A matheuristic for the stochastic facility location problem

Reference:
Turkeš Renata, Sörensen Kenneth, Cuervo Daniel Palhazi.- A matheuristic for the stochastic facility location problem
Journal of heuristics - ISSN 1381-1231 - Dordrecht, Springer, 27:4(2021), p. 649-694
Full text (Publisher's DOI): https://doi.org/10.1007/S10732-021-09468-Y
To cite this reference: https://hdl.handle.net/10067/1766670151162165141

Journal of Heuristics manuscript No.
(will be inserted by the editor)

A matheuristic for the stochastic facility location problem

Renata Turkeš · Kenneth Sörensen ·
Daniel Palhazi Cuervo

Received: April 04 2019 / Accepted: date

Abstract In this paper, we describe a matheuristic to solve the stochastic facility location problem which determines the location and size of storage facilities, the quantities of various types of supplies stored in each facility, and the assignment of demand locations to the open facilities, which minimize unmet demand and response time in lexicographic order. We assume uncertainties about demands, inventory spoilage, and transportation network availability. A good example where such a formulation makes sense is the the problem of pre-positioning emergency supplies, which aims to increase disaster preparedness by making the relief items readily available to people in need. The matheuristic employs iterated local search techniques to look for good location and inventory configurations, and uses CPLEX to optimize the assignments. Numerical experiments on a number of case studies and random instances for the pre-positioning problem demonstrate the effectiveness and efficiency of the matheuristic, which is shown to be particularly useful for tackling larger instances that are intractable for exact solvers. The matheuristic is therefore a contribution to the literature on heuristic approaches to solving facility location under uncertainties, can be used to further study the particular variant of the facility location problem, and can also support humanitarian logisticians in their planning of pre-positioning strategies.

Keywords Facility location · Iterated local search · Matheuristic · Humanitarian logistics

1 Introduction

The mathematical science of facility locating has attracted much attention in discrete and continuous optimization for over a century. Investigators have focused on both formulations and algorithms in diverse settings in both the private sectors (e.g., industrial plants, banks, retail facilities, etc.) and the public sectors (e.g., hospitals, post stations, etc.) [24]. Since demands, travel times and other inputs to the facility location models might be highly uncertain, the development of models and solution procedures for facility location under uncertainty has in particular become a high priority for researchers in both the logistics and optimization communities [69].

Renata Turkeš
University of Antwerp IDLab
Sint-Pietersvliet 7, Antwerp, Belgium
Tel.: +3232654133
E-mail: renata.turkes@uantwerpen.be

Kenneth Sörensen
University of Antwerp Operations Research group ANT/OR
Prinsstraat 13, Antwerp, Belgium

Daniel Palhazi Cuervo
SINTEF Mathematics and Cybernetics
Forskningsveien 1, Oslo, Norway

In this paper, we consider a stochastic capacitated facility location problem which determines the location and category of storage facilities, the quantities of various types of supplies stored in each facility, and the assignment of demand locations to open facilities which minimize

- unmet demand, and
- response time

in lexicographic order, under budget restrictions. We assume uncertainties about

- demands (which can also be zero, so that even the set of demand locations is uncertain),
- inventory spoilage (a facility might not be operational, or remain only partially operational with only some percentage of goods which remain usable), and
- transportation network availability (some transportation links might be unusable).

The particular version of the stochastic facility location problem under study is NP-hard (as an extension of the simplest, but NP-hard uncapacitated facility location problem [17, 28]) and (meta)heuristics are therefore needed to solve problem instances of realistic size. In addition, multi-objective problems in practice need to be solved multiple times, by considering e.g., different weights in the objective function, different values of ε in ε-constraint methods, or in order to let managers choose between a number of non-dominated solutions. Despite facility location being a strategic problem, it is for these reasons that an effective and efficient heuristic solution procedure is necessary. If there is no computational infrastructure available, or if the managers lack sufficient mathematical background, the simplicity of the solution procedure can also be crucial.

