Capacitated network-flow approach to the evacuation-location problem

Published in:
Computers and Industrial Engineering

Publication status and date:
Published: 01/01/2018

DOI (link to publisher):
10.1016/j.cie.2017.11.026

Document Version
Publisher's PDF, also known as Version of record

Document License/Available under:
Article 25fa Dutch Copyright Act

Citation for the published version (APA):
Farahani, MM., Chaharsooghi, SK., Van Woensel, T., & Veelenturf, L. (2018). Capacitated network-flow approach to the evacuation-location problem. *Computers and Industrial Engineering*, *115*, 407-426. https://doi.org/10.1016/j.cie.2017.11.026

Computers & Industrial Engineering 115 (2018) 407–426

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Capacitated network-flow approach to the evacuation-location problem

Mina Mazraeh Farahani[a], S. Kamal Chaharsooghi[a,*], T. Van Woensel[b], Lucas P. Veelenturf[b]

[a] Faculty of Industrial Engineering, Tarbiat Modares University, Teheran, Iran
[b] School of Industrial Engineering, Eindhoven University of Technology, The Netherlands

A R T I C L E I N F O

Keywords:
Evacuation
Location problem
Emergency-logistics
Max-flow problem

A B S T R A C T

Evacuating people to the safe zones is the most crucial operation in managing many disasters. A mathematical model is presented in this paper, combining locational decisions with the max-flow problem in order to select safe destinations which maximize the number of dispatched people. The existing frameworks for emergency logistics usually model the evacuation process based on fixed and pre-determined destinations with a strategic perspective. The unpredictable and turbulent nature of a disaster may; however, disrupt the predictions. Furthermore, the primary goal in emergency situations is to dispatch people from the danger zone to a safe place, no matter where. A mixed integer linear programming model is developed in this paper for selecting one or more destinations in a capacitated network. The special structure of the model and its similarity to the max-flow problem allow us to develop exact algorithms and heuristics both for the multiple and single destination location problem. The solution methods are based on existing algorithms for the max-flow problem. Our proposed heuristics use the idea of adding a super-sink to the network to generate upper bounds very fast. The exact algorithms as well as the heuristics are tested on randomly generated instances as well as a real world network. The most important statistics of their computation times are reported. They are also compared according to their performance (gap to optimality) and their behavior amongst different categories of the graphs. Finally we have presented a real-case addressing the problem of choosing a number of destination locations from a fixed set of pilot pre-determined locations. The problem of deciding on the destinations is considered under 5 grades of disaster severity and the related impacts on choosing the safe zones are analyzed.

1. Introduction

The UNISDR[1] (ISDR, 2009) defines a disaster as “a calamitous event that seriously disrupts the function of a community or society and causes human, material, and economic or environmental losses that exceed the community’s or society’s ability to cope with using its own resources”. Though often caused by nature, disasters can have human origins. A disaster occurs when a hazard impacts vulnerable people with the effects exceeding their coping capacity (IFRC[2]).

Natural disasters are brought about by a natural change in the environment or by what is known as an act of God. On the other hand, disasters like Jilin chemical plant explosions (Fu, Fu, Skøtt, & Yang, 2008) or Mina Stampede (Saudi Arabia) (Ganjeh & Einollahi, 2016) are influenced by humans and they are often the result of negligence and human error. Whether the disaster is natural or man-made, the manner in which an action should be taken is too complex to be determined all along the disaster. In both cases, casualties should be moved to a safe place and treated immediately and the best way to meet this end is by putting the necessary measures in place that counteract this.

Emergency management is performed with a long-term horizon and includes three types of operations: 1. Long time before the disaster (strategic decisions), e.g. facility location, and stock pre-positioning. 2. Immediately after the disaster (tactical decisions), e.g. relief distribution and casualty evacuation, and 3. Long time after the disaster, e.g. recovery and reconstructing operations. The literature is generally broken down into these parts: facility location, relief distribution and casualty transportation, resource allocation, and commodity flow. A comprehensive review of these operations and the corresponding optimization problems can be found in Caunhye, Nie, and Pokharel (2012).

In this paper we choose the best destinations amongst several possible locations with the objective of maximizing the number of evacuees by integrating two problems in emergency management: the evacuation and the location problem and we call it the Evacuation-Location

* Corresponding author.
E-mail addresses: mina.mazraeh@modares.ac.ir (M. Mazraeh Farahani), skch@modares.ac.ir (S.K. Chaharsooghi), T.v.Woensel@tue.nl (T.V. Woensel), l.p.veelenturf@tue.nl (L.P. Veelenturf).

[1] United Nations International Strategy for Disaster Reduction.

[2] International Federation of Red Cross and Red Crescent.

https://doi.org/10.1016/j.cie.2017.11.026
Received 28 March 2016; Received in revised form 11 September 2017; Accepted 26 November 2017
Available online 02 December 2017
0360-8352/

Problem (ELP). We develop mixed-integer linear programming (MILP) models to combine network design and flow models. Then, relying on the special structure of the model and because of the countable set of possible destinations, we are able to develop exact algorithms for solving them. These algorithms are based on existing max-flow algorithms and are implemented on different instances. Heuristics are then proposed which can find feasible solutions very fast. The performance of these heuristics is compared according to both their gap to optimality and their computation time. Other results are also extracted by categorizing the structures of the randomly generated graphs. The paper is organized as follows: the related literature is reviewed in Section 2. We will discuss the assumptions and implications of the mathematical modeling for the Evacuation-Location Problem (ELP) in Section 3. In this section, we will also develop and explain the mathematical model of the problem. Then optimization algorithms are developed for each model in Section 4. Computational results of implementing the exact algorithms as well as the heuristics on randomly generated problems are reported in Section 5. In order to further demonstrate the real-world implications of our approach to the evacuation process, we present the real-case of Tehran metropolis while considering different disaster's severity and its consequences on locating the safe destinations. We will conclude the paper in Section 6 and give some recommendations for future works.

2. Literature review

Post-disaster phase logistics models fall under two major categories: (a) relief delivery/casualty transport models, (b) mass evacuation models. Özdamar and Ertem (2015) provide extended lists and categories of models and solution methodologies in this area. Considering the number of publications on humanitarian logistics, evacuation modeling and planning has gained a lot of mindshare amongst transportation professionals. Most previous studies in evacuation modeling tackle the problem by extending the applications of conventional transportation planning and traffic management models.

Mass evacuation models concentrate on car evacuation (CE) and traffic flow management, and also, on evacuation by public transit (PT) (Özdamar & Ertem, 2015).

In the conventional evacuation planning process, evacuees are assigned to fixed destinations mainly based on the criterion of geographical proximity. However, such pre-specified facilities almost always lead to a sub-optimal evacuation process due to additional uncertainties such as congestion, road blockage, and any other hazards associated with the emergency. By relaxing the constraint of assigning evacuees to pre-specified destinations, the idea of choosing the evacuation destination has the potential of greatly improving the evacuation efficiency.

Although the transient and possibly chaotic nature of evacuation makes it very challenging to manage the flow of people and causalities, there lies a unique advantage in evacuation planning. In other words, as long as an evacuee safely exits the evacuation zone in a timely fashion, it is not so important which route is taken or at which location the evacuee leaves the zone. This flexibility in an evacuee's destination selection, and its associated benefit to the planning process, is neither well understood nor often exploited.

At the same time, in the absence of strategic infrastructures, the importance of in-time (short term) decision making for evacuation and habitation becomes more significant. Experience shows that this is a completely probable situation in developing countries (for more details on the relationship between development and disaster risks see e.g. Pelling, Maskery, Ruiz, & Hall, 2004). An example of the need for immediate evacuation planning could be Bam earthquake in Iran (2003), in which about 40% of the patients stayed within the surrounding area of their homes for 8–10 h. The majority of casualties (57.6%) were transferred manually to the first place of settlement and 45.8% were taken to the second place of settlement using blankets. The emergency medical service system in Bam was destroyed and was not able to respond adequately (Mirhashemi, Ghanjal, Mohebbi, & Moharamzad, 2007). One of the most important and ignored issues in this research field is the selection of the best destination(s) for establishing relief centers or providing temporary shelters immediately after the disaster. Past experience suggests that a major problem in evacuation operations is the insufficiency of existing routes in terms of number and capacity in a large-scale emergency evacuation (Perry, 1985).

At the strategic level, existing literature addresses the evacuation problem from two different aspects: analyzing the impact of behavioral and managerial factors on evacuation (Dow & Cutter, 1998; Drabek Thomas, 1999; Lin & Jaillet, 2014; Urbina & Wolshon, 2003) and locating safe zones and allocating evacuation zones to the disaster zones before the disaster (Sbayti & Mahmassani, 2006).

In most cases, exploiting new transportation infrastructures is simply too cost-prohibitive to be considered practical in case of emergency. Therefore, finding ways to improve the planning and operational aspects of the evacuation process to maximize the utility of the existing transportation network has often been the focus of past studies. Some of these efforts include the implementation of counter flow and contra flow lanes (Muoz, Sun, Horowitz, & Alvarez, 2006), staggering departure times (Sbayti & Mahmassani, 2006), traffic signal control, multi-jurisdiction coordination, special routing consideration for heavy vehicles (Franzese & Han, 2001), and so on. Yet no effort had recognized or explored the aforementioned flexibility in evacuees' destination selection until Yuan and Wang (2009) proposed the concept of the most-desirable destination(s) for evacuees. In another study, Yuan, Han, Chin, and Hwang (2006) present a framework for the simultaneous optimization of evacuation-traffic distribution and assignment. They apply their approach in a hypothetical nation-wide simulation case study but they don't develop an analytical algorithm for the problem. They assume a fixed demand in the source node and no limitation on the number of destinations.

Ahuja, Magnati, and Orlin (1993) describe the problem of "building evacuation" modeled by network flow theory. In this approach the building (zone) to be evacuated is represented by a network in which the nodes correspond to relevant parts of the evacuation object (e.g., rooms in a building, intersections of streets in a zone), and arcs represent connections between these parts (e.g., doors between rooms, streets). One or several of the nodes are distinguished as sources (e.g., rooms containing evacuees, the origins of the disaster occurrence) and one or multiple others as sinks (representing the safe destinations). The group of evacuees is modeled as the flow which passes through the network over time. Important questions including the followings can be answered by using methods of dynamic network flow theory (Hamacher, Heller, & Rupp, 2013):

- How many persons can be evacuated before a given time T?
- How much time does it take to evacuate all persons in a given evacuation scenario?
- What are the bottlenecks in the evacuation network?

Network flow models were extendedly applied to produce lower bounds for evacuation time. To the best of our knowledge the survey of Chalmet, Francic, and Saunders (1982) is the first work in this area. They were later used by many other researchers (e.g. see Hoppe & Tardos, 1994).

Hoppe and Tardos (1994) provided a polynomial algorithm for the evacuation problem with a fixed number of sources. They also gave the only known polynomial algorithm for the discrete quickest trans-shipment problem with an arbitrary number of terminals in Hoppe and Eva (2000). Fleischer and Tardos (1998) extended this result for continuous time. Sayyady and Eksioglu (2010) provide a mixed integer program with the objective of simultaneously minimizing the total evacuation time and the number of casualties. They design a Tabu search algorithm that finds evacuation routes for transit vehicles.

In a critical review, Abdelgawad and Abdulhai (2009) state that the evacuation planning process is generally modeled as a network design problem (NDP). Conceptually, the objective of the NDP is to optimize a given system performance measure such as maximizing the total number of evacuees or minimizing the total system travel cost while accounting for the route choice behavior of network users. As their study shows, no effort was made regarding the restriction of fixed destinations, except the work of Sbayti and Mahmassani (2006). Sbayti and Mahmassani (2006) introduced an optimal evacuation scheduling but have made major assumptions, by relaxing both the destination selection procedure and evacuees compliance. Nevertheless, the final output of their study is an optimal loading curve that would minimize the total network clearance time.

Lim, Zangeneh, Baharnemati, and Assavapokee (2012) present a capacity constrained network flow optimization approach. Dijkstra's algorithm is used in their study for finding evacuation paths, flows and schedules which maximize the total number of evacuees for short notice evacuation planning. To the best of our knowledge there are no papers considering location decisions in emergency conditions as tactical and on-line except the study done by Yuan and Wang (2009). However, their study does not take into account the location decision variables for improving the evacuation flow. Many studies incorporate uncertainties of emergency logistics and solve it with a stochastic programming approach (see. e.g. Bozorgi-Amiri, Jabalameli, & Mirzapour Al-e Hashem, 2011; Tofighi, Torabi, & Mansouri, 2016).

In a recent survey Hamacher et al. (2013) combine two modeling tools: dynamic network flow and locational analysis with the aim of placing emergency/aiding units on the routes (arcs) of a network. They develop two static and dynamic integer-programming models and propose exact and approximate algorithms for solving them. On the other hand, most humanitarian emergency logistics facility location problems were found to be minisum, set covering, miximal covering, and minimax facility location problems. Obnoxious facility location problems were the least proposed problems (Boonmee, Arimura, & Asada, 2017). From grouping the facility location types in humanitarian logistics, according to disaster phases and the corresponding inetr-facility flows, further insights are provided about the existing and potential objective functions. Fig. 1 summarizes the facility location types, their associated flows and different objective functions implied by the facility type and disaster phase. The representation is separated into pre and post-disaster phases.

Different directed arcs are used in Fig. 1 to show human and commodity flows between the facilities. The boxes show the facility type which has to be located before or after the disaster. Abounacer, Rekik, and Renaud (2014) studied a facility location problem with a transportation problem for disaster response. Afshar and Haghani (2012) proposed a mathematical model that integrated a relief commodity flow problem, a facility location problem, a vehicle routing problem, and a transportation problem. Bayram, Tansel, and Yaman (2015) developed a model that optimally located shelters and assigned evacuees to the nearest shelter site considering potential disaster sites. Khayal, Pradhananga, Pokharel, and Mutlu (2015) presented a network flow model for dynamic selection of temporary distribution facilities. Research until now advocates facility location separated from the evacuation process and as a pre-disaster operation. This leads to fixed-location shelters, distribution and medical centers during the response phase (Fig. 1a). In a dynamic and uncertain disaster environment; however, it is possible to consider facility location modeling in the post-disaster phase (Fig. 1b). Temporary distribution and treatment centers are oftentimes set up for disaster relief and casualty transportation, respectively. Facility location problems can be supported or developed to combine aspects such as routing problems, evacuation problems, relief distribution problems, casualty transportation and traffic problems, inventory problems and resource allocation problems. Most objectives have focused on minimum time, minimum cost, minimum distance, minimum number of located facilities, and coverage by a maximum number of demand points. New objective functions could be developed by integrating the facility location problem with the other above-mentioned problems (Boonmee et al., 2017) such as maximum number of evacuees or minimum time of evacuation. In the mitigation stage, future research could seek to manage the disaster risks by relocating inhabitants farther from the vulnerable area. In the post-disaster stage, best shelter locations must be selected or temporary shelters need to be rapidly identified as a tactical decision. Research could also investigate optimal locations for temporary distribution centers (sub-distribution centers) to ensure efficient commodity distribution, and also to determine the optimum placement of temporary medical centers to ensure that the wounded are treated rapidly. Finally, in the recovery stage, special attention should be paid to garbage dumps in order to remove any debris from the area. This has not been fully studied (Boonmee et al., 2017).

