is based on the last departure time belonging to a route k^P . When there are back-up routes belonging to the same origin as routes in \overline{P} , the last mentioned routes in \overline{P} are removed and the corresponding routes in P are replaced by as many as possible appropriate back-up routes (the shortest

back-up route first). The used back-up routes are eliminated from the set of back-up routes.

In step 8, the possibility to reach any of the destinations from all origins where routes in \overline{P} belong to is checked by registering all the accessible downstream links reachable from the aforementioned origins. If this is impossible for an origin, routes in P and \overline{P} corresponding to the origin are removed. This step differs from step 6: in step 6, it is about the accessibility of the origin itself, while in step 8, it is about the reachability of one of the destinations from the origin.

In step 9, new routes are generated to replace routes \overline{P} in P in the same way as routes are generated in steps 1 and 2, using a reduced network: only accessible links at departure time $\max\{K\} + k^*$ are included. For the overlap of the routes, both routes found in this step and routes in P are compared.

In step 10, departure times K are selected belonging to P . They are equal to all multiples of k^* from k^{earliest} up to and including k^P , where k^{earliest} is equal to $\max\{K\} + k^*$. All combinations of departure times $k \in K$ and routes $p \in P$ are new elements added to U .

3.2 Algorithm: EAS⁺ evacuation

The algorithm is based on ant colony optimization (ACO), a metaheuristic developed in [Dorigo and Stützle \(2004\)](#) based on the communication behavior of ants. This metaheuristic contains two phases: the *construction phase*, wherein solutions are constructed, and the *update of pheromone trails*, wherein information is updated based on the constructed solutions and their performance, to give direction to the search process. The algorithm presented in this section, *EAS⁺ evacuation (EAS⁺)*, is based on elitist ant system (EAS), one of the versions of the ACO metaheuristic described in [Dorigo and Stützle \(2004\)](#). The idea of EAS is to provide strong reinforcement to the parts of the *best-so-far* solution, the best solution found since the start of the algorithm. In EAS⁺, this reinforcement is even stronger.

EAS⁺ evacuation (see [Fig. 6](#)) is described in this section: the construction phase is described in [Sect. 3.2.1](#) and the update of pheromone trails in [Sect. 3.2.2](#). Although the complete algorithm is described in these sections, a pseudocode can be helpful for the implementation of the algorithm. The pseudocode of EAS⁺ evacuation is given in the [Appendix](#).

The application to the evacuation problem is not one of the usual applications of ACO algorithms. A well-known application is the application to the Traveling Salesman Problem (TSP). In this problem, cities and the distances between each pair of cities are given, and the objective is to find the shortest possible tour that visits each city exactly once. The reader who is interested in these differences can find them in [Sect. 3.2.3](#), wherein the differences between EAS⁺ evacuation and EAS applied to the TSP (EAS-TSP) are explicitly mentioned. However, no new information about EAS⁺ evacuation itself is given in [Sect. 3.2.3](#). Thus, the reader who is not interested in these differences can skip [Sect. 3.2.3](#) without missing any information about the functioning of EAS⁺ evacuation.

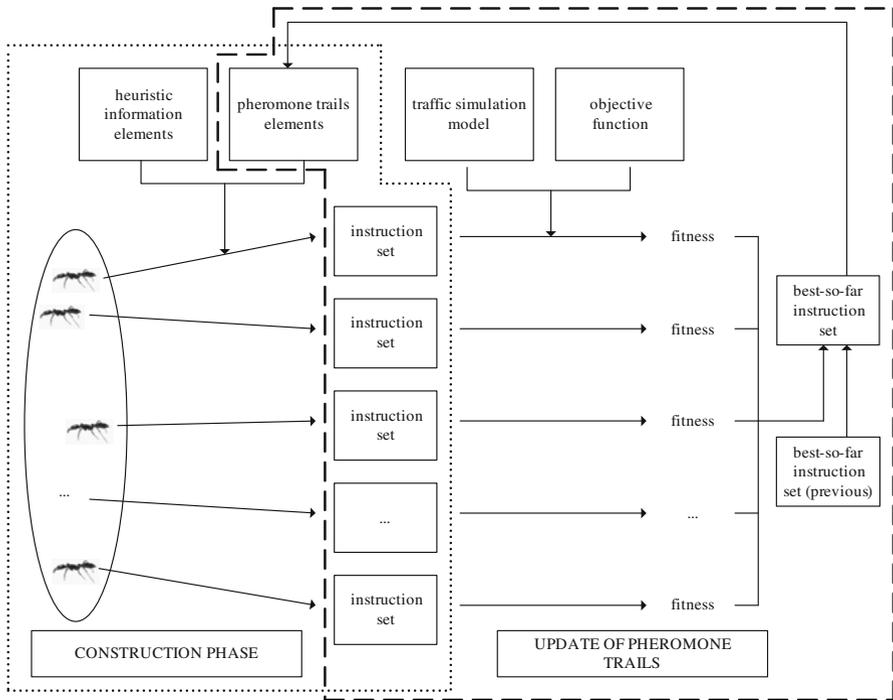


Fig. 6 EAS⁺ evacuation

3.2.1 Construction phase

In the construction phase, m ants concurrently build a solution, resulting in m solutions. In EAS⁺ evacuation, this solution is a set of evacuation instructions. This set consists of all groups of evacuees assigned to an element of the search space (a combination of departure time and route) belonging to the same origin as the group, whereby groups can be assigned to the same element.

The ants use a probabilistic action choice rule, called *random proportional rule*, to determine to which element each group is assigned. The probability with which an ant chooses to assign a group to element u is

$$h_u = \frac{[\tau_u][\eta_u]}{\sum_{u \in U_r} [\tau_u][\eta_u]}, 0 < h_u \leq 1, \tag{3}$$

where τ_u is the value of the so-called *pheromone trail* belonging to element u , η_u is a scalar for the so-called *heuristic information* for element u , and U_r is the set of elements belonging to origin r . By this probabilistic rule, the probability of choosing a particular element increases with the value of the associated pheromone trail and the value of the heuristic information. Heuristic information is constant for all iterations and gives elements which are expected to have a positive influence on the effectiveness of the evacuation instructions a relatively high selection probability. Given that

the objective function is not fixed in this optimization method, the heuristic information is also not fixed, but depends on the choice of the objective function. In Sect. 4, an objective function and heuristic information are proposed. Pheromone trails are not constant over the iterations: they give elements of good solutions of earlier iterations a relatively high probability. In the first iteration, the value for the pheromone trail is equal for all elements: $\tau_u = 1, u \in U$ in the first iteration. In Sect. 3.2.2, the determination of the pheromone trails in the next iterations will be explained.

The construction of the instruction sets can be parallel (at each construction step, all ants assign one group to an element) or sequential (an ant builds a complete instruction set before the next one starts to build a new one). Because the probabilities of (3) are updated in the next iteration and not during the construction steps of one iteration, parallel and sequential construction are equivalent: they do not influence the algorithm's behavior.

Each ant maintains a memory which contains the instruction set (groups assigned to the elements). When all ants have created their instruction set, this memory is used to determine the fitness value of each instruction set. This instruction set follows from a traffic simulation model and an objective function (the model and the objective function used within this paper are explained in Sect. 4). Based on the instruction sets and their fitness values, and the best-so-far instruction set determined in the previous iteration, the best-so-far instruction set to be used in the update of the pheromone trails is known.