Humanitarian logistics is a good example of an application domain where the aforementioned problem assumptions and heuristic requirements make sense. Indeed, the policies and models developed for commercial supply chains most often cannot be directly applied to manage humanitarian inventories, due to the unique characteristics of the humanitarian setting [35]. For example, even though both customer service and low logistics costs are important for humanitarian organizations and business enterprises, efficiency is crucial for commercial supply chains, whereas satisfying beneficiary needs is always the utmost priority for humanitarian organizations. In a commercial setting, the demands are known or can be easily predicted, whereas in most humanitarian settings, there is a high level of uncertainty associated with location, type and amount of demand. The network infrastructure is generally stable and reliable in a commercial supply chain, whereas post-disaster network may be damaged and involve uncertainties [5]. The same is true for the survivability of the stockpiles: the stored goods might be destroyed in the disaster, so that the storage facilities should not necessarily be located close to the disaster area, as they run the risk of also being affected by the disaster (whereas in business logistics, proximity to the customers is crucial) [11, 66]. For example, a look at the 2010 earthquake in Haiti, shows that local capacity in terms of relief items and human resources was in place before the disaster, yet these resources were located so close to the disaster site that they were destroyed during the tremor [14].

More precisely, the particular definition of the stochastic facility location problem above is a suitable description of the problem of pre-positioning emergency supplies, which aims to increase disaster preparedness by storing the relief items at strategic locations, in order to make them readily available to people in need [74]. Indeed, the agility and readiness in the distribution of critical relief commodities (such as water, food, or medicine) are shown to be crucial, especially in the first 72 hours after the event, so that rescue teams can begin their activities and victims can thus stabilize their lives. The importance of pre-positioning relief supplies was demonstrated when Hurricane Katrina devastated New Orleans in 2005. The lack of pre-positioned materials and the delay in arrival of these supplies hampered further relief to the victims [9]. Public records and interviews with the individuals directly involved in the logistical response indicate that the Federal Emergency Management Agency started deploying supplies just a day before the Katrina landfall (which was soon suspended because of the risks posed by the imminent strike of the hurricane) and was consequently still focused on procuring and deploying resources when it was expected to have those services already available to victims. Next to the time-consuming and bureaucratic nature of the procurement process, the supplies that can be expected to be high in demand tend to be more difficult to find, and it comes as no surprise that in this case the suppliers were not able to supply goods in the necessary quantities. It has also been argued that the selection of storage facilities delayed the immediate response, since some supplies were stored too far away from the affected areas. For these reasons, some people in need of assistance have not received the needed supplies until 7 to 10 days after landfall, and the quantities of supplies received were significantly lower than the quantities requested [37].

The requirements for the efficiency and simplicity of the solution techniques also makes sense for pre-positioning of relief items. Indeed, most relief organizations use public buildings like gyms or town halls as provisional warehouses or shelters for evacuees and do not build new facilities. Therefore, decisions where to store relief items are often made on a short-term basis (e.g., as soon as a hurricane forecast is available, usually three to five days prior to a landfall, although hurricane path and strength can change within few hours) [32]. As a result, pre-disaster decisions tend to develop a more operational character and hence solution times gain massively in importance [31]. In addition, the simplicity of the solution procedure might be very beneficial as it can be directly used for planning of emergency strategies. Despite a substantial body of literature on humanitarian logistics and discontent with current practices, interviews with humanitarian logistics experts indicate that decision support tools and advanced planning approaches are extremely rarely employed in the humanitarian sector (unlike in the private sector logistics) [82]. The main reasons for this are staff capacity (not enough staff, and staff that is not sufficiently skilled to use these techniques) and the related costs - the increase in effectiveness realized by advanced planning may not outweigh the cost of implementation, operation, and maintenance of an expensive tool or staff training ("such IT system may deliver slightly higher efficiency, but may cost one vehicle") [82]. In addition, systems involving black box optimization (i.e., a complex system or device whose internal workings are hidden or not readily understood) may fail to meet humanitarian standards concerning transparency [81].

The main contribution of this paper lies in the effective, efficient and easy-to-understand heuristic to solve a particular realistic variant of the stochastic facility location problem. In addition, the heuristic is used to find the first, and thus benchmark, solutions for the only publicly shared instances for the facility problem under study (with complete solutions made available). Finally, since the problem formulation corresponds to the problem of pre-positioning emergency supplies, the heuristic can be used to further study the pre-positioning problem and to support the planning of disaster preparedness strategies.

We start the paper with the literature review on the facility location problem, with the focus on the facility location under uncertainties in Section 2.1, and facility location for pre-positioning of relief items in Section 2.2. Section 3 gives a detailed description of the particular stochastic facility location problem under study, and introduces the corresponding mathematical model. The matheuristic we developed for the problem is introduced in Section 4, and evaluated in Section 5. The paper ends with a discussion about the main contributions and limitations, and the resulting interesting directions for future work in Section 6.