3. Evacuation-Location Problem (ELP)

Evacuation-Location Problem (ELP) is a combination of two subproblems: the problem of selecting the safe zones and the evacuation problem. In this paper the aim is to evacuate as many people as possible from a densely populated area s called source, to a fixed number of shelters p called destinations or sinks. By applying ELP in the post-disaster phase, we can reduce the challenges arising from uncertainties especially if the data about the routes and capacities are available. It also makes the ad hoc decision making, possible in order to effectively employ resources and handle emergency plans.

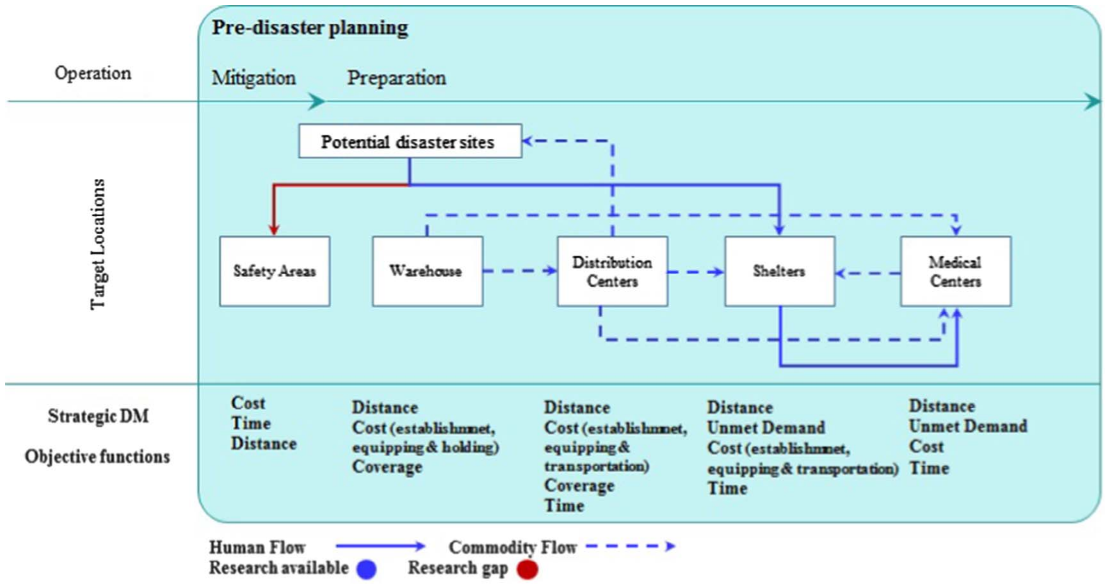
The assumption, here, is that the shelters or safe zones are not limited by their capacity and there is no preferences among them. We have also assumed integer capacities since the set of vehicles or people are denumerable. All nodes except the source and the sinks are intermediate (transit) nodes. We will use the capacitated max-flow network approach to model the evacuation problem. Our max-flow network model is modified to incorporate the locational decisions immediately after a disaster. Though the max-flow problem and the location problem are both solvable in polynomial time separately, this does not necessarily hold for the ELP.

3.1. Problem definition

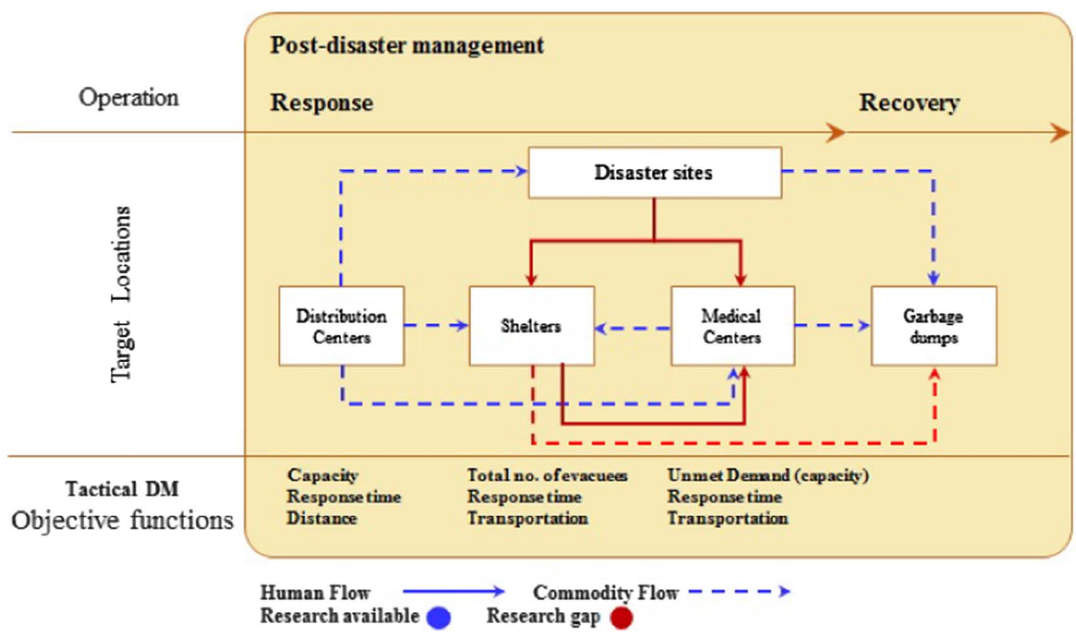
Choosing amongst a set of available locations for evacuees' accommodation impacts the efficiency of the evacuation process both in terms of the number of rescued people and the time of evacuation. In reality, for many towns and cities, emergency evacuation or disaster preparedness guides, not only provide a list of limited shelters but also emphasize that not all the shelters are available during a disaster. They urge the residents to check to see which shelters could potentially be opened. For example, it is emphasized in hurricane shelter guide of Polk County in Florida, U.S.A., that: "All shelters are not automatically activated for each possible emergency…they are not pre-assigned by geographic area" (B. o. C.C. Polk County, 2017). For other examples please see B.o.C.C. Alachua County (2017) and B. o. C.C. Collier County (2017).

This is due to the opening and equipping time and cost of the shelters, because the pre-defined shelters are mostly public schools, parks and/or stadiums. Without loss of generality, however, our approach covers the problem of evacuating to all possible destinations. This version of the problem will then be solvable in polynomial time with a reduction to multiple-sink max-flow problem (Kotnyek, 2003).

In addition to this, if the population at the disaster point(s) is less than the total capacity of the shelters, it would be more efficient to choose amongst the available shelters. We have addressed the real limitations of opening all possible destination locations by assuming a fixed number of destinations to be chosen/opened. Though for the sake of simplicity in this first attempt in combining location and evacuation problems, we have not assumed any differences among the locations in



(a) Pre-disaster facility location types, associated flows and objective functions



(b) Post-disaster facility location types, associated flows and objective functions

Fig. 1. Facility locations in humanitarian logistics and their corresponding objective functions.

terms of the opening or equipping costs.

In the maximum flow problem, the aim is to find the maximum flow between the source and the sink. A flow is defined on a directed graph N = (V,A) with the capacity function $u: A \longrightarrow \mathbb{Z}^+$ defined on arcs. A set of non-negative values $x: A \rightarrow \mathbb{Z}^+$ is a flow if it satisfies the flow conservation rule, meaning that the net flow at every node equals zero except for the sink and the source. A flow is feasible if it complies with the capacity constraints. The value of a flow f is usually denoted by $|f|$. We assume a familiarity with the mathematical formulation of the max-flow problem, so we go straight on to the definition of the ELP. Interested readers can refer to Ahuja et al. (1993) and Ford and Fulkerson (2015) for more details on flow network problem.

Definition 1. Consider the emergency logistics network $N = (V,A)$ where V is the set of vertices, $A \subseteq V \times V$ is the set of arcs with capacities defined by the function $u: A \longrightarrow \mathbb{Z}^+$. Suppose that $|A| = m, |V| = n$. Also, let L be the set of possible destinations (sinks) such that $|L| = q < n$. Also suppose that s is the source node. The set of arcs entering and leaving node i are denoted by $\delta^-(i) = \{(j,i) \in A | j \in V\}$ and $\delta^+(i) = \{(i,j) \in A | j \in V\}$ respectively. $\Delta_l = \{\sum_{i \in \delta^-(l)} u(i,l) | l \in L\}$ is called the "inflow capacity" of node $k \in L$. Without loss of generality we assume that $\delta^-(s) = \delta^+(l) = \varnothing, l \in L$.

We are looking for p terminal locations which maximize the amount of flow f in the network and we call this the Evacuation-Location Problem ELP. The MILP formulation of ELP is presented in the next section.

3.2. Model formulation

In this part we formulate the mixed integer programming model for the ELP based on the max-flow problem.

Table 1
Notations.

Sets	Description
V	Set of nodes including the set of sink nodes L; $\|V\| = n$
L	Set of destination locations; $\|L\| = q$
A	Set of arcs, $(i,j) = a_{ij}$ equals 1 if node i is connected to node j, otherwise is 0; $\|A\| = m$
$\delta^-(i)$	Set of arcs entering node i∈A; $\{(j,i) \in A \| j \in V\}$
$\delta^+(i)$	Set of arcs leaving node i∈A; $\{(i,t) \in A \| j \in V\}$
Parameters	Description
u_{ij}	The capacity of arc $(i,j) = a_{ij}$
p	Number of destinations to be selected;
q	Total number of possible destination locations
Variables	Description
y_l	equals 1 if the lth location is selected, otherwise is 0
ϑ	The amount of network flow
x_{ij}	The flow passing through the arc $(i,j) = a_{ij}$

3.2.1. Model 1 Evacuation-Location Problem

Based on the notations in Table 1 we present the mixed integer linear programming (MIP) model for Evacuation-Location Problem ELP.

$$\text{Max } \vartheta \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} - \sum_{h=1}^{n} x_{hi} = 0 \qquad \forall\, i \in V,\, i \neq 1,\, i \notin L \tag{2}$$

$$\sum_{j=1}^{n} x_{1j} = \vartheta \tag{3}$$

$$\sum_{l\in L} \sum_{i=1}^{n} x_{il} = \vartheta \tag{4}$$

$$\sum_{l\in L} y_l = p \tag{5}$$

$$\sum_{i=1}^{n} x_{il} \leqslant \Delta_l \cdot y_l \qquad \forall\, l \in L \tag{6}$$

$$y_l = \{0,1\},\ 0 \leqslant x_{ij} \leqslant u_{ij}, \vartheta \geqslant 0 \tag{7}$$

Our objective is to maximize the total flow of evacuation, reflected in Eq. (1). Eqs. (2), (3), (4), and (7) assure the feasibility of the flow. Constraint (5) ensures that only p locations are chosen and the connection between the maximum flow problem and the location decision is made with constraint (6), where the total flow entering a selected location must be less than or equal to the summation of the capacities of arcs ending in that location. Though we have formulated the single source network, the formulation can be extended to a multiple source network by adding a super-source S which is connected to every source $\{s_1,\dots,s_c\}$ with the arc capacity $u(S,s_j) = \Delta_{s_j}$ s.t. $1 \leqslant j \leqslant c$.

3.2.2. Model 2. Evacuation-Location Problem for $p = 1$

Putting $p = 1$ reduces the problem to the single Evacuation-Location Problem. Eq. (5) is then equivalent to $\sum_{l\in L} y_l = 1$. So, we are looking for one terminal location amongst q locations which maximizes the amount of flow f in the network. In real world, this may be the case when the population in the danger zone is not that big to be dispersed in multiple locations, or the cost of opening of any of the safe zones is too high. There may also be disasters with the risk of disease or viruses outbreaks, in which the decision makers prefer to keep the evacuees in as few places as possible, even in a single safe site.

4. Solution methodology

In this section first two upper bounds are presented based on the multi-terminal maximum flow algorithms. We then discuss two exact algorithms, exploiting the upper bounds and the structure of the problem. Two heuristics are described afterward which provide approximate solutions for the multiple and single ELPs, respectively.

4.1. Bounds

4.1.1. Upper bound 1: Relaxing the p destinations constraint

In order to reach an upper bound which could contribute in finding the optimum in less time we relax the constraint of choosing only p destinations among elements of L. Supposing the possibility of opening all locations, we can find an upper bound in polynomial time in the size of the input by solving a single-source multi-sink max-flow network problem (Kotnyek, 2003). For this we add a virtual sink t which is connected to all possible destinations with infinite capacity. It results in a single-source single-sink max-flow problem again.

Let N_e be the extended network constructed from N by adding the super-sink t and connecting all members of L to the super-sink as we are going to open all locations. They are connected to t by artificial arcs with infinite capacity. Consider $F_{max}^{N_e}$ to be the maximum flow value in N_e calculated by any of the max-flow algorithms and let $f_l^{N_e}$ be the flow on arc (l,t), such that $|f_{max}^{N_e}| = \sum_{l\in L} |f_l^{N_e}|$. Details can be seen in Algorithm UB1.To compute the max-flow we used the Edmonds and Karp (1972) max-flow algorithm, which is strongly polynomial and has the complexity of $O(n\cdot m^2)$.

Algorithm UB1. Upper bound for ELP.

Input: Network $N = (V,A,u), L \subseteq V, |L| = q$
Output: $f_{max}^{N_e}$
$V' = V \cup \{t\},\ A' = A \cup \{(l,t); l \in L\},\ u' = u \cup \{u(l,t) = \infty; l \in L\}$,
$N_e = (V',A',u')$
$f_{max}^{N_e} = Maximum\ Flow\ Edmonds{-}Karp(N_e = (V',A',u'))$
return $f_{max}^{N_e}$

4.1.2. Upper bound 2: Relaxing the capacity constraint of the sink nodes

Another upper bound of the objective function could be achieved by relaxing the capacity of the links reaching to the destination locations by eliminating the destination nodes from the network and substituting them with a single super-sink t. Doing this, we will further extend the search space by relaxing the capacity constraints for destination nodes.

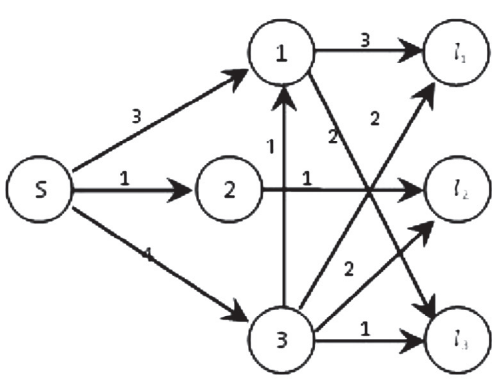
Assume that N_r is the *relaxed network* with super sink t such that $N_r = (V',A',u')$ such that $A' = A \cup \{(i,t) | (i,l) \in A \wedge i \in V \wedge l \in L\}$ and

$$u'(i,j) = \begin{cases} u(i,j) & j \neq t \wedge j \notin L \\ \infty & j = t \\ 0 & j \in L \end{cases}$$

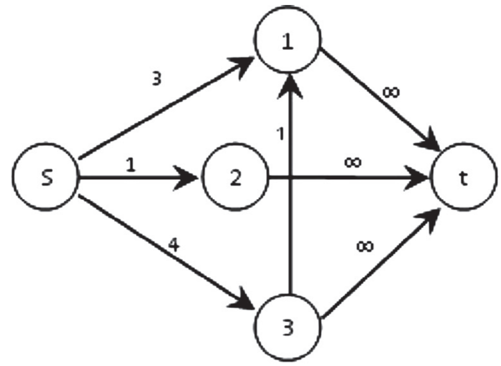
Let $f_{max}^{N_r}$ be the maximum flow in N_r. Since the feasible region generated from N_r is bigger than the one generated by $N, f_{max}^{N_r}$ will not be less than f^* (optimum of the objective function of the ELP) and is then an upper bound. The procedure is represented in Algorithm UB2.