3.2.2 Update of pheromone trails

After all ants have constructed their instruction sets, the pheromone trails are updated. In the first part of the update process, the pheromone value of each element is lowered by a constant factor. Then, pheromone is added on the elements that are part of the best-so-far instruction set E^{bs} . Pheromone evaporation is implemented by:

$$\tau_u \leftarrow (1 - \rho) \tau_u, u \in U, \quad (4)$$

where $0 < \rho \leq 1$ is the so-called *pheromone evaporation rate*. For τ_u holds the range: $\langle 0, \infty \rangle$. This evaporation is part of the algorithm to avoid unlimited accumulation of the pheromone trails and to enable the algorithm to “forget” bad instruction sets constructed in earlier iterations. If an element is not part of the best-so-far instruction set, its associated pheromone value decreases exponentially in the number of iterations. After evaporation, pheromone is deposited on the elements part of the best-so-far instruction set:

$$\tau_u \leftarrow \tau_u + \varepsilon \Delta \tau_u^{bs}, \quad (5)$$

where ε is a parameter that defines the weight given to the best-so-far instruction set, and thus indirectly influences the relative proportion of the exploration and concentration of the search process, and $\Delta \tau_u^{bs}$ is the amount of pheromone deposited on the

elements part of the best-so-far instruction set, defined as follows:

$$\Delta\tau_u^{bs} = \begin{cases} \frac{f_{E^{bs}}}{f_E^{\text{highest}}} \frac{\sum_{g \in G_u} B_{ru}^g}{B_r}, & \text{if } u \text{ belongs to } E^{bs}; \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where f_E^{highest} is the highest possible value for the fitness, and G_u is the set of groups assigned to u . For $\Delta\tau_u^{bs}$ holds: $0 \leq \Delta\tau_u^{bs} < 1$. The left fraction in (6) makes $\Delta\tau_u^{bs}$ depending on the effectiveness of the *best-so-far* instruction set scaled between 0 and 1. The better the best-so-far solution is the more pheromone the elements belonging to this instruction set receive. The right fraction of (6) is added to let $\Delta\tau_u^{bs}$ depend on the number of evacuees assigned to an element relative to the total population of the origin. The bigger the part of the residents of an origin assigned to a certain element is the more pheromone this element receive. The more pheromone an element receives, the more likely it is that this element will be chosen by ants to be part of an instruction set in future iterations of the algorithm.

Appropriate values of the parameters ρ in (4) and ε in (5) depend on the scale of the problem. When the number of possible evacuations instructions is higher (there are more departure times, routes and/or groups), a higher exploration and thus relatively low values for ρ and ε are useful. An appropriate value of ε also depends on the time available to find evacuation instructions: when the danger is critical, the value has to be relatively high to quickly find a solution. As mentioned by [Dorigo and Stützle \(2004\)](#), also the proportion between the initial values of the pheromone trails and the amount of pheromone with which the pheromone is updated influences the search process. The influence of the parameter settings is analyzed in the case study (Sect. 5).

3.2.3 Differences between EAS⁺ evacuation and EAS-TSP

All differences between EAS⁺ evacuation and EAS-TSP (as described by [Dorigo and Stützle 2004](#)) are explained. One of these differences changes the fundamentals of EAS; namely, in EAS all ants deposit pheromone on the elements of their solution, and pheromone is deposited on the elements of the best-so-far solution. In EAS⁺, pheromone is *only* deposited on the elements of the best-so-far solution, to increase the influence of this solution and to avoid influence of bad solutions constructed by the ants. The reason for this change was that tests showed that EAS⁺ leads to better results for the evacuation problem than EAS. Relative to other ACO algorithms, EAS gives the elements of the best-so-far solution a high influence. Because this influence of the best-so-far solution is made stronger in the new algorithm, the name of the algorithm is changed from EAS to EAS⁺.

All other differences are the consequence of the application to the evacuation problem instead of the TSP. The first difference is in the *solutions* that are created by the ants. In EAS-TSP, a solution is *a tour that takes the traveler through a given set of cities and then back home*, while in EAS⁺ evacuation, a solution is *a set of instructions*. The tour consists of arcs, while the set of instructions consists of numbers of people assigned to elements (combinations of departure times and routes). Thus, in

EAS⁺ evacuation, elements are either part of the solution, or not (like the arcs are in EAS-TSP), but for the elements that are part of the solution, there is also a number of people that is assigned to the element. This change in solution automatically leads to a change in the *memory*: in EAS⁺ evacuation, the memory of each ant contains the instruction set, while in EAS-TSP, the memory of each ant contains the visited cities in the order they were visited.

The second difference is in the *use of the random proportional rule* (3). One construction phase consists of multiple steps, whereby in each step one element is selected to be part of the instruction set of one of the ants. In EAS-TSP, (3) *changes during one construction phase*, since possible arcs to be selected for the rest of the tour and their probabilities depend on the arcs that are already part of the tour. In EAS⁺ evacuation, (3) *does not change during the construction phase*, since possible elements to be selected for one group and their probabilities do not depend on the elements that are already selected for other groups.

Also the *amount of pheromone deposited on the elements of the best-so-far solution* is different for EAS-TSP and EAS⁺ evacuation, being the third difference. In both algorithms, this amount is a function of the solution quality. Given that the applications (TSP and evacuation) are completely different, also the quality expressions are different (see (6) for the solution quality in EAS⁺ evacuation).

The final difference is in the *parameters in the random proportional rule*. The rule as formulated by Dorigo and Stützle (2004) contains parameters α and β , determining respectively the relative influence of the pheromone trail and the heuristic information. With these parameters, (3) changes in:

$$h_u = \frac{[\tau_u]^\alpha [\eta_u]^\beta}{\sum_{u \in U_r} [\tau_u]^\alpha [\eta_u]^\beta}, \quad 0 < h_u \leq 1. \quad (7)$$

In EAS⁺ evacuation, these parameters are removed (having the same effect as fixing these parameters to the value 1). This is done because in EAS⁺ evacuation, the relative influence of the pheromone trail is also influenced by the parameter ε in (5), and the relative influence of the heuristic information is also determined by the definition of the heuristic information (this definition is not fixed in EAS⁺ evacuation). We prefer to avoid having multiple parameters describing the same effect, therefore parameters α and β are not included in the random proportional rule.

4 Specification

As mentioned in Sect. 3, an objective function and a traffic simulation model are used in the optimization method to determine the effectiveness of a set of evacuation instructions. They are both to the choice of the analyst, with the requirement that evacuation instructions can be given as input to the simulation model. In this section, the objective function and simulation model are presented which are used in this paper. Based on this objective function, the structure of the heuristic information is developed and therefore also explained in this section. We also recommend the objective function,

heuristic information and simulation model for other applications of the optimization method because of their appropriateness, as described in this section.

4.1 Specification of the objective function

To compare the performance (fitness) of different evacuation instructions, an optimization objective is needed. An objective is applied dealing with uncertainties in a hazard that may strike a region, i.e., uncertainties in the time period the hazard strikes the region and the location of the hazard (see Franklin 2008). This is contrast to optimization objectives proposed in other studies that do not deal with these uncertainties (assuming that all people can be evacuated in time or that the available evacuation time is exactly known prior to an evacuation).