2 Literature review

Facility location models are used in a variety of applications, including locating warehouses within a supply chain to minimize the average time to market, locating noxious material to maximize their distances from the public, locating railroad stations to minimize the unpredictability of delivery schedules, locating automatic teller machines to serve bank customers better, etc. [24]. For surveys of the literature on facility location, see [1, 17, 18, 25, 27, 45, 47, 54, 63, 64, 70, 71].

The study of location theory formally began in 1909 when Alfred Weber considered how to position a single warehouse (in a plane) so as to minimize the total distance between the facility and customers [84]. Location theory gained renewed interest in 1964 with a publication by Hakimi [34], who considered the problem of locating one or more facilities on a network so as to minimize the total distance between customers and their closest facility, or to minimize the maximum such distance [58].

The most basic facility location problem formulations can be characterized as uncapacitated, static and deterministic, and can be classified into median, covering and center problems. The p-median or `minsum` problem looks for an assignment of all demand locations to a given number p of open facilities, which minimizes the total or average distance. For some applications (e.g., locating fire stations or ambulances), however, selecting locations which minimize the average distance traveled might not be the most suitable. Set covering problem aims to minimize the facility opening costs (which is equivalent to minimizing the number of open facilities, if there is no difference in facility costs) such that the distance between any demand location to an open facility (so called coverage) is under a given maximum acceptable limit. Since set covering problem becomes infeasible in many practical applications, it makes sense to consider a maximal covering problem which maximizes the amount of demand covered within a given acceptable distance. Another class of problems which can avoid the infeasibility of set covering problems is the center problem which endogenously determines the minimum coverage distance associated with locating a given number of facilities. The center problem, also known as the `minmax` problem, looks for the assignment of all demand locations to open facilities which minimizes the maximum distance between a demand location and the nearest facility [58].

The problem investigated in this paper resembles the median problem. For a given number of facilities to be open, the problem is polynomial, but it is NP-hard if the number of open facilities can vary [58]. Our problem further differs from the standard formulation with respect to a number of elements. Firstly, deterministic formulations are not able to adequately model the uncertainties inherent in making real-world strategic decisions. Section 2.1 gives a literature review on facility location under uncertainties. Next to the uncertainties about a number of classes of input parameters, we also consider different objective functions and incorporate many realistic assumptions, e.g., facilities are capacitated, we consider facilities with different capacities and opening costs (so that, in addition to the location, we also need to decide on the category of facility to be open), we consider multiple commodity types with different volumes and acquisition costs, etc. Note than only relaxing the assumption on uncapacitated facilities makes the problem much more complicated, since demand locations are naturally assigned to the nearest facility in the standard formulation, so that the integrality constraints on the assignment decision variables can be relaxed to a simple non-negativity constraint in the simplified problem [58]. This particular formulation make sense in practice as it can be used to describe the problem of pre-positioning emergency supplies; in Section 2.2 we discuss approaches to the stochastic facility problem in the humanitarian logistics literature.

2.1 Facility location under uncertainties

More often than not, the facilities, plants or distribution centers function for years or decades, during which time the environment (costs, demands, travel times) in which they operate may change substantially. It is therefore important to study facility location under uncertainties [69]. In certainty situations, all parameters are deterministic and known, whereas problems under uncertainty involve randomness, and the goal of is to find a solution that will perform well under any possible realization of the random parameters. Some reviews of the literature on facility location under uncertainties can be found in [58, 69].

One class of formulations of uncertainty are robust optimization problems, with no information about the probabilities of the uncertain parameters and the goal of optimizing worst-case (`minmax`) performance. Alternatively, the values of uncertain parameters can be governed by probability distributions known to the decision maker, with the goal of optimizing the expected value (`minsum`). This stochastic formulation can either explicitly consider the probability distribution of uncertain parameters, or it can capture uncertainty through scenario planning. The primary attraction of `minmax` measures is that they do not require the planner to estimate scenario probabilities, or even to formulate scenarios if data are described using intervals. However, `minmax` problems seem to be employed more in the academic literature than in practice. In many situations, it is more practical to plan based on a fractile target than on the worst case [69].