Algorithm UB2. Upper bound for ELP.

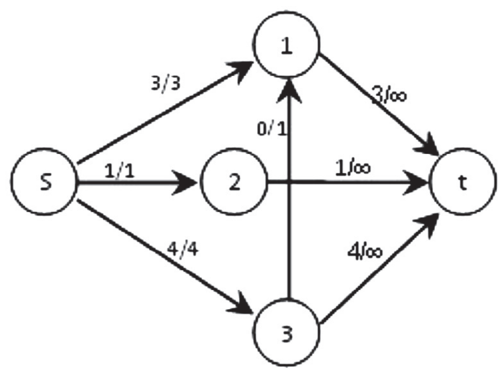
Input: Network $N = (V,A,u), L \subseteq V, |L| = q$, Number of destinations to be selected:p
Output: $f_{max}^{N_r}$
$V' = V \cup \{t\},\ A' = A \cup \{(i,t);(i,l) \in A \wedge l \in L\},\ N_r = (V',A',u')$
for all $i,j \in V$ **do**
  if $j \neq t \wedge j \notin L$ **then**
    $\acute{u}(i,j) \leftarrow u(i,j)$
  else if $j = t$ **then**
    $\acute{u}(i,j) \leftarrow \infty$



(a) Original network N with destination set $\{l_1, l_2, l_3\}$.



(b) Relaxed Network N_r with super-sink *(t)*



(c) Max-flow for Relaxed Network of the ELP problem in Example 2 $|f_{max}^{N_r}| = 8$

Fig. 2. Relaxed network of the ELP in Example 1.

else if $j \in L$ **then**
  $\acute{u}(i,j) \leftarrow 0$
end if
end for
$f_{max}^{N_r} = Maximum\ Flow\ Edmonds\text{–}Karp(N_r = (V',A',u'))$
return $f_{max}^{N_r}$

Corollary 1. *Suppose that S^{N_r},S^{N_e} and S are the feasible spaces of their corresponding max-flow problems for networks N_r,N_e and N, respectively. Since $S^{N_r} \geqslant S^{N_e} \geqslant S$, for flow vectors $f_{max}^{N_e}$,$f_{max}^{N_r}$ and f^* (as defined earlier), the following holds:*

$$|f_{max}^{N_r}| \geqslant |f_{max}^{N_e}| \geqslant |f^*|.$$

Example 1. Consider the primary network illustrated in Fig. 2. The numbers on each edge show the capacity u_{ij}. The relaxed network is shown in Fig. 2b with all candidate destinations being substituted with the super sink t. Fig. 2c shows the max-flow for N_r for which $|f_{max}^{N_r}| = 8$,$|f_1^{N_r}| = 3$,$|f_2^{N_r}| = 1$, and $|f_3^{N_r}| = 4$.

4.2. Exact algorithms

4.2.1. Exact Algorithm (E1) for ELP

In this part we introduce an exact algorithm which is also based on the maximum flow algorithm of Edmonds and Karp (1972). Since the number of possible locations is finite ($|L| = q$) the ELP can be solved by repeating the maximum flow algorithm $\binom{q}{p}$ times. In every iteration of the proposed algorithm, p amongst q locations will be chosen. Let $P(L)$ denote the set of all subsets of L (power-set of L), then we are looking for the set $P_p(L) = \{\gamma \in P(L)|;|\gamma| = p\}$. Accordingly, the problem will be solved as a multi-sink max-flow problem. In order to do this, we will connect each of the p locations to the virtual sink t with an arc with infinite capacity. Here we use the Edmonds-Karp algorithm (Edmonds & Karp, 1972). In step 6 of E1 we have the maximum flow reaching the virtual sink. This step has the complexity order of $(n+1)\cdot(m+p)^2$. As a result, E1 for the ELP has at most the overall complexity order of $\binom{q}{p}\cdot(n)\cdot(2m)^2$. See Algorithm E1 for details.

Algorithm E1. Exact solution for ELP $O\left(\binom{q}{p}\cdot(n)\cdot(2m)^2\right)$.

Input: Network $N = (V,A,u)$,$L \subseteq V$,$|L| = q$, Number of destinations to be selected:p
Output: f^*,L^* (V',A',u')
1 $f^* \leftarrow 0$;$L^* \leftarrow \varnothing$
2 $(V' = V \cup \{t\}$,$A' = A \cup \{(l_k,t)|k = 1,...,q\})$
3 **for all** combinations $\gamma \in P(L) s.t. |\gamma| = p$ **do**
4   $u' = u \cup \{u_{l_k,t} = \infty | k = 1,...,q\}$, $N' = (V',A',u')$
5   $L^* \leftarrow \gamma$
6   **for all** $l \notin L^*$ **do**
7     $\acute{u}_{l_k,t} \leftarrow 0$
8   **end for**
9   $Optflowtemp = Maximum\ Flow\ Edmonds\text{–}Karp(N' = (V',A',u'))$
10   **if** $f^* \prec Optflowtemp$ **then**
11     $f^* \leftarrow Optflowtemp$, $L^* \leftarrow \gamma$
12   **end if**
13 **end for**
14 **return** f^*,L^*

4.2.2. Exact Algorithm (E2) for ELP

In UB1 we relaxed the constraint of choosing only p destinations to reach an upper bound. Here, we will modify UB1 to find an exact solution. This algorithm has two building blocks, namely an upper bound and a narrowing procedure. The narrowing procedure uses the upper bound (UB1) as a basis for reducing the search nodes while guaranteeing the feasibility of the solution. Let N_e be the extended network defined in UB1. Consider $f_{max}^{N_e} = (f_1^{N_e},\dots,f_l^{N_e})$ to be the maximum flow in UB1 with all locations opened. Let $f_l^{N_e}$ be the flow on arc (l,t), such that $|f_{max}^{N_e}| = \sum_{l\in L} |f_l^{N_e}|$. Suppose that $\underbrace{(f_1^{N_e}, f_2^{N_e},\dots,f_p^{N_e})}_{p\ values}$ is a sorted vector such that $f_1^{N_e} \geqslant f_2^{N_e} \geqslant \dots \geqslant f_p^{N_e}$ and $f_1^{N_e}, f_2^{N_e},\dots,f_p^{N_e}$ are p largest values of $f_l^{N_e}$, $\forall\ l \in L$. The corresponding destination ordered $p-tuple$ is $\overrightarrow{\gamma^*} = (l_1^*, l_2^*,\dots,l_p^*)$. Then, we solve the multi-sink max flow problem for the new destination vector $\overrightarrow{\gamma^*}$ where

$$\acute{u}(l_k^*,t) = \begin{cases}\infty & l_k^* \in L\\ 0 & \text{Otherwise}\end{cases}$$

Let the resulting maximum flow be $f^{In} = (f_1^{In}, f_2^{In},\dots,f_p^{In})$. f^{In} is feasible but not necessarily the optimum flow. Without loss of generality we assume that the flow vector f^{In} is sorted in ascending order, i.e. $f_1^{In} \geqslant f_2^{In} \geqslant \dots \geqslant f_p^{In}$.

Proposition 1. *For node $l \in L$ to be capable of improving the value of f^{In}, it is necessary that $\Delta_l \geqslant f_p^{In}$.*

Proof. By contradiction; suppose that there is a node $v \in L$ with inflow capacity Δ_v strictly smaller than f_p^{In} where f_p^{In} is the p^{th} element of f^{In} as defined earlier, i.e. $\Delta_v < f_p^{In}$. Let $\widehat{L} = (l_1^{In}, l_2^{In},\dots,l_{p-1}^{In}, v)$ be the vector of new best locations. Solving the multi-sink max-flow problem for the network $\widehat{N_e} = (V',A',u')$ where $V' = V \cup \{t\}$, $A' = A \cup \{(l_k^{In},t); l_k^{In} \in \widehat{L}\}$, $u' = u \cup u(l_k^{In},t) = \{\infty; l_k^{In} \in \widehat{L}\}$ gives $\widehat{F} = (\widehat{f}_1,\dots\widehat{f}_{p-1}\widehat{f}_v)$ to be the maximum flow for $\widehat{N_e}$. Suppose that:

$$|\widehat{f}\,| = |f^{In}| + \epsilon; \epsilon \geqslant 0$$

and

$$|\widehat{F}| = \sum_{k=1}^{p-1} |\widehat{f}_k| + |\widehat{f}_v| \tag{8}$$

So, there should be at least one augmenting path with capacity ϵ from s to t. As $\Delta_v < f_p^{In}$ and there is no link between any two destinations (since $\delta^-(s) = \delta^+(l) = \varnothing, l \in L$) the augmenting path must pass through one of the locations $l_p^{In}, k = 1,\dots,p-1$ in $\widehat{N_e}$. Hence, there must exist an augmenting path from s to one of the locations $l_p^{In}, k = 1,\dots,p-1$ in $\widehat{N_e}$ and this contradicts the maximality of f^{In} and f_k^{In}, k = 1, …, p − 1. □

Note that if any of the destinations are connected to the other one by an arc, this connecting arc could generate an augmenting path when its tail is substituted, regardless of the inflow capacity of the tail.

Corollary 2. *Let $f^* = (f_1^*,\dots,f_p^*)$ be the optimal flow vector which is sorted in descending order i.e. $f_1^* \geqslant \dots \geqslant f_p^*$. Also, suppose that the corresponding location vector is $L^* = (1,\dots,p)$ with $p \leqslant |L| = q$. If f^* is the optimal solution for ELP with $N = (V,A,u,L)$ then $L^* = (1,\dots,h)$ where $h \leqslant p \leqslant q$ is the optimal location vector for ELP with the objective of selecting h destination locations in $N = (V,A,u,L)$ (Note that for f^* being optimal implies that f_k^* is also the maximum flow at node k for $k = 1,\dots,p$).*

Referring to Proposition 1 we will improve the computation time by removing some candidate locations from the search space. This is explained in our second exact algorithm (Algorithm E2). At first we calculate the upper bound flow vector as described in UB1 for N_e with all locations opened. Then, with the sorted vector $f^{In} = (f_{l_1}^{In},\dots,f_{l_p}^{In})$ at hand, we omit the destinations with the inflow capacity smaller than $f_{l_p}^{In}$; $p \leqslant |L| = q$. Then Algorithm E1 is applied to $N = (V,A,u,rL)$ where $rL = \{l \in L | \Delta_l \geqslant f_{l_p}^{In}\}$ is the residual set of destination locations.

The optimality of Algorithm E2 is guaranteed according to Proposition 1, although there is the possibility that no node will be removed. Therefore its complexity depends on the structure of the network. So, the worst case complexity of this heuristic will be of the order of Algorithm E1.

Algorithm E2. Exact solution for *ELP*.

Input: Network $N = (V,A,u), L \subseteq V, |L| = q$, Number of destinations to be selected: p
Output: f^*, L^*
$V' = V \cup \{t\}$, $A' = A \cup \{(l,t)|l \in L\}$, $u' = u \cup \{u_{lt} = \infty | l \in L\}$, $N_e = (V',A',u')$
$f_{max}^{N_e} \leftarrow UB1(N_e = (\acute{V},\acute{A},\acute{u},t))$
$f_{max}^{N_e} := (f_1^{N_e},\dots,f_p^{N_e})$
$\overrightarrow{L^{N_e}} := (l_1^{N_e},\dots,l_p^{N_e})$
$V^{``} = V \cup \{t\}$, $A^{``} = A \cup \{(l_k^{N_e},t)|k = 1,\dots,p\}$, $u^{``} = u \cup \{u(l_k^{N_e},t) = \infty | k = 1,\dots,p\}$, $N_{In} = (V^{``},A^{``},u^{``})$
$f^{In} = Maximum\ Flow\ Edmonds\text{–}Karp(N_{In} = N_{In} = (V^{``},A^{``},u^{``}))$
$f^{In} = (f_1^{In},\dots,f_p^{In})$ ◇f^{In} is assumed to be sorted in descending order
for $l = 1$: q **do**
  if $\Delta_l < f_p^{In}$ **then**
    $rL = L \backslash l$
  end if
end for
E1$(N = (V,A,u,rL))$
return f^*, L^*

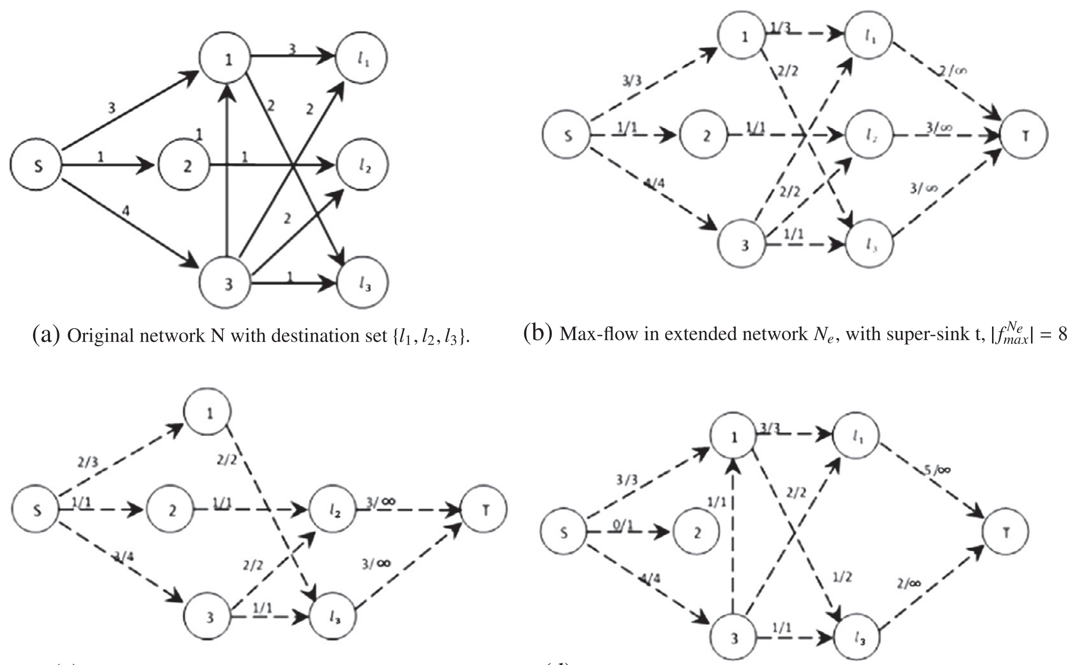
4.3. Heuristics

4.3.1. Heuristic 1. Satisfying the capacity constraints

In heuristic H1 we directly use the procedure for generating f^{In} in Algorithm E2 as a heuristic for the ELP. The initial solution of Algorithm E2 which is generated by applying and modifying Algorithm UB1 is a feasible flow which not only satisfies the capacity constraints but also chooses exactly p locations from L. See Procedure H1 for details. This heuristic has the complexity of the order of $(n+1)\cdot(m+q)^2 + (n+1)\cdot(m+p)^2 + O(Sort(f_{max}^{N_e}))$ using Edmonds-Karp algorithm (Edmonds & Karp, 1972) for the maximum flow calculation, where $O(Sort(f_{max}^{N_e}))$ is the order of the algorithm used for sorting the upper bound flow vector of the destination nodes.