The optimization objective is to maximize a function of the number of arrived evacuees for each time period, where early arrived evacuees can be appreciated higher than evacuees that arrived later. As there is a risk of not being able to evacuate everyone, it is preferred to evacuate people earlier than later. This is represented in the fitness of evacuation instructions, named the effectiveness of the set of evacuation instructions:

$$f_E = \sum_{t>0} \exp(-\beta t) q_E(t) \quad (8)$$

where $q_E(t)$ is the number of evacuees reaching a safe destination in time period t , depending on evacuation instructions E , and β is a weighting parameter with $\beta \geq 0$. This parameter makes the function generic: when $\beta = 0$, the optimization objective is equal to maximizing the arrived evacuees. When the value of β is higher, the importance of early arrivals is increased. Because β is multiplied with t , the appropriateness of the value for β is influenced by the total evacuation time. Therefore, by determining β , a global estimation of this time can be helpful. An appropriate estimation is the duration of an evacuation without instructions (a voluntary evacuation). This duration can be estimated using a simulation model like the model presented in the Sect. 4.3. Given the definition of the fitness in (8), the number of arrivals in all time periods of the evacuation has to be output of the simulation model. The fitness is maximized in the evacuation problem as given in (1).

In (6), the highest possible value for the fitness is used. For this objective function, f_E^{highest} is defined as the population B . When all people would arrive at a safe destination at the start of the evacuation, the fitness value would equal the population. This course of the evacuation is impossible in practice, but can function as an upper bound for the fitness value.

4.2 Specification of the heuristic information

The heuristic information is used to give elements which are expected to have a positive influence on the effectiveness of the evacuation instructions a relatively high selection probability. Going with the objective presented in the previous section, these are elements leading to arrivals whereby early arrived evacuees can be higher appreciated than later arrived evacuees. The heuristic information is based on the information

about the departure times and routes belonging to the elements. For the scalar of this information for each element u , κ_u , holds:

$$\kappa_u = \vartheta_{p_u} \zeta_{k_u} \tag{9}$$

where ϑ_{p_u} is a scalar representing information about the free flow travel time of route p belonging to element u , the highest possible value and ζ_{k_u} is a scalar representing information about departure time k belonging to element u . For κ_u holds $0 < \kappa_u \leq 1$. A relatively high value of κ_u gives element u a relatively high selection probability. The boundaries for κ_u follow on the boundaries of both information parts described hereafter.

The information about the free flow travel time of route p belonging to element u gives an element with a relative low free flow travel time a relative high value for the heuristic information. Therefore holds:

$$\vartheta_{p_u} = \left(\frac{\min_{u \in U_r} \{ \tau_{p_u}^{\text{free}} \}}{\tau_{p_u}^{\text{free}}} \right)^{\xi_2} \tag{10}$$

where ξ_2 is a weighting parameter with $0 \leq \xi_2 \leq 2$. When $\xi_2 = 0$, the heuristic information does not depend on the travel times of the routes. The higher the value for ξ_2 is, the larger the distinction in the heuristic information based on the differences in travel times is. The maximum value of ξ_2 avoid the situation wherein the selection probability of an element becomes negligible. As a consequence of the boundaries for ξ_2 and the factor ξ_1 , it holds that $1/\xi_1 \leq \vartheta_{p_u} \leq 1$.

The information about departure time k belonging to element u gives an element with a relative early departure time a relative high value for the heuristic information. Therefore, ζ_{k_u} is determined as follows:

$$\zeta_{k_u} = 1 - \xi_3 \frac{k_u}{\max K_{r_u}} \tag{11}$$

where K_{r_u} is the set of departure times for which there are elements belonging to r_u , the origin r belonging to element u . For ζ_{k_u} holds $0 < \zeta_{k_u} \leq 1$. For weighting parameter ξ_3 holds that $0 \leq \xi_3 < 1$. When $\xi_3 = 0$, the heuristic information does not depend on the departure times. The higher the value for ξ_3 is, the larger the distinction in the heuristic information based on the differences in departure times is.

This interpretation of the heuristic information is suitable for an optimization with the objective function presented in the previous section, but also with other objective functions applied to the evacuation problem. For example, this heuristic information can be used when maximizing the arrivals or minimizing the evacuation time. There may be objective functions for which other interpretations of the heuristic information fit better, like minimizing the distance traveled by the evacuees. However, for all objective functions which are most usually applied to the evacuation problem, related to the number of arrivals over time, the heuristic information as presented in this section is appropriate.

4.3 Specification of the evacuation simulation model

The traffic flows resulting from a set of evacuation instructions are simulated using the evacuation simulation model EVAQ developed by Pel et al. (2008). This model contains dynamic travel demand (evacuation and departure time choice), en-route travel choice behavior (destination and route choice) and dynamic network loading (propagation of the traffic flows through the network). The network degeneration caused by a hazard is included in the model. The number of arrivals in all time periods of the evacuation is part of the output of EVAQ, and therefore it is possible to determine the effectiveness of the instructions using (8). The effects of network degeneration caused by the hazard are also included in EVAQ.

The model EVAQ is able to analyze both mandatory (full compliance with the instructions) and recommended (partial compliance) evacuations. In case of partial compliance, the resulting departure time, route, and destination for an evacuee are the result of the instructions for the evacuee, the departure time, route, and destination preferred by the evacuee, and a compliance factor. Thus, by using EVAQ as simulation model in the optimization approach presented in this paper, the optimization of both mandatory and recommended evacuation instructions is possible. This is an improvement compared to earlier developed optimization methods for evacuation instructions, wherein full compliance was assumed.

In reality, compliance of the evacuees with the instructions will be more complicated than is described in EVAQ; namely, evacuees will most probably make their decisions based not only on their own preferences and the instructions, but also on the reactions of other evacuees. Traffic related effects are included in EVAQ: for example, if congestion arises at a certain road, the probability that an evacuee chooses that road declines, since the expected travel time becomes higher. However, it could also happen that people will drive on a road because other people are driving on that road, or that people will depart at the moment their neighbors depart. Last mentioned effects are not modeled by EVAQ or one of the other existing evacuation traffic simulation models. Thus, with the existing evacuation traffic flow models it is impossible to include these effects. If a traffic simulation model including these effects will be developed, these effects can also be included in the optimization of the instructions using that simulation model in the application of the optimization method presented in this paper.

5 Case study

In this section, the optimization method is applied to a hypothetical flooding of part of the Netherlands and the results are analyzed. The section starts with a description of the flood scenario (Sect. 5.1), after which a comprehensive analysis follows in Sects. 5.2–5.7. To understand these analysis sections, it is not needed to read all of them: the reader can choose to read the sections of his or her interest. To help the reader in selecting the sections, an overview of the analysis is given:

- Section 5.2 shows the convergence to an optimized instruction set and discusses the computational time.

- In Sect. 5.3, the arrivals over time during the evacuation are compared for the situation wherein the optimized instructions are applied, and a situation without instructions.
- In Sect. 5.4, the arrivals over time during the evacuation are compared for the situation wherein the optimized instructions are applied, and a situation wherein instructions set up by simple decision rules are applied.
- The departure times, routes, and destinations the evacuees are assigned to in the optimized instructions are analyzed in Sect. 5.5.
- In Sect. 5.6, the use of the network when applying the optimized instructions is compared with the optimal use of the network to analyze the “deviation from optimality” of the optimized instructions.
- In Sect. 5.7, the influence of the parameter settings is analyzed.

The objectives of this analysis are to show the performance of the method (Sect. 5.2), the need to give instructions to the evacuees (Sect. 5.3), the need to optimize these instructions (Sect. 5.4), patterns in the optimized departure time, route, and destination instructions (Sect. 5.6), the near-optimality of the optimized instructions (Sect. 5.5), and the influence of the parameter settings (Sect. 5.7).