In 1961, Manne published one of the earliest papers [53] to consider stochastic problem inputs. In this paper, the problem of capacity expansion over an infinite horizon is examined, with the objective of selecting expansion sizes which minimize the sum of discounted installation costs. Cooper considers the Weber problem in which the locations of the demand points may be random [16]. A bivariate normal distribution for these locations is assumed. The objective is to choose a point for the single facility location to minimize the expected demand weighted distance to the customers, and the iterative algorithm that solves the first-order conditions is shown to be globally convergent in [41]. Manne and Cooper model demand probabilistically.

However, the scenario approach generally results in more tractable models, and furthermore, it has the advantage of allowing parameters to be statistically dependent. Dependence is often necessary to model reality, since, for example, demands are often correlated across time periods or geographical regions [69]. The problem studied in this paper belongs to the last class of stochastic, scenario approaches, and is commonly modelled as a two-stage stochastic linear program [8, 40]. The first-stage facility decisions are optimized before any particular scenario outcome, and the second-stage recourse decisions on the allocation of demand locations to open facilities define the actions to be taken in response to each random outcome.

The L-shaped method [79] (also known as the stochastic Benders decomposition) is a classical approach for solving this class of problems [32]. Louveaux presents the stochastic version of the capacitated p-median problem in which demands, production costs, and selling prices are random, and the constraint requiring p facilities to be opened is replaced by a budget constraint on the total cost [51]. The goal is to choose facility locations, determine their capacities, and decide which customers to serve and from which facilities to maximize the expected utility of profit. Since demands are random and facilities are capacitated, the facilities chosen in the first stage may be insufficient to serve all of the demands in the second stage; hence a penalty for unmet demand is included in the models. Louveaux and Peeters present a dual-based heuristic for this problem [52], and Laporte et al. present an optimal algorithm based on the L-shaped method in [46]. Although the (adjusted) L-shaped method exploits the specific structure of the problem, many large scale problems (e.g., with a large number of scenarios) remain unmanageable without the use of heuristics. The facility decisions may even be better than the ones derived by exact methods from a simplified problem formulation using only a small number of scenarios [31].

Weaver and Church present a Lagrangian relaxation algorithm for the stochastic p-median problem on a general network discussed by Mirchandani and Odoni [55], relaxing the assignment constraints [83]. Mirchandani et al. [56] begin with the same formulation as Weaver and Church in [83], also suggesting a Lagrangian relaxation method, but instead of relaxing the assignment constraints, they relax the single constraint requiring p facilities to be opened. The authors solve this subproblem using the DUALOC algorithm [23] and update the multiplier using a subgradient method. A location-inventory model is introduced in [19] which minimizes the expected cost of locating facilities, transporting material, and holding inventory under stochastic daily demand, and is solved using Lagrangian relaxation. The same model is also solved using column generation in [68].

A stochastic programming model is introduced in [49] for choosing facility locations involving the collection, recycling, and reuse of sand from demolition sites. The uncertainties are assumed for the demands (for recycled sand) and for the amount and location of supply (of waste sand). The model is solved using CPLEX. In [12], "stochastically processed demands" are considered, i.e., demands that arise from a queueing process at the customer, with the objective of minimizing the expected cost. A heuristic using stochastic decomposition (an extension of Benders decomposition) and space-filling curves is employed to solve the problem.

Exact solution techniques are often not the most suitable approach for realistic facility location problems, as they are not able to solve larger instances and/or to provide solution within a reasonable computation time. Moreover, understanding and implementing these or related techniques requires sufficient mathematical background (as most of them are based on Benders decomposition and/or Lagrangian relaxation, column generation or dual problem) and computational infrastructure, and they therefore cannot be easily used in many practical applications.

Widely used straightforward heuristic approaches to solving different location problems belong to the class of locate-allocate heuristics [39], first proposed in [15]. The location-allocation heuristic alternates between solving the location and allocation sub-problems that are easy to solve in separation: given a facility configuration, a simple allocation method is to assign every demand vertex to its closest facility; given the allocation of demand locations among the facilities, the problem is reduced to the solution of a number of independent single facility problems [80]. This fails to remain true for the stochastic facility location problem, since different sets of demand locations are allocated to a facility across different scenarios.

In the existing literature on stochastic facility location which introduces heuristic solution approaches for the problem, a number of simplifying assumptions are considered, compared to our problem definition. A literature review on facility location under uncertainties [58] underlines the need for improved heuristics to support the solution of larger, more complex and more realistic facility location problem instances. The increased use of scenario planning techniques will drive such solution advances, as scenario-based models grow rapidly with the number of scenarios generated [58].