Procedure H1. Heuristic Solution for **ELP**

Input: Network $N = (V,A,u), L \subseteq V, |L| = q$, Number of destinations to be selected: p
Output: *Maxflow, BestLoc*
$V' = V \cup \{t\}$, $A' = A \cup \{(l,t)|l \in L\}$, $u' = u \cup \{u_{lt} = \infty | l \in L\}$, $N_e = (V',A',u')$
$f_{max}^{N_e} \leftarrow UB1(N_e = (\acute{V},\acute{A},\acute{u},L,p))$
$f_{max}^{N_e} := (f_1^{N_e},\dots,f_p^{N_e})$ ◇$f_{max}^{N_e}$ is assumed to be sorted in descending order
for $i = 1$: p **do**
  $BestLoc(i) \leftarrow (f^{Inverse})_i^{N_e}$
end for
for all $i \notin BestLoc$ **do**
  $u'(i,t) \leftarrow 0$



(a) Original network N with destination set $\{l_1, l_2, l_3\}$.

(b) Max-flow in extended network N_e, with super-sink t, $|f_{max}^{N_e}| = 8$

(c) Solution for ELP applying H1 to network N, $|Max\_flow| = 6$, selected destinations=$\{l_2, l_3\}$

(d) Optimum Solution for ELP applying E1 to network N, $|Max\_flow|$=7, selected destinations=$\{l_1, l_3\}$.

Fig. 3. Counterexample demonstrating that H1 may not always result in optimum solution.

end for
$Maxflow = Maximum\ Flow\ Edmonds{-}Karp(N_e = (V', A', u'))$
return $Maxflow, BestLoc$

For a counterexample that demonstrates H1 will not always lead to an optimal solution, see Fig. 3. Assume that we are going to choose two destinations from the set of all destinations $\{l_1, l_2, l_3\}$ with the objective of maximizing the number of evacuees from the source s as depicted in Fig. 3a. The numbers on the edges correspond to their capacity. Applying H1 for the 3-sink max-flow problem in Fig. 3 we get to $|f_{max}^{N_e}| = 8$, $|f_{l_1}^{N_e}| = 2$, $|f_{l_2}^{N_e}| = 3$ and $|f_{l_3}^{N_e}| = 3$ which is represented in Fig. 3b The numbers in Fig. 3b correspond to the maximum flow on every edge. Edges with zero flow are deleted for clarity. According to H1, l_2 and l_3 are the two selected destinations. Solving the max-flow problem for the multi-sink network including two sinks l_2 and l_3 the max-flow value is 6 units, the corresponding flow vector of which is shown in Fig. 3c. We solve the problem once again using the exact Algorithm E1, which results in selecting another set of destinations, i.e. l_1 and l_3 and the maximum flow will be equal to 7 units, as demonstrated in Fig. 3d.

4.3.2. Heuristic 2. Approximate solution for the single destination (p = 1) ELP

We introduce another heuristic here which again uses the idea of adding a super-sink to the network, but in a different way than H1. It is also based on our second upper bound, Algorithm UB2 which can only be applied for $p = 1$.

Referring to Algorithm UB2 let $f_i^{N_r}$ be the flow on arc (i,t) in N_r, such that $|f_{max}^{N_r}| = \sum_{i:(i,l)\in A} |f_i^{N_r}|$. We transfer the information of the second upper bound to the original network ***N***. $f_{max}^{N_r}$ satisfies the flow constraints of the original network for all the nodes except those connected to the sink. The maximum feasible flow at every second last node i (i.e. the nodes directly connected to the destinations $k \in L$) is $f_i^{N_r}$. In order to satisfy all capacity bounds we dispatch $min(f_i^{N_r}, u_{ik})$ units of flow from second last node i toward destination k for all $k \in L$. This is explained in Procedure H2.

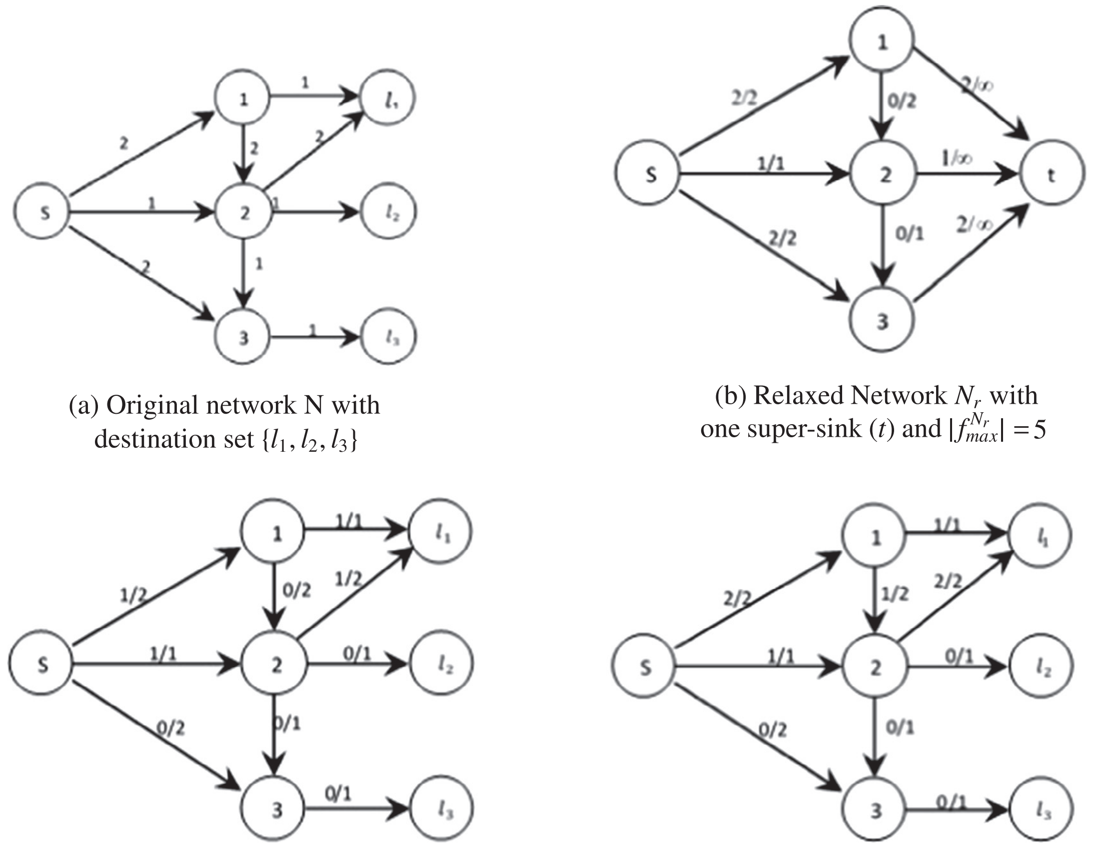
It is possible that a second last node is connected to several sinks, Procedure H2 will not work for choosing several locations because in step 5 of Procedure H2 we are again confronted with a multiple destination ELP. In this case, at every second node i it is not clear which k is considered for fulfilling the capacity bounds, as we have to consider a p-tuple combination of locations. So Procedure H2 is only proposed for ELP with $p = 1$.

While this heuristic gives a feasible solution, the result may not always be optimal. In networks that the second last nodes are connected, this solution is suboptimal, because the heuristic neglects the augmenting paths which go from one second last node through another one for reaching the sink. As a counterexample for the optimality of Procedure H2 for ELP see Fig. 4.

Procedure H2. Heuristic Solution for **ELP**.

Input: Network $N = (V, A, u), L \subseteq V, |L| = q$, Number of destinations to be selected: $p = 1$
Output: $Maxflow, BestLoc$

1 $f_{max}^{N_r} = \mathbf{UB2}(N = (V, A, u, L, p))$
2 $f_{max}^{N_r} := \sum_{i\in N: l\in L \wedge (i,l)\in A} f_i^{N_r}$
3 $f(i,l) \leftarrow 0 \quad \forall i \in V$ and $l \in L$
4 $f_l := \sum_{i:(i,l)\in A} f(i,l)$
5 $f_l \leftarrow 0 \quad \forall l \in L$
6 **for** $l = 1$: q **do**
7   **for** i = 1:n **do**
8   **if** $((i,l) \in A)$ **and** $(f_i^{N_r} \neq 0)$ **then**
9       $|f(i,l)| \leftarrow min(u_{il}, |f_i^{N_r}|)$
10       $|f_i^{N_r}| \leftarrow |f_i^{N_r}| - |f(i,l)|$
11   **end if**
12   **end for**
13 **end for**
14 **for** $i = 1$: n **do**
15   **while** $f_i^{N_r} > 0$ **do**
      augment path (i-s)
16   **end while**
17 **end for**



(a) Original network N with destination set $\{l_1, l_2, l_3\}$

(b) Relaxed Network N_r with one super-sink (t) and $|f_{max}^{N_r}| = 5$

(c) Maximal-flow and best location for network N, applying H2, with $Opt\_flow_{Opt\_Loc} = |f_{l_1}| = 2$

(d) Maximal-flow and best location for network N, applying E1, with $Opt\_flow_{Opt\_Loc} = |f_{l_1}| = 3$

Fig. 4. Counterexample demonstrating that H2 may not always result in optimum solution.

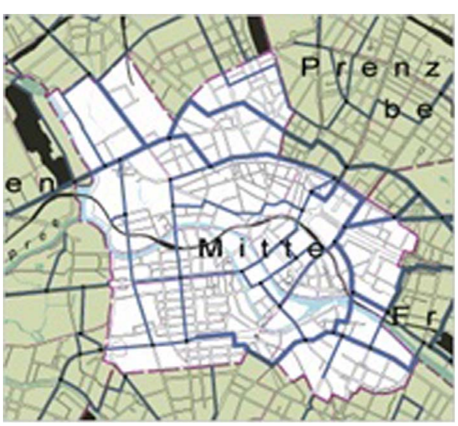


Fig. 5. Map of Mitte-center Berlin.
Photo credit:http://www.stadtentwicklung.berlin.de.

Table 2
Computation times(s) of solution algorithms for Mitte-center Berlin (p = 5).

n	UB1	UB2	E1	E2	H1
10	0.006	0.006	0.322	0.278	0.019
15	0.009	0.011	0.741	0.756	0.009
20	0.016	0.02	1.394	1.313	0.016
25	0.006	0.01	2.169	1.834	0.013
Ave.	0.009	0.012	1.157	1.045	0.014

Table 3
Gap to optimality of solution algorithms for Mitte-center Berlin (p = 5).

q	%GapUB1	%GapUB2	%GapH1
10	10	38.9	0
15	55	97.5	0
20	105.23	119.02	0
25	120.54	184.2	0
Ave.	72.693	109.905	0

```
Update f_l
Maxflow_BestLoc = max(f_l), l ∈ L
return Maxflow, BestLoc
```

The original network N is represented in Fig. 4a. The objective is to choose the location which maximizes the network flow, satisfying the capacities shown by numbers on the edges. The relaxed network is developed in Fig. 4b. According to Algorithm UB2 for which the $|f_{max}^{N_r}| = 5$ units of flow and $|f_1^{N_r}| = 2$ for the path $(s{-}1{-}t)$, $|f_2^{N_r}| = 1$ for the path $(s{-}2{-}t)$ and $f_3^{N_r} = 2$ for the path $(s{-}3{-}t)$. We continue implementing Procedure H2 for candidate destinations $\{l_1, l_2, l_3\}$; for example for l_1 we have:

for node 1: $u_{1l_1} = 1, |f_1^{N_r}| = 2 \Rightarrow |f(1,l_1)| = min(1,2) \Rightarrow |f_1^{N_r}| = 1.$
for node 2: $u_{1l_2} = 1, |f_1^{N_r}| = 1 \Rightarrow |f(2,l_1)| = min(2,1) \Rightarrow |f_2^{N_r}| = 0.$

At last, $f_1^{N_r} = 1$ and no augmenting path exists toward the destinations, we have to augment the excess flow on node 1 toward node s.

Table 4
Computation times(s) of solution algorithms for ELP ($p = 0.2q$).

	p = 0.2q																					
	UB1				UB2				E1				E2				H1				Cplex*	
n	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD
												q = 10										
100	0.01	0.01	0.04	0.01	0.01	0.01	0.09	0.00	0.40	0.20	0.54	0.18	0.01	0.20	1.01	0.00	0.02	0.00	1.72	0.00	1.20	0.02
200	0.01	0.01	0.03	0.00	0.05	0.01	0.10	0.00	2.18	0.27	3.71	0.92	0.05	0.02	0.09	0.00	0.09	0.01	0.15	0.02	1.87	0.52
300	0.01	0.01	0.08	0.01	0.14	0.01	0.23	0.02	6.33	0.75	11.08	1.05	0.15	0.04	0.34	0.02	0.28	0.02	0.58	0.09	5.61	1.33
400	0.02	0.01	0.04	0.00	0.42	0.02	0.63	0.06	18.93	1.65	28.00	3.41	0.42	0.15	0.82	0.01	0.84	0.03	1.39	0.23	25.32	4.12
500	0.04	0.02	0.06	0.02	1.19	0.02	2.50	0.90	54.15	3.63	73.50	12.85	1.21	0.07	2.31	0.08	2.40	0.07	3.93	1.08	55.01	3.77
												q = 15										
100	0.01	0.00	0.04	0.00	0.01	0.01	0.11	0.01	3.74	0.80	5.54	1.71	0.11	1.51	13.23	0.29	0.02	0.00	0.03	0.01	11.35	3.01
200	0.01	0.02	0.03	0.00	0.04	0.01	0.12	0.03	21.80	1.22	38.06	8.97	2.12	0.18	3.88	0.04	0.09	0.00	0.24	0.00	58.65	4.65
300	0.01	0.00	0.03	0.01	0.13	0.03	0.28	0.04	62.29	6.00	113.68	10.27	6.75	0.39	17.71	0.98	0.27	0.02	0.76	0.05	–	–
400	0.38	0.03	0.43	0.09	0.41	0.08	0.76	0.12	190.57	13.20	287.28	40.08	26.41	1.86	35.34	1.32	0.83	0.77	1.74	0.02	–	–
500	1.98	0.02	2.01	0.07	2.18	0.92	3.30	0.46	893.28	29.87	974.61	163.89	213.29	1.00	226.61	3.59	0.89	0.70	1.67	0.19	–	–
												q = 20										
100	0.09	0.00	0.12	0.03	0.01	0.00	0.02	0.00	38.83	0.97	56.51	12.36	4.05	0.53	8.31	1.45	0.01	0.00	0.02	0.00	175.00	23.13
200	0.11	0.01	0.25	0.06	0.05	0.01	0.21	0.02	231.00	21.94	619.06	128.01	55.02	2.61	72.32	7.63	0.10	0.00	0.18	0.01	854.05	18.94
300	0.09	0.01	0.21	0.05	0.13	0.01	0.51	0.09	669.11	12.71	740.03	253.17	211.59	1.99	397.88	8.73	0.27	0.02	0.58	0.01	–	–
400	0.08	0.01	0.11	0.03	0.41	0.05	0.91	0.02	2023.38	131.52	3710.16	1750.35	835.54	4.37	1867.32	34.07	0.83	0.85	1.35	0.23	–	–
500	0.09	0.01	0.27	0.04	0.84	0.03	1.91	0.07	4021.08	100.53	5851.68	2883.07	1983.40	25.66	2301.03	303.19	0.83	0.78	1.28	0.33	–	–
												q = 25										
100	0.07	0.03	0.38	0.03	0.01	0.00	0.01	0.00	444.73	23.04	738.00	358.00	70.10	3.05	86.23	30.05	0.02	0.02	0.23	0.01	–	–
200	0.21	0.01	0.51	0.08	0.05	0.01	0.06	0.03	2530.86	16.28	2568.31	2501.54	1330.48	74.87	2559.80	94.49	0.10	0.01	0.12	0.08	–	–
300	0.18	0.09	0.29	0.09	0.13	0.02	0.16	0.09	6735.46	477.80	7346.70	5838.34	4866.50	178.00	5674.00	255.44	0.25	0.03	0.31	0.20	–	–
400	1.09	0.07	2.21	0.07	0.41	0.01	0.42	0.41	22184.42	64.39	22353.30	22048.52	17099.07	5346.16	22263.73	6449.41	0.83	0.01	0.84	0.81	–	–
500	3.18	0.13	4.66	1.07	1.25	0.72	2.15	0.73	38252.35	125.80	41359.02	18257.33	20135.03	1232.38	21322.08	6987.35	3.65	1.32	5.08	2.01	–	–

* Cplex run-time was set to one hour. We have not reported the computation time if the optimum was not reached in 1 h.