5.1 Flood scenario

The optimization method is applied to a hypothetical flooding of Walcheren (see Fig. 7), a peninsula in the southwest of the Netherlands. Over 120,000 residents from an area of 216 squared kilometers have to be evacuated, whereby the number of evacuees per vehicle is assumed to be equal to 2.5 (reflecting the average number of evacuees per vehicle). Walcheren is flooded in 4 h. The evacuation is assumed to start 2 h before the flood, by which the time available to evacuate is equal to 3 h. Figure 7 shows which part of the network is flooded after 2, 3, 4, 5, and 6 h.

For this flood scenario, evacuation instructions are optimized with the method presented in this paper. For this application, full compliance of the evacuees with the instructions is assumed. The reason for this is that the primary objective of this section is to discuss the functioning of the method (in terms of results and efficiency). By doing this under the assumption of full compliance, the functioning can more easily be compared with the functioning of other methods (the methods introduced in Sect. 1 do all assume full compliance). However, the method can also be applied under assumption of partial compliance. For examples of applications of the method presented in this paper under assumption of partial compliance (see Pel et al. 2009).

The parameter settings for the application of the optimization method are given in Table 1. These settings have been found to perform well for several case studies. In Sect. 5.7, the influence of the parameter settings is analyzed for this specific case study. For the objective function, the importance of early arrivals is assumed to be of such a level that a suitable value for the parameter for the objective function, β , is 0.1. This means that evacuees arriving at their destination after 6 h of evacuation have a weight of 55 compared with 100% for evacuees arriving at their destination at the start of the evacuation.

The application of the optimization method to the hypothetical flooding resulted in an optimized instruction set.

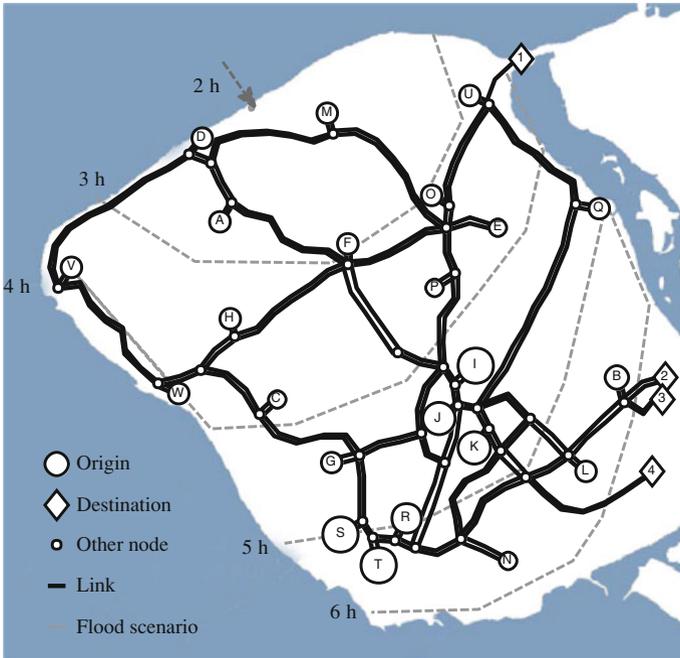


Fig. 7 Flood scenario for Walcheren. The network consists of origins (A–W), destinations (1–4), other nodes and 158 directed links. The size of the origin indicates the population size, varying between 500 and 17,000

Table 1 Values for the parameters in the optimization method

Symbol	Explanation	Value
Generation of elements		
k^*	Period between two following departure times which can be instructed	0.5 h
ψ_{routes}^{max}	Maximum number of routes combined with a departure time	5
ξ_1	Factor setting the maximum allowed free flow travel time relatively to the shortest travel time	2
$B^{constant}$	Group size	10,000
$\theta_{p^*}^{route,max}$	Maximum overlap factor	0.8
Algorithm		
ξ_2	Weighting parameter to influence the dependency of the heuristic information on the travel times of the routes	1
ξ_3	Weighting parameter to influence the dependency of the heuristic information on the departure times	0.5
$ m $	Size ant colony	10
ρ	Pheromone evaporation rate	0.02
ε	Parameter to influence exploration and concentration of the search process	0.1

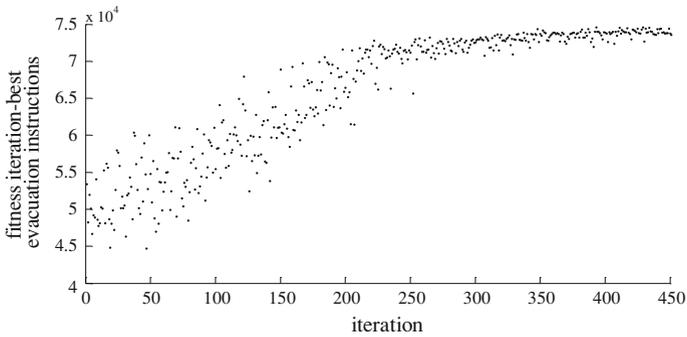


Fig. 8 Effectiveness of the iteration-best evacuation instructions

5.2 Performance of the optimization method

The effectiveness (f_E) of the optimized instruction set is equal to 74,636. Figure 8 shows the convergence towards this optimized instruction set. Exploration is visible in the first iterations (the differences between the effectiveness of the iteration-best evacuation instructions are relatively large) and concentration in the last iterations (the differences are relatively small).

The optimization method is programmed in Matlab and the application took 12-h computation time using a desktop computer with an Intel Pentium Core 2 Duo @ 2.5 Ghz and 2GB RAM. This computation time can be drastically lowered by parallel programming (now the optimization method is programmed serially) and using fast computers, which makes the computation time relatively short for real scale scenarios. The most computationally intensive part is running the traffic simulation model.

5.3 An evacuation with optimized instructions compared to an evacuation without instructions

In this section, the effectiveness of the optimized instructions is compared to the efficiency of an evacuation without instructions (a voluntary evacuation). The optimized instructions are both applied under full compliance (a mandatory evacuation), and under partial compliance (recommended evacuations). The effectiveness of the instructions follows from (8) (using EVAQ), with the optimized instruction set as instruction set E . The efficiency of the voluntary evacuation follows from the same equation, however for this evacuation the instruction set E is an empty set.

The efficiency of the different evacuations is equal to, approximately:

- 75,000 (mandatory evacuation);
- 63,000 – 69,000 (recommended evacuation);
- 46,000 – 56,000 (voluntary evacuation).

For the recommended and voluntary evacuations, bandwidths of the performance values are given. The first reason for these bandwidths is that several runs were needed for a voluntary and recommended evacuation because of a stochastic route choice

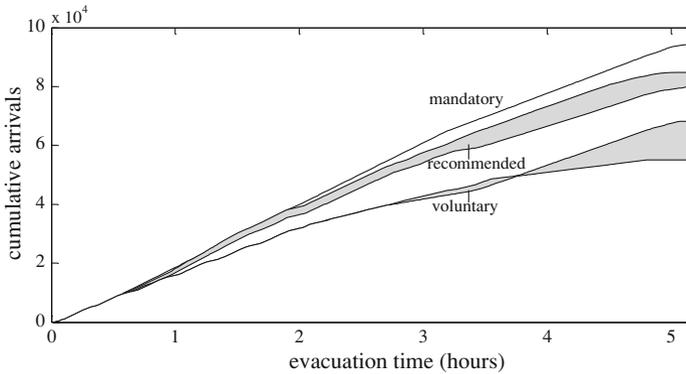


Fig. 9 Arrival patterns of voluntary, recommended and mandatory evacuations, applying the optimized evacuation instructions

generation process included in EVAQ for these evacuation types. Second, for the recommended evacuation, different degrees of control of the evacuation are tested.