Table 5
Computation times(s) of solution algorithms for ELP ($p = 0.4q$).

	p = 0.4q																					
	UB1				UB2				E1				E2				H1				Cplex	
n	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD
												q = 10										
100	0.01	0.01	0.02	0.00	0.02	0.01	0.09	0.01	5.05	0.03	6.57	4.78	0.03	0.04	0.08	0.01	0.01	0.01	0.08	0.00	3.78	1.50
200	0.01	0.01	0.03	0.00	0.04	0.01	0.10	0.02	9.07	1.07	11.27	8.00	0.05	0.02	0.08	0.04	0.09	0.03	0.17	0.02	12.67	3.27
300	0.02	0.01	0.04	0.01	0.15	0.05	0.21	0.09	31.58	5.09	39.52	29.18	0.31	0.14	0.76	0.09	0.30	0.09	0.58	0.09	67.80	9.71
400	0.06	0.02	0.07	0.02	0.64	0.03	0.94	0.11	135.31	17.24	167.35	125.06	1.34	0.83	1.97	0.91	1.28	0.10	1.88	0.63	–	–
500	0.57	0.10	0.93	0.08	1.09	0.22	1.73	0.38	256.47	18.61	304.51	224.98	13.25	10.18	31.54	9.21	1.35	0.22	1.75	0.75	–	–
												q = 15										
100	0.02	0.01	0.04	0.01	0.01	0.01	0.09	0.01	135.25	13.24	174.11	59.63	24.37	14.30	41.28	17.65	0.06	0.02	0.10	0.04	27.94	2.03
200	0.08	0.03	0.09	0.03	0.04	0.02	0.10	0.02	223.69	18.10	286.41	97.27	45.28	11.87	56.38	33.17	0.09	0.02	0.11	0.03	95.63	13.11
300	0.23	0.07	0.61	0.11	0.15	0.09	0.21	0.05	795.58	71.78	1012.51	346.63	261.38	47.09	301.27	175.03	0.31	0.09	0.49	0.11	–*	–
400	0.98	0.11	1.09	0.03	0.60	0.09	0.98	0.30	3079.50	274.26	3912.31	2465.68	1315.38	185.40	1630.21	1103.44	1.21	0.09	1.98	0.67	–	–
500	2.94	0.97	4.97	2.08	2.01	0.11	3.78	1.17	5024.00	12576.36	6381.18	3723.56	2980.61	12023.00	3189.74	2421.30	4.03	0.51	5.01	3.18	–	–
												q = 20										
100	0.04	0.01	0.09	0.02	0.05	0.01	0.08	0.03	357.64	78.12	524.32	295.65	124.87	72.86	167.08	98.24	0.05	0.01	0.01	0.02	384.65	33.57
200	0.07	0.03	0.10	0.05	0.09	0.02	0.34	0.05	685.10	65.45	901.36	586.24	468.64	68.06	748.67	412.34	0.05	0.01	0.03	0.02	786.34	95.13
300	0.97	0.07	1.13	0.32	0.15	0.40	0.31	0.06	1268.25	124.13	1784.31	857.64	877.94	114.04	1023.35	706.34	0.09	0.01	0.12	0.03	–	–
400	1.72	0.23	2.13	0.84	0.78	0.12	0.97	0.21	5846.34	387.25	67540.21	4990.11	2358.45	321.09	21044.30	1467.32	0.12	0.08	0.34	0.03	–	–
500	3.78	1.08	4.81	2.08	1.98	0.31	2.15	0.27	7695.08	1247.05	8124.70	7032.84	5121.32	945.11	73864.65	4786.30	1.67	0.17	2.08	0.70	–	–
												q = 25										
100	0.05	0.02	0.08	0.02	0.10	0.01	0.21	0.03	896.50	38.60	1023.50	537.90	783.18	32.96	876.11	524.10	0.09	0.02	0.18	0.03	2054.80	203.10
200	0.13	0.03	0.33	0.08	0.09	0.02	0.16	0.03	3036.25	426.52	4050.37	1821.75	2595.54	364.24	3067.54	1823.05	0.16	0.09	0.41	0.09	–	–
300	0.15	0.04	0.27	0.07	0.24	0.07	0.61	0.11	7940.35	1824.51	10589.17	4764.21	6749.32	3382.64	7603.11	5476.02	0.49	0.09	0.79	0.12	–	–
400	2.09	0.68	3.71	1.51	0.94	0.08	1.18	0.09	23287.09	7831.28	31051.49	13972.25	19748.01	5904.78	21254.31	10254.35	1.94	0.10	2.08	0.33	–	–
500	4.33	0.93	4.65	2.99	2.09	0.10	3.20	0.98	32654.08	8144.48	43540.81	19592.45	27681.85	6955.38	29145.28	18079.35	3.08	0.71	5.08	1.09	–	–

* Cplex run-time was set to one hour. We have not reported the computation time if the optimum was not reached in 1 h.

Table 6
Computation times(s) of solution algorithms for ELP ($p = 0.8q$).

	p = 0.8q																					
	UB1				UB2				E1				E2				H1				Cplex*	
n	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD
												q = 10										
100	0.01	0.01	0.06	0.01	0.01	0.01	0.04	0.01	0.25	0.21	0.75	0.18	0.01	0.01	0.27	0.01	0.01	0.01	0.05	0.01	2.65	0.54
200	0.01	0.02	0.04	0.01	0.04	0.01	0.11	0.01	1.71	0.03	2.31	0.91	0.06	0.02	0.09	0.01	0.08	0.02	0.10	0.02	3.11	0.60
300	0.02	0.01	0.04	0.01	0.12	0.01	0.32	0.04	5.11	1.05	6.30	2.61	0.11	0.31	1.05	0.03	0.23	0.09	0.41	0.07	7.06	1.08
400	0.02	0.01	0.08	0.01	0.22	0.03	0.57	0.09	9.93	0.16	11.03	8.01	0.22	0.09	0.87	0.02	0.44	0.11	0.83	0.21	–	–
500	0.08	0.02	0.43	0.04	0.30	0.02	0.91	0.11	18.95	2.36	21.87	10.23	3.54	0.28	6.11	1.98	0.85	0.09	1.93	0.30	–	–
												q = 15										
100	0.06	0.02	0.13	0.02	0.01	0.01	0.07	0.01	2.47	1.67	3.25	1.25	1.41	0.02	2.01	0.50	0.01	0.01	0.08	0.01	28.65	2.09
200	0.03	0.05	0.09	0.01	0.04	0.01	0.13	0.01	16.99	4.45	25.09	7.11	10.93	1.36	13.69	5.36	0.07	0.01	0.23	0.01	86.62	11.80
300	0.02	0.02	0.08	0.01	0.11	0.05	0.28	0.03	50.98	9.74	80.05	41.49	37.38	6.05	41.68	19.87	0.22	0.05	0.71	0.03	–	–
400	0.07	0.02	0.11	0.03	0.25	0.05	0.30	0.01	112.11	36.48	168.85	75.07	105.42	10.32	118.36	73.32	0.49	0.06	0.59	0.07	–	–
500	0.11	0.09	0.29	0.06	0.35	0.07	0.75	0.90	706.35	73.14	1562.71	543.21	782.05	52.00	925.40	588.31	1.03	0.03	1.69	0.10	–	–
												q = 20										
100	0.12	0.07	0.32	0.08	0.01	0.01	0.05	0.01	24.98	18.06	40.22	11.61	23.76	16.41	35.99	7.56	0.01	0.01	0.08	0.01	384.45	52.09
200	0.09	0.06	0.11	0.03	0.04	0.01	0.06	0.01	176.65	63.94	221.51	65.82	176.97	58.06	254.40	96.03	0.07	0.02	0.16	0.02	–	–
300	1.09	0.10	1.89	0.27	0.11	0.03	0.28	0.07	558.35	128.35	708.02	383.83	533.62	137.47	804.04	387.64	0.21	0.03	0.00	0.09	–	–
400	2.34	0.31	3.09	0.95	0.24	0.01	0.27	0.00	1175.00	592.93	1630.39	694.45	1175.73	538.57	1692.01	899.01	0.49	0.03	0.93	0.12	–	–
500	1.06	0.28	1.99	0.34	0.56	0.21	1.05	0.18	10854.57	1198.09	12164.25	5024.74	9845.12	1088.41	15630.60	7254.32	0.95	0.10	1.38	0.05	–	–
												q = 25										
100	0.03	0.01	0.09	0.01	0.01	0.01	0.10	0.01	290.38	28.12	402.11	182.36	202.50	25.24	358.12	75.36	0.01	0.01	0.05	0.01	1897.31	65.35
200	0.14	0.02	0.23	0.06	0.04	0.01	0.09	0.01	1929.11	106.12	2214.38	883.65	1023.40	94.66	1854.60	754.25	0.07	0.01	0.04	0.01	–	–
300	0.38	0.02	0.71	0.11	0.12	0.02	0.16	0.09	6536.70	455.62	7078.03	5766.16	4562.50	227.00	6457.00	4984.30	0.24	0.02	0.28	0.20	–	–
400	1.24	0.03	2.37	0.98	0.25	0.04	0.34	0.19	13196.48	1957.31	16298.98	10309.25	12506.00	895.50	13874.00	8547.00	0.50	0.08	0.62	0.39	–	–
500	3.79	1.68	5.62	1.01	0.42	0.02	0.92	0.12	18243.00	2035.61	21345.00	15683.01	13524.11	1811.90	16320.01	7895.25	0.84	0.07	1.02	0.04	–	–

* Cplex run-time was set to one hour. We have not reported the computation time if the optimum was not reached in 1 h.

Table 7
Percentage of the gap to the optimal solution for ELP ($p = 0.2q$).

	p = 0.2q														
	UB1				UB2				H1				Cplex*	
n	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD
							q = 10							
multicolumn1l100	349.00	12.64	425.34	297.30	421.36	21.30	510.69	218.65	0.26	0.09	1.02	0.00	0.00	0.00
200	373.38	12.35	436.97	218.34	386.56	19.45	429.21	198.60	0.01	0.20	0.30	0.00	0.00	0.00
300	294.74	15.27	345.32	180.17	303.84	24.70	375.04	204.05	0.02	0.35	0.11	0.00	0.00	0.00
400	297.06	14.31	318.11	167.21	318.65	18.67	387.11	276.54	0.85	0.13	1.35	0.00	0.00	0.00
500	225.29	25.05	276.01	176.25	298.36	19.24	312.54	173.20	0.95	0.11	1.12	0.20	0.00	0.00
							q = 15							
100	311.54	13.65	347.31	217.62	398.45	24.75	463.12	287.64	0.93	0.10	1.18	0.00	0.00	0.00
200	235.01	14.64	297.11	194.35	305.98	15.13	394.41	206.64	1.70	0.17	2.01	0.00	0.00	0.00
300	173.25	16.01	231.00	104.33	239.57	28.15	301.12	176.34	1.49	1.34	2.19	0.07	56.47	26.75
400	184.20	11.34	191.36	121.54	226.05	22.12	268.94	111.38	1.83	1.02	3.20	0.10	83.87	19.12
500	103.74	12.00	137.16	78.06	195.23	21.89	263.60	99.21	2.09	1.34	2.98	0.18	93.25	13.05
							q = 20							
100	187.65	11.35	201.87	109.51	237.10	16.38	278.54	181.35	3.41	1.09	4.75	1.94	0.00	0.00
200	139.34	9.65	168.21	94.38	197.25	38.25	237.10	103.74	4.51	0.87	6.37	2.08	0.00	0.00
300	103.44	16.37	149.02	69.40	160.19	25.40	198.32	112.67	3.34	1.27	5.11	2.51	94.03	12.36
400	100.34	11.60	160.71	77.21	151.97	19.78	187.30	102.30	2.71	0.90	3.21	0.97	112.87	18.65
500	86.32	10.17	98.24	53.10	133.15	19.74	175.30	83.27	2.24	0.92	3.87	0.98	289.38	25.03
							q = 25							
100	120.12	23.58	184.10	95.06	138.14	24.18	150.31	98.35	4.65	0.66	6.50	1.50	184.36	25.67
200	87.64	27.20	100.65	63.74	111.37	10.68	126.37	89.81	5.26	1.06	6.30	1.25	248.07	18.74
300	83.80	19.37	112.35	65.14	104.85	11.25	131.20	73.41	4.31	1.11	5.78	0.97	327.12	28.60
400	69.35	20.41	103.70	48.39	104.41	9.87	127.09	81.73	3.73	1.05	4.39	0.90	394.27	26.78
500	57.61	18.37	78.31	43.11	91.20	13.84	108.47	71.25	2.98	0.97	4.09	0.08	473.08	28.43

* Cplex run-time was fixed to 1 h, for two trials.

Table 8
Percentage of the gap to the optimal solution for ELP ($p = 0.4q$).

	p = 0.4q														
	UB1				UB2				H1				Cplex*	
n	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD
							q = 10							
100	78.35	27.18	123.08	53.02	98.57	18.39	189.20	49.35	0.06	0.05	1.60	0.00	0.00	0.00
200	93.28	19.58	151.67	38.91	107.39	21.09	168.74	51.40	0.03	0.01	0.95	0.00	0.00	0.00
300	95.25	18.27	194.67	28.13	115.61	19.65	203.14	52.94	0.02	0.01	0.78	0.01	0.00	0.00
400	89.33	22.37	131.64	32.18	119.28	17.64	201.53	49.25	0.10	0.02	0.90	0.00	78.64	38.50
500	101.35	17.54	168.31	29.68	117.62	23.08	231.00	42.68	0.08	0.05	1.02	0.03	93.25	27.14
							q = 15							
100	67.67	18.09	127.58	37.45	89.60	16.85	201.31	35.07	1.02	0.35	2.04	0.08	0.00	0.00
200	75.79	21.14	217.34	28.11	71.22	20.64	241.30	34.22	1.40	0.25	3.00	0.00	0.00	0.00
300	61.47	12.57	110.68	42.09	71.53	17.00	137.40	23.70	1.30	0.91	1.99	0.06	112.17	36.80
400	65.36	19.47	141.01	28.97	71.64	21.30	267.91	34.08	1.08	0.30	2.53	0.10	132.40	22.05
500	72.00	12.35	108.30	33.25	74.52	18.57	197.03	52.10	0.82	0.41	1.84	0.09	158.36	18.69
							q = 20							
100	79.40	15.90	183.34	37.25	85.38	28.05	190.12	55.47	1.03	0.35	2.50	0.00	0.00	0.00
200	86.93	18.58	124.78	29.74	93.47	19.84	203.78	71.24	0.98	0.21	1.06	0.00	0.00	0.00
300	73.67	14.32	141.46	19.81	78.05	21.68	190.25	39.15	2.12	0.95	3.81	0.02	198.11	39.54
400	77.26	14.35	140.85	20.13	83.08	14.69	175.04	32.58	3.47	1.02	5.21	0.07	237.14	23.80
500	83.42	10.86	132.11	42.18	89.70	17.12	183.67	27.64	3.78	0.97	4.95	0.12	221.37	17.15
							q = 25							
100	101.26	23.47	168.35	41.01	113.95	19.50	210.64	53.47	3.08	2.19	16.58	0.00	0.00	0.00
200	84.19	19.48	195.60	38.15	94.32	25.38	144.54	39.67	5.26	3.29	14.47	0.60	178.11	23.75
300	76.21	13.70	123.41	26.60	92.34	13.98	187.20	39.18	4.15	2.08	7.91	0.00	192.30	16.58
400	96.91	11.58	148.11	46.28	108.95	18.36	164.30	58.27	6.58	3.73	9.19	0.44	219.11	18.15
500	94.30	13.22	175.14	32.08	105.95	20.47	198.70	43.81	7.10	2.18	12.87	1.11	235.60	16.12

* Cplex run-time was fixed to 1 h, for two trials.

Table 9
Percentage of the gap to the optimal solution for ELP ($p = 0.8q$).

	p = 0.8q													
	UB1				UB2				H1				Cplex*	
n	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD
							q = 10							
100	14.10	7.03	20.51	4.82	23.76	18.39	47.30	49.35	0.01	0.05	1.60	0.00	0.00	0.00
200	16.98	5.81	25.28	3.89	26.46	21.09	33.75	11.42	0.05	0.01	0.95	0.00	0.00	0.00
300	18.70	4.54	29.06	3.13	25.02	19.65	41.67	11.76	0.02	0.01	0.78	0.00	0.00	0.00
400	19.14	5.92	32.91	6.44	23.01	17.64	41.35	10.94	0.56	0.02	0.90	0.00	38.65	18.20
500	18.64	8.77	24.04	4.24	28.45	23.08	47.24	9.48	0.02	0.05	1.02	0.03	48.61	17.39
							q = 15							
100	3.39	2.81	7.42	0.44	7.89	4.55	16.92	35.07	0.26	0.35	2.04	0.08	0.00	0.00
200	3.20	2.32	8.94	0.41	7.59	5.58	20.28	34.22	0.15	0.25	3.00	0.00	0.00	0.00
300	3.14	1.82	7.64	0.40	7.49	4.59	11.55	23.70	0.26	0.91	1.99	0.06	88.79	32.02
400	0.61	1.55	2.21	0.08	3.45	5.76	22.51	34.08	0.04	0.30	2.53	0.10	64.85	19.18
500	1.40	1.78	4.30	0.18	4.72	5.02	16.56	52.10	0.04	0.41	1.84	0.09	93.15	16.26
							q = 20							
100	0.02	2.47	1.94	0.00	4.50	3.42	13.82	1.00	0.03	0.35	2.50	0.00	0.00	0.00
200	0.08	2.04	4.62	0.02	2.40	2.42	14.81	1.08	0.01	0.21	1.06	0.00	59.25	18.47
300	0.01	4.87	2.66	0.00	1.49	2.64	13.83	0.48	0.07	0.05	1.03	0.00	96.39	26.49
400	0.01	0.60	1.23	0.00	1.38	1.79	12.72	0.92	0.04	0.02	0.12	0.00	115.13	15.95
500	0.01	1.56	0.45	0.00	1.59	2.09	13.35	0.97	0.07	0.23	1.10	0.02	107.56	11.49
							q = 25							
100	0.01	2.16	6.60	0.00	3.41	4.56	9.81	0.32	0.01	2.19	3.88	0.00	0.00	0.00
200	0.01	1.84	6.23	0.00	1.31	4.53	6.73	0.30	0.01	1.03	3.39	0.60	103.60	20.66
300	0.01	1.30	3.80	0.01	0.40	2.50	8.30	0.31	0.23	0.91	1.85	0.00	114.83	14.42
400	0.03	1.07	3.98	0.00	0.29	2.91	6.28	0.09	0.44	1.27	2.15	0.00	149.29	15.79
500	0.01	1.23	4.03	0.00	0.50	2.35	5.96	0.02	0.52	0.87	1.21	1.11	136.95	14.02

* Cplex run-time was fixed to 1 h, for two trials.

Indeed, the flow vector on path ($s{-}1{-}l_1$) equals 1. As a result:

$$|f_{l_1}| = \sum_{i \in V} |f(i,l_1)| = 2, |f_{l_2}| = 0, |f_{l_3}| = 1$$

And hence $Maxflow_{Best\,Loc} = |f_{l_1}| = 2$. But it is shown in Fig. 4c. that the optimal flow equals 3 for optimum location l_1 for single destination ELP.

5. Computational results

5.1. Case Study: Mitte-center Berlin

In this section, first we test our algorithms, namely UB1, UB2, E1, E2, and H1 on a real world transportation network. In the next subsection; however we will implement and compare all of the algorithms on a range of problem groups. A set of networks from the Berlin area is provided in Bar-Gera (2005) and was used in Jahn, Ohring, Schulz, and As (2005). We have used one of the data sets for Mitte-center Berlin, which best fits the assumptions of our model, because it is a simple graph with constant integer arc capacities. Mitte-center district has an area of 39 sq.km, a map of this district can be seen in Fig. 5.

The data set consists of 398 nodes and 871 links with constant integer capacities. All solution methods are performed for all problem instances where $q = \{10,15,20,25\}$ destination locations are chosen randomly from the nodes. The number of destinations to be selected is set to 5. Tables 2 and 3 summarize the results.

Considering the performance and the computation time of the algorithms for this real case, they all run very fast due to the size of the problem. Though H1 is not necessarily an exact algorithm, it finds the optimum for all values of q. However, the density of the network is low (it is about 0.5%) compared to many real networks. The exact Algorithms E1 and E2 are also efficient for an evacuation-location plan for this real world network, regarding their computation times.

5.2. Experiment setup

We apply the algorithms introduced above in extensive numerical experiments. In this section the performance of the proposed upper bounds, heuristics and exact Algorithms UB1, UB2, E1, E2, and Procedures H1 and H2 is compared in terms of the gap to optimality and the computation time both for multiple and single destination ELP. The solutions as well as the upper bounds are calculated for randomly generated graphs. Random graphs with 100, 200, 300, 400 and 500 vertices with an edge density of 40% and with a maximum capacity of 1000 units are generated. The number of possible destinations is set to $|L| = q = \{10,15,20,25\}$. The possibility of each pair of nodes being connected is set to 0.4 (to reach the density of 40%). The capacities are also assigned from uniformly-distributed integers in $[0,1000]$. The resulting graphs are then tuned to guarantee the conditions of Definition 1. The tuning process includes the omission of self-loops, multiple edges between two nodes and the links between the sinks. The set of possible destinations for each of the instance problems is also generated randomly as a subset of V and four values are considered for the number of destinations to be selected as the ratios of q as well as the single destination case; i.e. $p = \{1,0.2q,0.4q,0.8q\}$. These were randomly generated by MATLAB. Each algorithm or heuristic is implemented for 30 different configurations of the networks for each of the problem categories. The most important statistics of the results are reported here; the average, the standard deviation, the minimum and the maximum gap to the optimal solution and the computation time of the iterated implementation for 30 different configurations.[3]

All algorithms were implemented on an Intel® Core™ i7 processor with 4.00 GB of DDR RAM in the environment of 64-bit Microsoft Windows 10 and were coded in MATLAB R2011b.

[3] All instances can be downloaded here: github.com/MionaFarm/Random-Evacuation-Networks.

Table 10
Computation times(s) of solution algorithms for ELP ($p = 1$).

	p = 1																									
	UB1				UB2				E1				E2				H1				H2				Cplex*	
n	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD
													q = 10													
100	0.01	0.02	0.32	0.00	0.01	0.02	0.30	0.00	0.04	0.02	0.26	0.00	0.02	0.05	0.84	0.01	12.51	9.27	47.70	1.09	0.00	0.02	0.25	0.00	23.31	16.70
200	0.06	0.02	0.45	0.04	0.03	0.05	0.37	0.00	0.29	0.05	1.69	0.08	0.08	0.26	1.26	0.04	19.83	14.69	119.89	1.73	0.02	0.05	0.62	0.00	38.64	11.27
300	0.19	0.14	0.77	0.10	0.11	0.07	0.23	0.01	0.88	0.06	1.59	0.10	0.21	0.71	2.16	0.11	148.36	109.90	2516.17	12.93	0.05	0.21	2.59	0.00	163.23	34.06
400	0.75	0.20	2.18	0.39	0.64	0.10	1.05	0.06	1.78	0.09	5.75	0.35	0.74	0.93	2.60	0.14	233.06	172.64	6209.09	38.36	0.11	0.13	1.60	0.03	357.36	29.11
500	1.99	0.31	4.59	1.03	1.24	0.21	2.61	0.12	8.92	0.15	24.97	2.03	1.97	2.64	6.01	0.39	409.00	302.97	19122.47	86.56	0.78	0.08	1.79	0.02	879.11	33.18
													q = 15													
100	0.01	0.01	0.04	0.00	0.01	0.01	0.06	0.00	0.06	0.02	0.60	0.02	0.02	0.06	0.85	0.01	19.90	14.74	94.49	2.83	0.00	0.03	0.36	0.00	101.36	21.18
200	0.06	0.02	0.19	0.01	0.03	0.05	0.32	0.01	0.42	0.12	1.77	0.13	0.09	0.24	1.22	0.04	27.78	11.82	60.78	3.81	0.04	0.11	0.10	0.00	347.11	34.08
300	0.19	0.02	0.61	0.02	0.11	0.04	1.03	0.0.7	1.24	0.38	4.45	0.38	0.23	0.65	2.04	0.03	150.60	111.56	1975.39	19.17	0.13	0.22	0.39	0.02	464.21	49.39
400	0.71	0.03	2.34	0.07	0.61	0.05	2.00	0.10	2.26	0.76	7.75	0.70	0.72	1.33	2.91	0.11	743.55	401.92	24110.27	93.28	0.11	0.27	0.27	0.01	610.03	76.11
500	1.94	0.19	6.39	0.21	1.21	0.81	4.05	0.63	13.01	3.81	42.70	3.06	2.00	2.86	6.06	0.13	598.52	443.35	29337.32	75.16	0.82	0.17	2.05	0.09	1241.34	173.28
													q = 20													
100	0.01	0.01	0.02	0.00	0.01	0.01	0.02	0.00	0.08	0.03	0.21	0.02	0.02	0.12	0.81	0.01	101.64	21.17	143.14	86.47	0.00	0.01	0.89	0.00	539.21	98.60
200	0.06	0.70	1.41	0.13	0.04	0.02	0.04	0.01	0.53	0.13	1.00	0.11	0.09	0.09	0.65	0.00	261.27	54.43	800.73	79.00	0.04	0.01	0.89	0.00	876.25	87.90
300	0.19	0.11	0.24	0.02	0.11	0.10	0.20	0.03	1.61	0.39	2.87	0.09	0.23	0.07	1.31	0.11	168.60	35.13	456.96	43.28	0.16	0.20	0.91	0.30	1634.08	987.00
400	0.77	0.21	0.50	0.04	0.53	0.21	0.45	0.09	5.61	0.77	6.03	31.00	0.79	0.03	0.42	0.10	357.60	74.50	2055.65	137.19	0.23	0.31	1.33	0.06	–	–
500	1.46	0.90	1.95	0.17	0.98	0.87	1.74	0.09	13.86	2.65	20.06	2.67	3.75	0.78	6.39	0.75	612.00	127.50	6020.83	266.13	1.11	0.73	1.94	0.05	–	–
													q = 25													
100	0.01	0.03	0.11	0.00	0.01	0.20	0.38	0.00	0.11	0.07	0.61	0.03	0.03	0.20	1.38	0.01	141.14	38.23	207.60	12.14	0.00	0.03	0.04	0.00	–	–
200	0.06	0.07	0.31	0.02	0.04	0.03	1.01	0.01	0.66	0.13	1.69	0.09	0.10	0.23	1.60	0.03	701.21	146.09	987.53	46.38	0.02	0.01	0.04	0.00	–	–
300	0.21	0.22	0.96	0.08	0.12	0.05	1.17	0.03	2.24	1.23	11.16	0.49	0.29	0.34	2.38	0.09	833.54	173.65	1173.90	110.26	0.08	0.09	0.19	0.03	–	–
400	1.27	0.65	2.86	0.11	1.24	0.09	2.45	0.07	7.11	1.28	17.35	1.36	1.63	0.98	5.41	0.02	879.35	183.20	1238.42	118.17	0.19	0.73	0.96	0.09	–	–
500	3.78	1.27	5.69	0.79	2.68	0.65	6.13	0.28	18.34	2.39	38.48	3.87	4.01	3.21	17.53	0.07	1024.36	213.41	16867.82	219.00	0.98	0.66	1.84	0.02	–	–

* Cplex run-time was set to one hour. We have not reported the computation time if the optimum was not reached in 1 h.

Table 11
Percentage of the gap to the optimal solution for ELP ($p = 1$).