In addition to the efficiency values, the arrival patterns for the different evacuations are given in Fig. 9. The results show that evacuations with instructions perform better (both in the efficiency value determined by (8) and in the cumulative arrivals) than a voluntary evacuation, and that the performance of mandatory instructions is higher than the performance of recommended instructions. To conclude, instructions are needed to raise the efficiency of an evacuation.

5.4 An evacuation with optimized instructions compared with an evacuation with instructions created by simple rules

In this section, the effectiveness of the optimized evacuation instructions is compared to the effectiveness of evacuation instructions set up by a set of simple (but possibly naïve) evacuation rules. These rules are (1) the nearest destination is instructed, (2) the fastest free-flow route is instructed, and (3) the departure times are instructed in such a way that no congestion occurs, where evacuees whose origin will be flooded first are evacuated first. These kind of rules are used in practice, and are also presented in other optimization methods (see e.g. Saadatseresht et al. (2009), wherein the shortest routes are chosen to lead evacuees from their origins to their destinations).

The effectiveness of the different evacuations (both under assumption of full compliance is equal to, approximately:

- 75,000 (optimized instructions);
- 37,000 (instructions created by simple rules).

In addition to the effectiveness values, the arrival patterns for the evacuations under both kinds of instructions are given in Fig. 10. The results show that the optimized instructions perform better (both in the effectiveness value determined by (8) and in the cumulative arrivals) than the instructions created by simple rules. Thus, application of an optimization method like the method presented in this paper seems to be

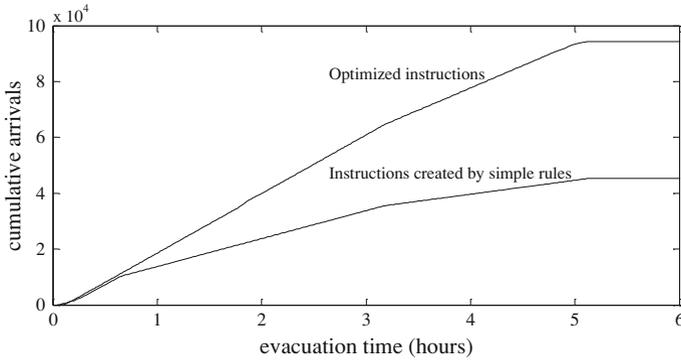


Fig. 10 Arrival pattern of the evacuation by applying the optimized instructions and the instructions created by a set of simple rules

necessary to create evacuation instructions, and using simple rules to create evacuation instructions seems to be inappropriate.

The comparison between the effectiveness of the instructions created by simple rules (equal to 37,000) and the efficiency of an evacuation without instructions (equal to 46,000–56,000, adopted from Sect. 5.3) gives an important insight: the performance of an evacuation without instructions is higher than the performance of an evacuation with instructions created by simple rules. Thus, when giving instructions to evacuees, it is important that they are properly created; otherwise it may even be better to give no instructions at all.

5.5 Analysis of the optimized instructions

In Sect. 5.4 is showed that optimized instructions perform better than instructions created by simple rules (evacuating to the nearest destination, following the fastest free-flow route, and instructing the departure times in such a way that no congestion occurs, where evacuees whose origin will be flooded first are evacuated first). Most likely, this difference in performance is caused by the relationship between the instructions in both sets. The results indicate that it is for example not effective to instruct all people to follow the shortest route. In this section, the optimized instructions are analyzed to see if there is a pattern in these departure time, route, and destination instructions explaining the relatively high performance.

Starting with the departure times, Fig. 11 shows the departure times against the moment that the hazard strikes the origin of the evacuees. There does not seem to be any relation between the departure times and the moment the hazard strikes the origin, except the logical observation that no evacuees are assigned to a departure time later than the moment the hazard strikes their origin. Regarding the routes, the percentage of people assigned to the shortest free flow routes is equal to 36%, the other evacuees are spread over the rest of the routes. The percentage of people assigned to the nearest destination (based on the shortest free flow route) is equal to 55%, the other evacuees are assigned to other destinations.

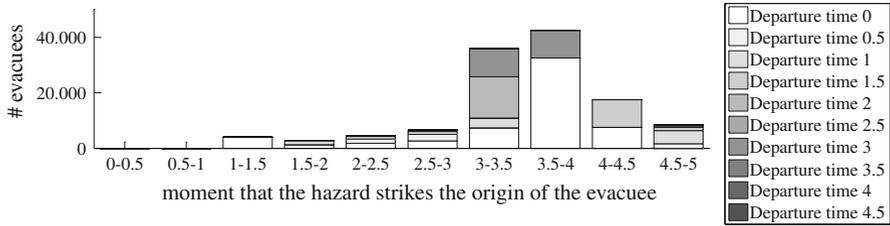


Fig. 11 Instructed departure time for each evacuee for the moment the origin of the evacuee is flooded

Table 2 Comparison between the instructions created by simple rules and the optimized instructions

	Instructions created by simple rules	Optimized instructions
Departure times	Evacuees whose origin will be flooded first depart the earliest (and the schedule is set up such that no congestion arises)	No relation between the departure time and the moment the origin is flooded
Routes	100% of the people instructed to follow the shortest free flow route	36% of the people instructed to follow the shortest free flow route
Destinations	100% of the people instructed to travel to the nearest destination	55% of the people instructed to travel to the nearest destination

There are both similarities and differences between the optimized instructions and the instructions created by simple rules, see Table 2 for an overview of this comparison. Apparently, it is efficient to instruct part of the people to follow the shortest route and to travel to the nearest destination, but it is not efficient to give these instructions to all people. This can be explained by the fact that part of the network is unused when giving these instructions to all the people. It seems to be more efficient to use the complete (or a bigger part of the) network by spreading the evacuees over more routes and destinations. By applying the simple rules, evacuees whose origin will be flooded first, are evacuated first. Such a relation between the departure time and the moment the origin is flooded is not recognized in the optimized instructions.

This analysis of the optimized instructions is adopted from Huibregtse et al. (2010), wherein more details can be found. In Huibregtse et al. (2010), similar relations for the departure time, destination and route instructions were found for other applications of the optimization method.

To conclude, it seems to be efficient to spread the people over the routes and destinations in the network, whereby *part* of the people are instructed to follow the shortest route and to travel to the nearest destination. Optimized instructions appear to have a complex structure, which cannot be captured in simple decision rules.

5.6 Near-optimality of the performance of the instructions

An evacuation with optimized instructions performs better than an evacuation without instructions (result of Section 5.3) and the effectiveness of optimized instructions is higher than the effectiveness of instructions created by simple rules (result

of Section 5.4). However, so far it is unknown how close to optimal the optimized instructions are. This is analyzed in this section.

The best way to analyze the optimality of the optimized instructions is to compare their effectiveness with the effectiveness of the optimal instructions. Unfortunately, but logically, the optimal instructions are unknown (otherwise the optimization method presented in this paper would not be necessary). However, an upper bound on the performance of instructions is known: this upper bound is equal to the maximal arrival rate of the network.

This maximal arrival rate is equal to 21,000 evacuees per hour plus 4,790 evacuees over the whole evacuation. This rate is derived as follows. The maximal arrival rates to destination 1, 2, 3, and 4 (see Fig. 7) separately are respectively 5,000, 11,000, 5,000, and 5,000 evacuees per hour (based on the capacities of the links upstream of these 's during this period is equal to 67,790 evacuees (based on the maximal arrival rate). The number of arrivals during this period when applying the optimized instructions is equal to about 61,000. Thus, the number of arrivals during this period is equal to 90% of the maximum number of arrivals.

Figure 12 shows that during a substantial part of the evacuation period, the arrival rates at the destinations are equal to the maximal arrival rates. In the first moments of the evacuation no arrivals are possible since the evacuees have to travel from their origin to the destination. At the end of the evacuation, no arrivals are possible because of the network degeneration caused by the hazard. Other deviations from the maximal arrival rates (like the 'gap' in the arrivals at destination 4) could indicate sub-optimality of the instructions.

This analysis of the near-optimality is adopted from [Huibregtse et al. \(2010\)](#), wherein more details can be found. In [Huibregtse et al. \(2010\)](#), similar numbers were found for other applications of the optimization method.

To conclude, the near-optimality (expressed as the number of arrivals divided by the maximum number of arrivals) of the optimized instructions is equal to 90% during the analyzed period of the evacuation. This indicates a sub-optimality of the optimized instructions. This sub-optimality can be a logical consequence of the used simulation-optimization approach; it can also be caused by the group size. It can be more efficient to split a big group into smaller ones, to spread the evacuees over the network. At the same time, the instructions do also have to be communicated to the evacuees. From this point of view, it might be unfeasible to specify the instructions for small groups of evacuees. However, the near-optimality of the optimized instructions is not a prove of the sub-optimality of the instructions. The maximum number of arrivals is an upper bound on the performance of the optimal instructions as mentioned before. It could be that the number of arrivals when applying the optimal instructions is lower than the maximum number of arrivals (this depends on the specific network design, hazard pattern or demand distribution). Because the optimal instructions are unknown, this can not be analyzed in more details.

5.7 Influence parameters and consistency

In this section, the influence of the parameter settings on the efficiency of the method and the effectiveness of the resulting instruction set is analyzed. In addition, the

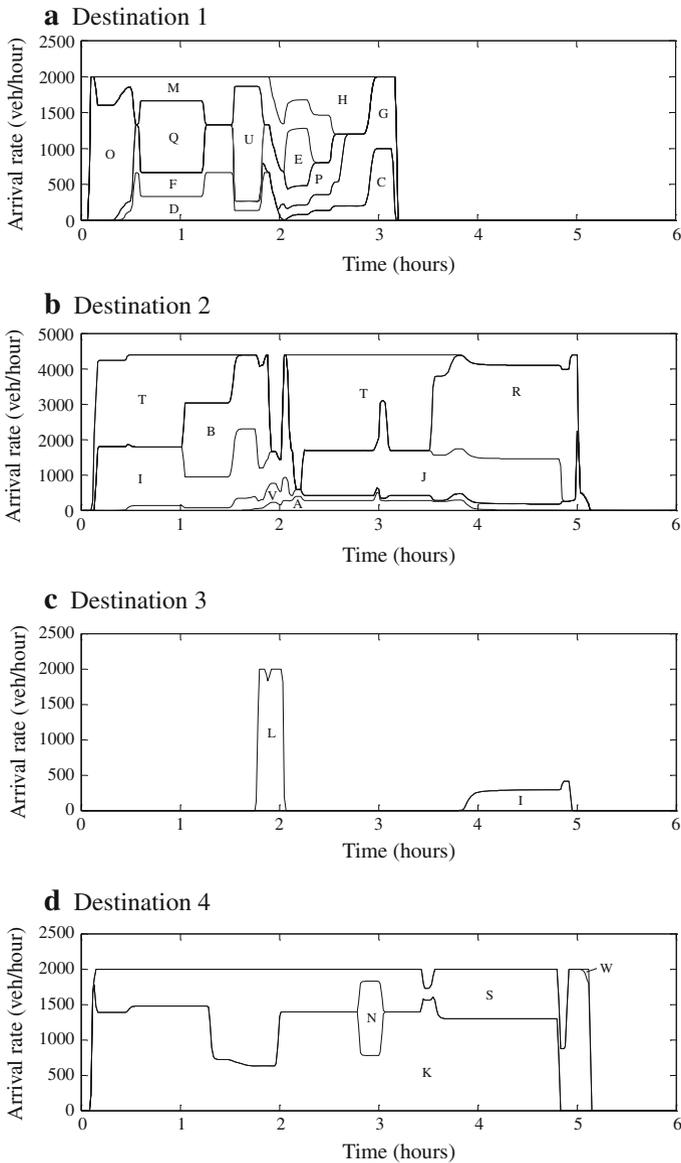


Fig. 12 Arrival rates at the destinations of the Walcheren network (the number of evacuees per vehicle is assumed to be equal to 2.5). The letters are the origins of the evacuees

consistency of the results is investigated by analyzing different runs with the same parameter settings.

The parameters that are analyzed in this section are the parameters in the algorithm of the optimization method, EAS⁺ evacuation. Optimal parameter settings can not be based on other applications, firstly because EAS⁺ evacuation is not applied before, and secondly because optimal parameter settings for ant colony optimization methods

Table 3 Values for the parameters in the optimization method

Symbol	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
ξ_2	1	0	2	1	1	1	1
ξ_3	0.5	0	0.999	0.5	0.5	0.5	0.5
$ \{m\} $	10	10	10	5	15	10	10
ρ	0.02	0.02	0.02	0.02	0.02	0.1	0.2
ε	0.1	0.1	0.1	0.1	0.1	1	10

Table 4 Effectiveness (fitness value) optimized evacuation instructions

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Run 1	74,636	72,167	72,750	71,764	71,709	73,368	70,574
Run 2	75,569	70,353	72,779	73,641	74,734	73,136	65,769
Run 3	74,965	74,976	73,026	74,119	75,176	73,922	66,674
Run 4	75,548	73,163	73,483	74,722	71,928	68,334	68,461

are problem specific, as showed by [Gaertner and Clark \(2005\)](#) and [Wong \(2008\)](#). The other part of the optimization method, the generation of the search space, does also contain parameters. However, these parameters influence the feasibility of the evacuation instructions and therefore appropriate values of these parameters depend on the desires of the authority (for example the maximum number of different instructions). Thus, parameters in the generation of the search space are not analyzed in this section.

The influence of the parameters is analyzed by applying the optimization method for different parameter settings. Because the influence is unknown, the following extreme situations are tested:

1. Same settings as applied before;
2. No heuristic information (no information about the departure times and the routes);
3. Relatively high values for the heuristic information;
4. Relatively low number of ants in the colony;
5. Relatively high number of ants in the colony;
6. Relatively small exploration;
7. Relatively small exploration (smaller than the exploration in test 6).

These situations are translated in the parameter settings showed in [Table 3](#). At the same time, the consistency is checked by applying the optimization method four times for each situation. [Table 4](#) shows the effectiveness of the optimized instructions for each run (all runs were stopped when there was convergence in the results).