	UB1				UB2				H1				H2				Cplex*	
n	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD	Max.	Min.	Ave.	SD
									q = 10									
100	51.95	8.25	79.57	10.34	550.78	65.20	714.91	69.40	0.29	0.09	1.02	0.01	0.30	0.44	1.61	0.01	0.0	0.0
200	44.30	7.23	68.51	9.32	624.67	79.38	810.82	81.17	0.10	0.20	1.30	0.01	0.15	0.23	0.83	0.03	0.00	0.00
300	40.89	6.35	62.17	8.45	689.37	76.10	894.80	91.47	0.30	0.35	2.11	0.00	0.01	0.08	0.26	0.00	0.00	0.00
400	37.64	6.70	60.08	8.79	672.31	73.57	876.76	78.76	0.85	0.13	1.35	0.00	0.03	0.04	0.16	0.00	0.00	0.00
500	36.89	6.24	57.80	11.21	697.00	65.79	908.21	97.69	1.09	0.41	8.62	0.20	0.12	0.24	0.84	0.04	0.00	0.00
									q = 15									
100	59.67	9.28	81.47	11.37	619.97	75.45	804.72	80.42	6.60	0.45	8.24	2.03	12.15	4.73	17.36	2.45	0.00	0.00
200	53.26	8.42	73.05	10.52	775.13	96.91	1006.12	105.13	12.06	0.74	14.03	3.73	8.51	2.98	11.79	0.89	0.00	0.00
300	47.41	6.90	63.62	11.38	809.56	101.67	1050.81	143.50	10.56	5.54	15.29	2.31	8.60	3.03	11.93	1.28	0.00	0.00
400	47.35	7.63	65.29	7.38	829.48	89.64	1076.66	113.78	12.99	4.22	22.34	2.43	8.63	3.04	11.97	1.21	0.00	0.00
500	47.79	7.69	65.87	9.79	836.83	91.38	1086.21	114.95	14.83	5.54	20.80	2.75	6.38	1.96	8.54	0.98	0.00	0.00
									q = 20									
100	61.86	8.60	82.06	10.69	646.27	79.09	867.40	145.01	24.22	4.51	33.16	7.83	17.71	6.40	24.75	2.48	0.00	0.00
200	55.27	7.82	73.66	9.92	796.20	99.82	1080.30	168.88	32.02	3.61	44.46	8.39	17.05	6.08	23.74	0.97	0.00	0.00
300	54.27	7.70	72.38	10.02	802.12	87.16	1088.71	169.83	23.71	5.25	35.67	10.12	15.96	5.56	22.08	1.29	0.00	0.00
400	50.65	7.47	68.20	9.57	824.40	89.03	1120.35	173.37	19.22	3.73	22.41	3.93	14.43	4.82	19.73	1.28	53.06	23.47
500	51.43	6.37	66.40	8.47	841.60	106.10	1144.78	176.11	15.89	3.81	27.01	3.97	16.55	5.84	22.98	1.00	67.34	34.25
									q = 25									
100	63.98	11.16	90.21	22.68	699.72	86.48	843.31	153.52	14.39	3.26	40.82	1.56	19.90	7.45	22.51	2.73	48.18	12.38
200	57.09	10.10	80.83	22.15	796.50	99.87	980.74	168.93	16.25	4.98	39.56	1.31	21.51	8.22	24.38	1.07	52.51	13.69
300	55.39	9.84	78.52	19.64	840.29	105.92	1042.91	175.90	13.33	5.19	36.30	1.02	18.36	6.71	20.71	2.15	60.24	22.31
400	54.36	9.68	77.12	21.94	841.86	98.20	1045.14	176.15	11.55	6.47	32.50	0.95	20.43	7.71	23.12	1.41	72.50	13.25
500	55.78	9.90	79.05	22.05	854.36	107.87	1062.89	178.14	9.22	4.59	25.69	0.11	20.51	7.74	23.22	1.37	77.31	12.35

* Cplex run-time was fixed to 1 h, for two trials.

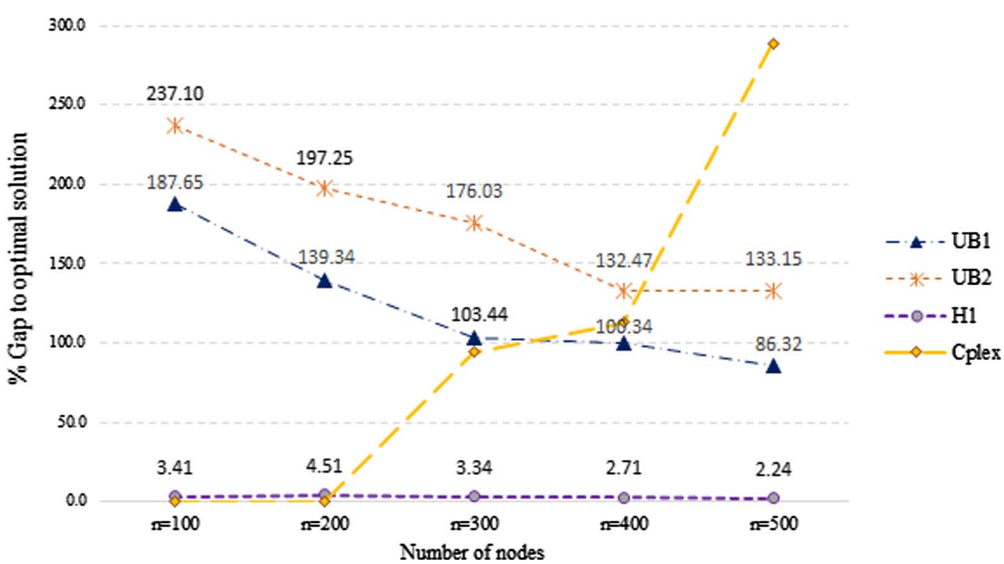


Fig. 6. Comparative average performance of the algorithms for $q = 15$ and $p = 0.2q$.

Since the MILP formulation of the problem in **Model1** is at hand we also tried to solve the model by IBM_ILOG_CPLEX 12.3 (mathematical solver), however due to the size and complexity of the problem the run time of the solver was fixed to one hour for two trials.

5.3. Numerical results

The numerical results are represented in different categories according to the problem parameters. The computation times are reported in Tables 4–6 for $p = 0.2q$, $p = 0.4q$ and $p = 0.8q$ respectively. The performance of the aforementioned algorithms is reported in Tables 7–9 for all values of p. For the single destination ELP the same results are represented in Tables 10 and 11. Tables 10 and 11 also include the results for H2, as this heuristic can only be applied for p = 1. In all tables n is the number of nodes and q is the number of possible destinations for the ELP.

For the upper bounds, the gap represents the percentage above the optimal solution. For all tested instances the average computation time of Algorithm E1 is clearly longer than the computation times of Procedure H1 and both upper bounds. This is not a counter-intuitive result, since Algorithm E1 always iterates through all possible combinations of locations $\binom{q}{p}$ and finds the maximum flow vector. Algorithm E1; however, always runs in a shorter time than Cplex, keeping in mind that the latter even fails to find the optimal for most instances in one hour.

The other exact algorithm, E2, is able to find the optimum in less time than Algorithm E1 for these random instances, yet the difference

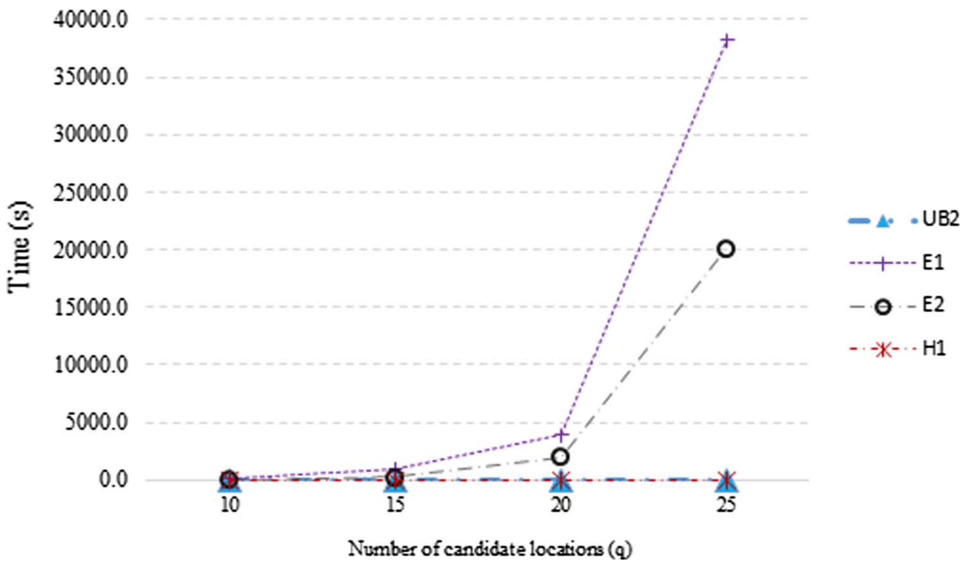


Fig. 7. Comparison of the running times of the proposed algorithms for different values of q.

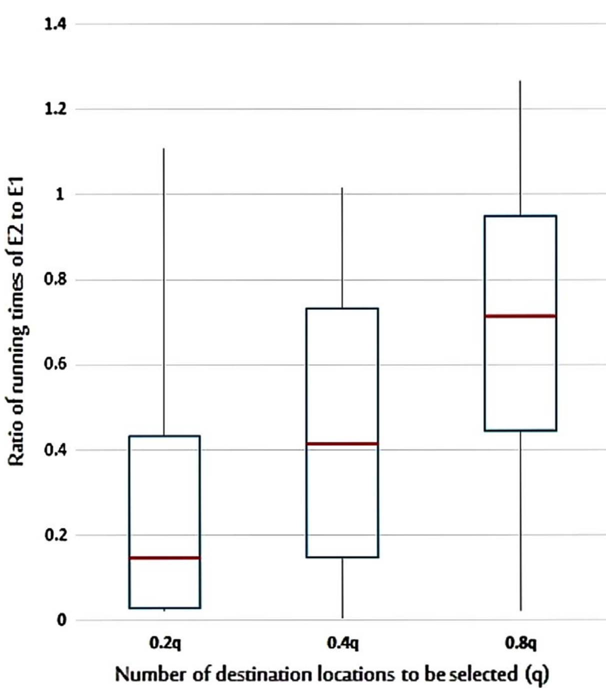


Fig. 8. Boxplot for comparison of E2 and E1 computation time.

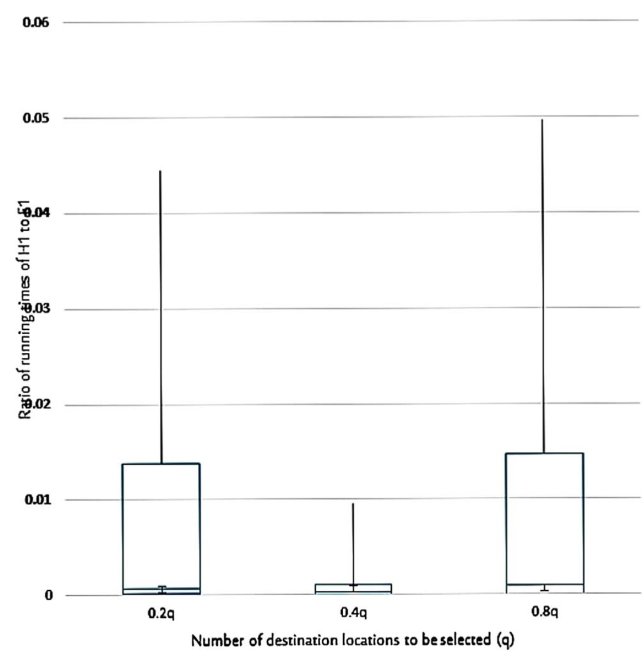


Fig. 9. Boxplot for comparison of H1 and E1 computation time.

between the average of its computation time and the one for the exact Algorithm E1 varies for different values of p and q (Tables 4–6). Upper bounds Algorithms UB1 and UB2 both run in very shorter times, compared to E1 and E2. Anyhow, in general, the closer q/p is to 1, (e.g. see the %GapUB1 for $q = 10$ and $p = 8$ in Table 6) the less the gap to the optimal solution. Where $q/p = 1$, the problem reduces to the general multi-terminal max-flow problem.

Overall, Heuristic H1 outperforms the exact solution algorithms in terms of computation time and at the same time its average gap to optimality stays below 7.1%, for all problem instances (the maximum average gap to optimality of H1 is equal to 16.58%, for a network with 500 nodes and 25 candidate destination locations). It even succeeds in finding the optimum for numerous instances, as can be seen in Table 4 for instances with zero gap to optimality. The comparative average performance of the algorithms is represented in Fig. 6 as an illustrative example for $q = 15$ and $p = 0.2q$. As can be seen in Fig. 7, while the running time of Algorithms E1 and E2 grows with a steep slope for bigger qs in our random instances, the running time of Procedure H1 follows a steady running time trend with a slightly greater than zero gap to the optimum.

The distribution of the ratio of computation time of Algorithm E2 to the computation time of Algorithm E1 should desirably be as small as possible, because a less distributed ratio shows a more uniform comparative behavior of Algorithms E2 and E1 for different values of p. Besides, smaller ratios indicate better computation times for Algorithm E2, compared to Algorithm E1. Fig. 8 is a box plot that shows an uptrend of this ratio with the increase of p (the number of locations to be selected). On the other hand, according to the experiments performed in this paper, H1 is able to find feasible solutions very fast while its gap to optimal solution reduces from 7.1 for $p = 0.4q$ to 0.52 for $p = 0.8q$ for q = 15 and n = 500 (see Tables 8 and 9) and at the same time the distribution of the ratios of the running time of the Procedure H1 to that of Algorithm E1, stays very close to zero meaning that Procedure H1 is running efficiently for our random instances (see Fig. 9).

The computational results for the single destination (p = 1) ELP are

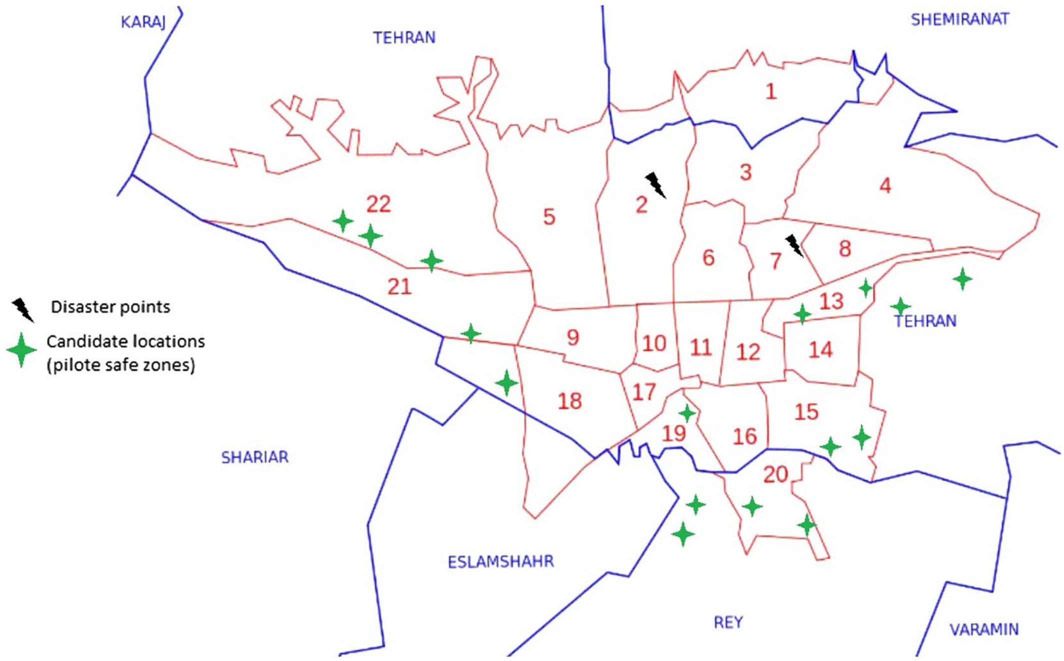


Fig. 10. Map of municipal districts of Tehran, the candidate safe zones and the potential disaster points.

Table 12
Interval of capacity-decrease and spread-increase parameters under different grades of disasters.

	Area I	Area II	Area III	Area IV
Disaster grade 0	$\alpha = 1, \beta = 0$	$\alpha = 1, \beta = 0$	$\alpha = 1, \beta = 0$	$\alpha = 1, \beta = 0$
Disaster grade 1	$\alpha \in (0.9,1.0), \beta \in (0.00,0.05)$	$\alpha = 1, \beta = 0$	$\alpha = 1, \beta = 0$	$\alpha = 1, \beta = 0$
Disaster grade 2	$\alpha \in (0.8,0.9), \beta \in (0.05,0.10)$	$\alpha \in (0.9,1.0), \beta \in (0.00,0.05)$	$\alpha = 1, \beta = 0$	$\alpha = 1, \beta = 0$
Disaster grade 3	$\alpha \in (0.7,0.8), \beta \in (0.10,0.15)$	$\alpha \in (0.8,0.9), \beta \in (0.05,0.10)$	$\alpha \in (0.9,1.0), \beta \in (0.00,0.05)$	$\alpha = 1, \beta = 0$
Disaster grade 4	$\alpha \in (0.6,0.7), \beta \in (0.15,0.20)$	$\alpha \in (0.7,0.8), \beta \in (0.10,0.15)$	$\alpha \in (0.8,0.9), \beta \in (0.05,0.10)$	$\alpha \in (0.9,1.0), \beta \in (0.00,0.05)$

Table 13
The results of implementing Algorithm E1 for Tehran network.

Disaster severity grade	Running-time for Algorithm E1	Total no. of evacuees	Selected destinations
Grade 0	22.015625	1,049,721	[1,5,9,12,15]
Grade 1	20.015625	1,002,075	[1,5,9,8,15]
Grade 2	19.390625	821,047	[1,2,5,9,10]
Grade 3	19.078125	551,678	[1,2,3,4,5]
Grade 4	18.734375	295,337	[10,2,3,4,5]

presented in Tables 10 and 11. It is noticeable from Table 10 that though all of the solutions run greatly faster in the single destination case, E1 and Cplex are still dominated by the other algorithms in terms of the computation time.

The gap to optimality of each of the proposed solution methods for the single destination-location problem is reported in Table 11. As can be seen from the results, in contrast to the results for the multiple destination ELP, the efficient computation time of the exact algorithms makes it justified to apply them for larger networks. The reason behind the huge gap of upper bounds UB1 and UB2 is that they relax the constraint of choosing only one destination. Nonetheless, they can be considered as the first attempt for finding the optimum, due to their efficient computation time.

Furthermore, T-tests were performed to examine the significance of differences in performance or computation time of the proposed algorithms. The results of the tests confirm that Algorithm E2 is significantly faster than Algorithm E1 (*p-value* = 0.013). Therefore, Algorithm E2 can be preferred above Algorithm E1. On the other hand, there is no significant difference between Algorithms UB1 and UB2 in terms of computation time (*p-value* = 0.13), but as Algorithm UB1 is by definition tighter than Algorithm UB2 (see Section 4.1), it is preferred above Algorithm UB2.

For the special case of p = 1 comparisons based on the T-test were conducted for Algorithms H1 and H2 at the significance level of $\alpha = 0.05$. The test result confirms that Procedure H2 is significantly faster than Procedure H1 with *p-value* = 0.00. However, Procedure H1 is significantly better than Procedure H2 in terms of the gap to optimality with p-value = 0.00. Therefore, whether to use Algorithms E2, H1 or H2, depends on the preference of the user in terms of the balance between computation time and solution quality.

5.4. Further discussion

One of the factors affecting the availability of the safe locations during an evacuation process is the severity and spread of the disasters. Finally, to further demonstrate the value of incorporating the locational decisions during the evacuation process, we consider a real case with two different sets of decrease and increase parameters α_{ij} and β_{ij} to reflect the extent of the disaster. Decrease parameter α_{ij} can reflect the instantaneous influence of disasters on the capacities of the arcs, a smaller α_{ij} means a greater decrease in the capacity of the arc. Increase parameter β_{ij} can reflect the influence of the disasters on the number of the corrupted arcs, a larger β_{ij} means more arcs or routes have been proposed to destruction. Therefore, different sets of α_{ij} and β_{ij} can reflect the impacts of disasters in different areas.

We consider Tehran municipal metropolis, the capital of Iran with a population of more than 12 million, which according to the earthquake scenarios developed under the JICA-CEST project, has high seismic potential with many active faults (Khorsi, Bozorgi-Amiri, & Ashjari, 2013). Tehran Disaster Mitigation and Management Organization (TDMMO) has planned and implemented a pilot emergency evacuation program for two municipal regions. It has also an emergency habitation plan with 15 predefined safe zones in and around the city. The municipal zones and the pre-defined safe locations are shown Fig. 10.

The emergency habitation plan includes several steps e.g. setting up tents or cabins, providing basic supplies, drinking water, food, heating or cooling facilities and other basic equipment. As stated by the experts of the organization in several interviews, in case of a disaster, they do not consider opening all pre-defined locations in practice. We have implemented our model to make decisions to evacuate from two pilot zones while incorporating the locational decisions about choosing amongst the pre-defined safe zones. In order to address the extent of the disaster, we divide the logistics network with 92 nodes (representing the main intersections) and 2547 arcs with given integer capacities, into four areas (arcs are assumed to belong to the area of their head). The objective is to maximize the number of evacuees by choosing five safe zones. In each area, the decrease and increase parameters are generated randomly at different intervals. We have followed the parameter settings of Yuan and Wang (2009) to define four grades of disasters severity as shown in Table 12.

The characteristics of the network are extracted from the data sets provided by TDMMO, except some minor modifications to adapt the data to the assumptions of the model. The results of implementing Algorithm E1 for Tehran network, are reported in Table 13. The last column of the table reports the array of the selected safe zones according to their indices in $\|L\| = \{15\}$ The results show that the increase in severity and spread of the disaster not only reduces the total number of evacuees but also changes the destination safe zones, to the extent that there is only 1 destination in common between 0 and 4 disaster grades.

6. Conclusion and further remarks

We have demonstrated the new potentials of the max-flow problem in modeling real world evacuation problems, especially in emergency logistics. Incorporating locational considerations when optimizing network flow is a better reflection of real world evacuation situations. As the first attempt in this area and benefiting from the flexibility in an evacuee's destination selection in the planning process, we have developed a mixed integer linear programming model. This model aims to find a number of safe locations amongst a set of candidate destinations which maximize the flow of evacuees. Exploiting the structure of the model we were able to solve the model to optimality using established max-flow algorithms. However, for the sake of improving the computation time, two heuristics as well as two upper bounds, were proposed for the problem. They were implemented on randomly generated graphs and were compared in terms of average computation time and the gap to optimality.

To further illustrate the real world implications of the evacuation process and the impacts of the severity of the disasters, we have also implemented one of the exact algorithms (Algorithm E1) on the logistics network of Tehran.

The general idea of combining the locational decisions with that of network flow optimization yet remains motivating, both theoretically and empirically. In this paper we considered flow maximization as the objective function; however other objective functions in network optimization have the same importance in evacuation planning. To name a few, minimization of the transit time of a fixed flow F when there is a fixed population or supply in the source node or minimization of both time and cost of openings of destination nodes. Also, the objectives of the quickest flow problem and earliest arrival problem can be considered in combination with location decisions. Moreover, the idea can be extended to dynamic network problems, where the capacities are not assumed to be constant at the first step, or change due to disastrous circumstances.

References

Abdelgawad, H., & Abdulhai, B. (2009). Emergency evacuation planning as a network design problem: A critical review. *Transportation Letters: The International Journal of Transportation Research, 1*(1).

Abounacer, R., Rekik, M., & Renaud, J. (2014). An exact solution approach for multi-objective location-transportation problem for disaster response. *Computers & Operations Research, 41*, 83–93.

Afshar, A., & Haghani, A. (2012). Modeling integrated supply chain logistics in real-time large-scale disaster relief operations. *Socio-Economic Planning Sciences, 46*(4), 327–338.

Ahuja, R. K., Magnati, T. L., & Orlin, J. B. (1993). *Network flows*. Prentice-Hall.

Bar-Gera, H. (2015). Berlin mitte center network < https://github.com/bstabler/TransportationNetworks/tree/master/Berlin-Mitte-Center > .

Bayram, V., Tansel, B.Ç., & Yaman, H. (2015). Compromising system and user interests in shelter location and evacuation planning. *Transportation Research Part B: Methodological, 72*, 146–163.

B.o.C.C. Alachua County (2017). *Emergency management; first steps, shelters* < http://www.alachuacounty.us/Depts/PublicSafety/em/firstSteps/beInformed/Pages/Shelters.aspx > .

B. o. C.C. Collier County (2017). *Evacuation & shelter information* < http://www.colliergov.net/home/showdocument?id=64880 > .

B. o. C.C. Polk County (2017). 2017 hurricane shelter map < https://www.polk-county.net/boccsite/Documents/2017-Hurricane-Shelter-Map/ > .

Boonmee, C., Arimura, M., & Asada, T. (2017). Facility location optimization model for emergency humanitarian logistics. *International Journal of Disaster Risk Reduction, 24*, 485–498. http://dx.doi.org/10.1016/j.ijdrr.2017.01.017 ISSN 2212-4209.

Bozorgi-Amiri, A., Jabalameli, M. S., & Mirzapour Al-e Hashem, S. M. J. (2011). A multi-objective robust stochastic programming model for disaster relief logistics under uncertainty. *OR Spectrum, 35*(4), 905–933.

Caunhye, A. M., Nie, X., & Pokharel, S. (2012). Optimization models in emergency logistics: A literature review. *Socio-Economic Planning Sciences, 46*(1), 4–13.

Chalmet, L. G., Francic, R. L., & Saunders, P. B. (1982). Evacuation network models for building evacuation. *Management Science*(January).

Dow, K., & Cutter, S. (1998). Crying wolf: Repeat responses to hurricane evacuation orders. *Coastal Management, 26*(4), 237–252.

Drabek Thomas, E. (1999). Understanding disaster warning responses. *Social Science, 36*(3), 515–523.

Edmonds, J., & Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM), 19*(2), 248–264.

Fleischer, L., & Tardos, E. (1998). Efficient continuous-time dynamic network flow algorithms. *Operations Research Letters, 23*, 71–80.

Ford, L. J., & Fulkerson, D. R. (2015). *Flows in networks*. Princeton University Press.

Franzese, O., & Han, L. D. (2001). A methodology for the assessment of traffic management strategies for large-scale emergency evacuations. In *Proceedings of the ITS 2001 conference*. Miami Beach, FLA.

Fu, W., Fu, H., Skøtt, K., & Yang, M. (2008). Modeling the spill in the Songhua river after the explosion in the petrochemical plant in Jilin. *Environmental Science and Pollution Research, 15*(3), 178.

Ganjeh, M., & Einollahi, B. (2016). Mass fatalities in Hajj in 2015. *Trauma Monthly, 21*(5).

Hamacher, H. W., Heller, S., & Rupp, B. (2013). Flow location (FlowLoc) problems: Dynamic network flows and location models for evacuation planning. *Annals of Operations Research*, 161–180.

Hoppe, B., & Tardos, É. (1994). Polynomial time algorithms for some evacuation problems. In *SODA* (Vol. 94, pp. 433–441).

Hoppe, B., & Eva, T. (2000). The quickest transshipment problem. *Mathematics of Operations Research, 25*(1), 36–62.

ISDR, U. (2009). *UNISDR terminology on disaster risk reduction*, May. Geneva, Switzerland.

Jahn, O., Ohring, R. H. M., Schulz, A. S., & As, N. (2005). System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations Research, 53*(4), 600–616.

Khayal, D., Pradhananga, R., Pokharel, S., & Mutlu, F. (2015). A model for planning locations of temporary distribution facilities for emergency response. *Socio-Economic Planning Sciences, 52*, 22–30.

Khorsi, M., Bozorgi-Amiri, A., & Ashjari, B. (2013). A nonlinear dynamic logistics model for disaster response under uncertainty. *Significance, 3*(4).

Kotnyek, B. (2003). *An annotated overview of dynamic network flows*, PhD thesis. INRIA.

Lim, G. J., Zangeneh, S., Baharnemati, M. R., & Assavapokee, T. (2012). A capacitated network flow optimization approach for short notice evacuation planning. *European Journal of Operational Research, 223*(1), 234–245.

Lin, M., & Jaillet, P. (2015). On the quickest flow problem in dynamic networks: A parametric min-cost flow approach. *Proceedings of the twenty-sixth annual ACM-SIAM symposium on discrete algorithms* (pp. 1343–1356). Society for Industrial and Applied Mathematics.

Mirhashemi, S., Ghanjal, A., Mohebbi, H., & Moharamzad, Y. (2007). The 2003 Bam earthquake: Overview of first aid and transport of victims. *Prehospital and Disaster Medicine, 22*(6), 513–516.

Muoz, L., Sun, X., Horowitz, R., & Alvarez, L. (2006). Piecewise-linearized cell transmission model and parameter calibration methodology. *Transportation Research Record: Journal of the Transportation Research Board 1965*, 183–191.

Özdamar, L., & Ertem, M. A. (2015). Models, solutions and enabling technologies in

humanitarian logistics. *European Journal of Operational Research, 244*(1), 55–65.
Pelling, M., Maskery, A., Ruiz, P., & Hall, L. (2004). Reducing disaster risk: A challenge for development; a global report.
Perry, R. (1985). *Comprehensive emergency management: Evacuating threatened populations.* London: Jai Press.
Sayyady, F., & Eksioglu, S. D. (2010). Optimizing the use of public transit system during no-notice evacuation of urban areas. *Computers & Industrial Engineering, 59*(4), 488–495.
Sbayti, H., & Mahmassani, H. (2006). Optimal scheduling of evacuation operations. *Transportation Research Record: Journal of the Transportation Research Board (1964),* 238–246.
Tofighi, S., Torabi, S. A., & Mansouri, S. A. (2016). Humanitarian logistics network design under mixed uncertainty. *European Journal of Operational Research, 250*(1), 239–250.
Urbina, E., & Wolshon, B. (2003). National review of hurricane evacuation plans and policies: A comparison and contrast of state practices. *Transportation Research Part A: Policy and Practice, 37*(3), 257–275.
Yuan, F., Han, L., Chin, S.-M., & Hwang, H. (2006). Proposed framework for simultaneous optimization of evacuation traffic destination and route assignment. *Transportation Research Record: Journal of the Transportation Research Board 1964,* 50–58.
Yuan, Y., & Wang, D. (2009). Path selection model and algorithm for emergency logistics management. *Computers & Industrial Engineering, 56*(3), 1081–1094.