The results show consistency in the effectiveness values for Test 1 and 3, and inconsistency for the other tests. Regarding the effectiveness values, relatively high values were found for Test 1, and relatively low values for Test 6 and 7. However, also the optimized instructions with low effectiveness values are valuable compared to a situation without optimized instructions: the effectiveness of all optimized instructions in this section is higher than the efficiency of an evacuation without instructions (between 46,000 and 56,000, adopted from [Sect. 5.3](#)) and higher than the effectiveness

of instructions created by simple rules (33,000, adopted from Sect. 5.4). Finally, the efficiency of the optimization method under the different parameter settings is compared based on the effectiveness of the iteration-best instructions as plotted in Fig. 13. The biggest difference in the number of iterations needed to let the results converge is the number of iterations needed by Test 1–5 on the one hand, and Test 6–7 on the other hand. Because the traffic simulation is computationally the most intensive part, the computational time for one iteration is approximately the same for the tests with same size of the ant colony (Tests 1–3 and 6–7). In contradiction, one iteration of Test 4 and 5 needs respective half and double the time of one iteration of the other tests. This means that Test 7 needs the lowest amount of computational time, followed by Test 4 and 6, and Test 5 is computationally the most expensive.

To conclude, different parameter settings lead to different results. However, also the optimized instructions with low effectiveness values are valuable because their effectiveness is higher than the efficiency of an evacuation without instructions and higher than the effectiveness of instructions created by simple rules. The settings do not only influence the effectiveness of the optimized instructions, they do also influence the efficiency of the method. This means that the authority developing the instructions has to choose the parameter settings based on both the desired effectiveness of the instructions and the available computational time.

6 Conclusions

In this paper, a method is presented to optimize evacuation instructions. The route, destination, and departure time instructions are simultaneously optimized. To determine the effectiveness of a set of evacuation instructions, an objective function and a traffic simulation model are used, which can be chosen by the analyst. Thus, advanced models and functions can be used. Therefore, it is possible to deal with the compliance of people with the instructions.

The method is applied to a hypothetical flood of part of the Netherlands, resulting in an optimized set of instructions for this situation. The case study shows that the efficiency of an evacuation whereby instructions are given (under the assumption of both full and partial compliance) is higher than the efficiency of an evacuation without instructions. Thus, instructions are needed. The case study also shows that the instructions have to be optimized, they can not be constructed by simple decision rules (like instructing the people to follow the shortest route, travel to the nearest destination and depart with such a schedule that the people whose origin will be flooded first are the first to evacuate (without congestion)).

Further analysis shows that the near-optimality (expressed as the number of arrivals divided by the maximum number of arrivals) of the optimized instructions is equal to 90%. This indicates a sub-optimality of the optimized instructions. However, the number of arrivals as consequence of the (unknown) optimal instructions is most probably lower than the maximum number of arrivals (based on the capacity of the network). Among other things, this can be caused by network degeneration caused by the hazard. Thus, the deviation between the number of arrivals under optimized and optimal instructions is 10% at maximum.

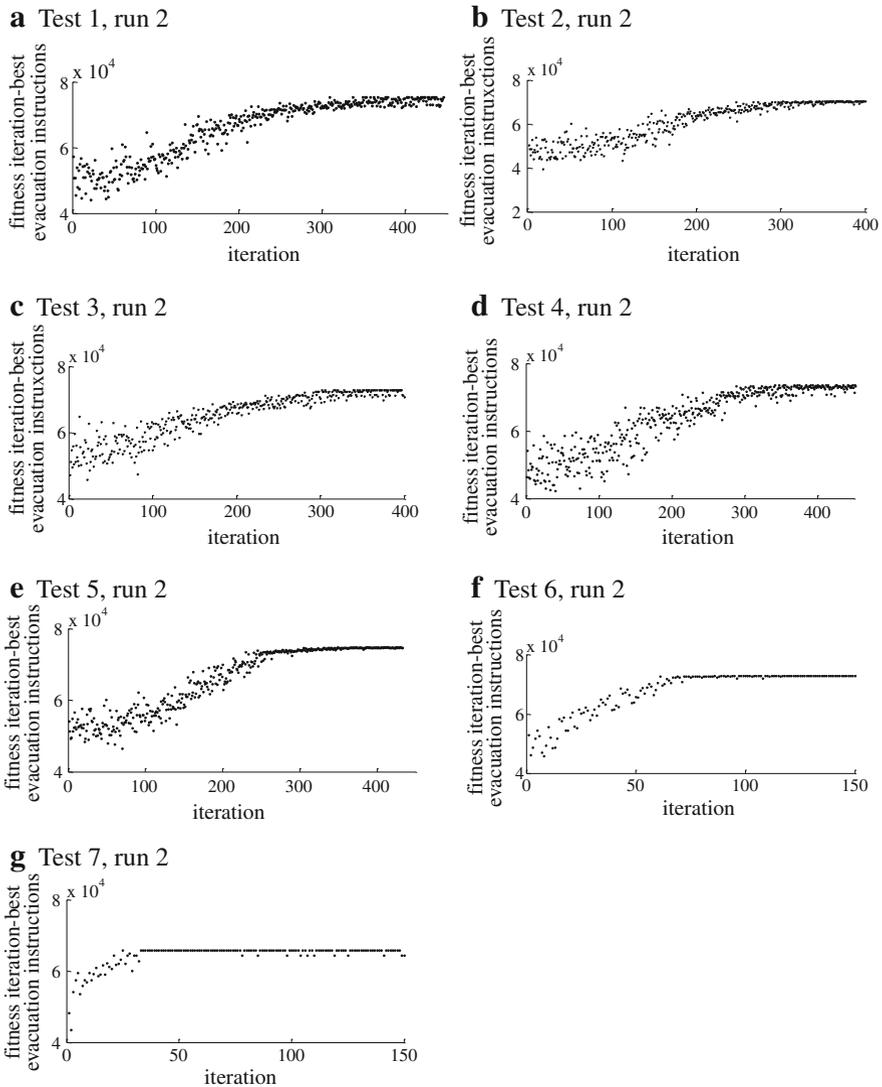


Fig. 13 Effectiveness of the iteration-best evacuation instructions for specific runs

The analysis of the parameter influences shows that different parameter settings lead to different values of both the effectiveness of the optimized instructions and the efficiency of the optimization method. Thus, the authority developing the instructions has to choose the parameter settings based on both the desired effectiveness of the instructions and the available computational time.

In the future, the development of an optimization method for evacuation instructions will go in two directions. The first direction is to deal with uncertainty. There are many uncertainties in the evacuation problem, related to the hazard (location, time,

intensity), the evacuees (behavior, demand) and the system (capacities), influencing the situation during the evacuation significantly. By developing robust instructions (appropriate for all situations which can arise as consequence of the uncertainty), or by developing instructions that can easily be adapted to the real scenario during the evacuation, the uncertainty in the evacuation problem will be taken into consideration.

The second direction is to extend the problem in such a way that if not all the people can be evacuated on time (as is the case in the application in this paper), an optimal decision will be made about which part of the evacuees has to be evacuated, and which part has to go to a safe area in the region. Because this safe area is located in the threatened region, an extra (transportation) problem arises; namely, how to supply the people with their first necessities of life. The reason to extend the problem in this way is that while the evacuation problem is solved in a more realistic way compared to how the problem is solved in earlier research (partial compliance of the evacuees to the instructions can be taken into account, and not only the question how fast the people can be evacuated, but also the question if they can be evacuated is dealt with), a new question arises. Namely, how will people behave if they cannot evacuate on time, and is it even acceptable from a social point of view to apply a plan that does not take care of all the people? For both future research directions holds that the method presented in this paper has to be extended to be able to solve the problems.

Acknowledgments This work was supported by ITS Edulab, a cooperation between Rijkswaterstaat Centre for Transport and Navigation and Delft University of Technology. The authors are grateful for the valuable suggestions and comments of the anonymous reviewers.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

Appendix: pseudocode EAS⁺ evacuation

The pseudocode of EAS⁺ evacuation is given in Fig. 14, and two subcodes are given in Figs. 15 and 16. The structure of an instruction set in these codes is *ant.origin.element*. This structure gives for each ant, for each origin, for each element corresponding to that origin the number of people assigned to the element.

Fig. 14 Pseudocode EAS⁺ evacuation

```

procedure EASPlusEvacuation
  Set heuristic information: heuristic_information[element]
  Set parameters: pheromone_evaporation_rate, epsilon
  input search_space
  // Initialize pheromone trails
  for number_element from 1 to final_element
    pheromone[number_element] ← 1
  end-for
  fitness_value_best_so_far ← 0
  while (termination condition not met) do
    ConstructAntSolutions
    UpdatePheromones
  end
end

```

```

procedure ConstructAntSolutions
  // Emptying the ants' memory
  for number_ant from 1 to final_ant
    for number_origin from 1 to final_origin
      for number_element_by_origin[number_origin] from 1 to ...
        final_element_by_origin[number_origin]
        ant[number_ant].origin.[number_origin].element[number_element_by_origin] ← 0
      end-for
    end-for
  end-for
  // Determining probabilities
  for number_origin from 1 to final_origin
    sum_probabilities ← 0
    for number_element_by_origin[number_origin] from 1 to ...
      final_element_by_origin[number_origin]
      sum_probabilities ← sum_probabilities + ...
      pheromone[number_element_by_origin] · ...
      heuristic_information[number_element_by_origin]
    end-for
    for number_element_by_origin[number_origin] from 1 to ...
      final_element_by_origin[number_origin]
      probabilities[number_origin].element[number_element_by_origin] ← ...
      pheromone[number_element_by_origin] · ...
      heuristic_information[number_element_by_origin] / sum_probabilities
    end-for
  end-for
  // Determining instruction sets
  for number_ant from 1 to final_ant
    for number_origin from 1 to final_origin
      for number_group_by_origin[number_origin] from 1 to ...
        final_group_by_origin[number_origin]
        random_number ← random[0,1]
        number_element ← 1
        probability_by_element ← ...
        probabilities[number_origin].element[number_element]
        while (probability_by_element < random_number)
          number_element ← number_element + 1
          probability_by_element ← probability_by_element +
            probabilities[number_origin].element[number_element]
        end-while
        ant[number_ant].origin.[number_origin].element[number_element] ← ...
        ant[number_ant].origin.[number_origin].element[number_element] + ...
        number_of_group_members[number_group_by_origin]
      end-for
    end-for
  end-for
  // Determining the fitness values for the created instruction sets
  for number_ant from 1 to final_ant
    Determine the fitness_value for the instruction_set created by number_ant
    (ant.origin.element) using a simulation model and an objective function
    if fitness_value > fitness_value_best_so_far
      fitness_value_best_so_far ← fitness_value
      best_so_far_instruction_set ← instruction_set
    end-if
  end-for
end

```

Fig. 15 Pseudocode construction phase

```

procedure UpdatePheromones
  for number_element from 1 to final_element
    pheromone[number_element] ← ...
     $(1 - \textit{pheromone\_evaporation\_rate}) \cdot \textit{pheromone}[\textit{number\_element}]$ 
  end-for
  for number_element_in_best_so_far_instruction_set from 1 to ...
    final_element_in_best_so_far_instruct
    pheromone[number_element_in_best_so_far_instruction_set] ← ...
     $\textit{pheromone}[\textit{number\_element\_in\_best\_so\_far\_instruction\_set}] + \textit{epsilon} \cdot \dots$ 
     $(\textit{fitness\_value\_best\_so\_far} / \textit{fitness\_highest}) \cdot \dots$ 
     $(\textit{number\_of\_people\_assigned\_to\_element} / \textit{number\_of\_people\_in\_corresponding\_origin})$ 
  end-for
end

```

Fig. 16 Pseudocode update pheromone phase

References

- Afshar AM, Haghani A (2008) Heuristic framework for optimizing hurricane evacuation operations. *Transport Res Rec* 2089:9–17
- Ben-Tal A, Mandala SR, Yao T (2009) Evacuation under data uncertainty: robust linear programming model. In: Proceedings of the 88th annual meeting of the transportation research board, Washington DC, USA
- Bliemer MCJ, Taale H (2006) Route generation and dynamic traffic assignment for large networks. Conference proceedings 1st DTA conference, Leeds, UK
- Chiu Y, Zheng H, Villalobos J, Gautam B (2007) Modeling no-notice mass evacuation using a dynamic traffic flow optimization model. *IIE Trans* 39(1):83–94
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Num Math* 1:269–271
- Dixit VV, Radwan E (2009) Optimal scheduling of evacuation orders for cities. In: Proceedings of the 88th annual meeting of the transportation research board, Washington DC, USA
- Dorigo M, Stützle T (2004) Ant colony optimization. MIT Press, Cambridge
- Franklin JL (2008) 2007 National Hurricane Center Forecast Verification Report, National Hurricane Center, NOAA/NWS/NCEP/Tropical Prediction Center
- Gaertner D, Clark K (2005) On optimal parameters for ant colony optimization algorithms. In: Proceedings of the international conference on artificial intelligence 2005, Las Vegas, USA, pp 83–89
- Huibregtse OL, Bliemer MCJ, Hoogendoorn SP (2010) Analysis of near-optimal evacuation instructions. In S.P. Hoogendoorn, A.J. Pel, M.A.P. Taylor, and H. Mahmassani (Eds.), 1st International conference on evacuation modeling and management Vol. 3. *Procedia engineering elsevier* (pp. 189–203). Amsterdam, Elsevier
- Li C-L, McCormick ST, Simchi-Levi D (1992) Finding disjoint paths with different path-costs: complexity and algorithms. *Networks* 22:653–667
- Liu Y, Lai X, Chang G (2006) Two-level integrated optimization system for planning of emergency evacuation. *J Transport Eng* 32(10):800–807
- Miller-Hooks E, Sorrel G (2008) The maximal dynamic expected flows problem for emergency evacuation planning. *Transport Res Rec* 2089:26–34
- Pel AJ, Bliemer MCJ, Hoogendoorn SP (2008) EVAQ: A new analytical model for voluntary and mandatory evacuation strategies on time-varying networks. In: Proceedings of the 11th IEEE intelligent transportation systems conference, Beijing, PR China, pp 528–533
- Pel AJ, Huibregtse OL, Hoogendoorn SP, Bliemer MCJ (2009) Model-based optimal evacuation planning anticipating traveler compliance behavior. In: Proceedings of the 12th international conference on travel behaviour research, Jaipur, India
- Saadatseresh M, Mansourian A, Taleai M (2009) Evacuation planning using multi-objective evolutionary optimization approach. *Eur J Oper Res* 198:305–314
- Sbayti H, Mahmassani HS (2006) Optimal scheduling of evacuation operations. *Transport Res Rec* 1964:238–246

-
- Stepanov A, Smith JM (2009) Multi-objective evacuation routing in transportation networks. *Eur J Oper Res* 198:435–446
- Wong KJ (2008) Parameter tuning for ant colony optimization: a review. In: *Proceedings of the international conference on computer and communication engineering*, Kuala Lumpur, Malaysia, pp 542–545