2.2 Pre-positioning emergency supplies

Even in the growing body of literature on the topic of pre-positioning emergency supplies (for a recent literature survey, see [5, 31]), the particular variant of the stochastic facility location problem which describes the pre-disaster planning has not yet been solved with an effective, efficient and transparent heuristic. Indeed, as we summarize in Table 1, most of the articles on emergency pre-positioning do not incorporate on or more of the elements listed below:

- realistic problem assumptions
- proper formulation of the aid distribution sub-problem
- appropriate objective function
- (simple) heuristic solution algorithm
- multiple problem instances to reliably demonstrate the heuristic effectiveness,

as we elaborate in greater detail in the remainder of this section.

Firstly, there are many formulations of the pre-positioning problem that fail to consider the crucial complexities of the problem: some do not consider multiple facility categories (that can be opened at any potential facility location) or commodity types, some do not consider uncertainties about demand, survivability of pre-positioned aid or network damage, some do not incorporate facility or inventory decisions (Table 1). Furthermore, in some articles the authors assume uncapacitated facilities or, e.g., assume a single open facility, or a single demand location in each disaster scenario. These assumptions significantly simplify the mathematical formulation of the problem, and in particular the algorithms that solve the problem. Our problem definition adopts the assumptions introduced in [60], seen as a benchmark in the literature [32].

Table 1 Literature review on the problem of pre-positioning emergency supplies shows that most of the articles do not consider all the problem aspects, often minimize costs and only employ a commercial solver to solve a single case study.

Article	Instance					Solution			Mathematical model		Heuristic	Various instances
	Multiple facility categories	Multiple commodity types	Uncertainty about demand	Uncertainty about aid survivability	Uncertainty about network damage	Facility decisions	Inventory decisions	Distribution decisions	Distribution formulation	Objective function		
[2]	-	✓	✓	✓	✓	✗	✓	✓	Network flow problem	Sum of logistics and penalty costs	✓	✓
[3]	✗	✓	✓	✗	✓	✓	✓	✓	Maximum covering problem	Met demand	✗	✗
[7]	✗	✓	✗	✗	✗	✓	✓	✓	Maximum covering problem	Coverage; Opening cost; Sum of transportation and penalty costs	✗	✗
[13]	-	✗	✓	✓	✗	✗	✓	✓	Network flow problem	Sum of logistics and penalty costs	✗	✗
[20]	-	✗	✓	✓	✓	✗	✓	✓	Network flow problem	Sum of logistics and penalty costs	✗	✗
[9]	✗	✓	✓	✗	✓	✓	✓	✓	Transportation problem	Sum of logistics and penalty costs	✗	✗
[21]	✗	✓	✓	✗	✓	✓	✓	✓	Network flow problem	Sum of logistics and penalty costs	✓	✓
[22]	✗	✓	✓	✗	✗	✓	✓	✓	Transportation problem	Demand-weighted time	✗	✗
[30]	✓	✓	✓	✓	✓	✓	✓	✓	Network flow problem	Sum of logistics and penalty costs	✗	✗
[33]	✗	✗	✗	✗	✗	✓	✗	✓	Network flow problem	Logistics costs; Met demand	✗	✓
[38]	✓	✗	✓	✗	✓	✓	✓	✓	Network flow problem	Logistics costs	✗	✗
[42]	-	✓	✓	✗	✓	✗	✓	✓	Network flow problem	Sum of logistics and penalty costs	✗	✗
[43]	✗	✓	✗	✗	✗	✓	✓	✓	Network flow problem	Sum of logistics and penalty costs	✗	✗
[44]	✗	✗	✓	✗	✗	✓	✗	✓	Assignment problem	Minimum and average weight of open facilities; Distance	✗	✓
[48]	✗	✓	✓	✗	✓	✓	✓	✓	Network flow problem	Sum of logistics and penalty costs	✗	✗
[60]	✓	✓	✓	✓	✓	✓	✓	✓	Network flow problem	Sum of logistics and penalty costs	✓	✗
[61]	✓	✓	✓	✓	✓	✓	✓	✓	Network flow problem	Sum of logistics and penalty costs	✗	✗
[62]	✗	✓	✓	✗	✓	✓	✗	✓	Routing problem	Sum of met demand utility and residual budgets	✗	✓
[65]	✗	✓	✓	✓	✓	✓	✓	✓	Transportation problem	Demand-weighted time; Sum of logistics and penalty costs	✗	✗
[67]	-	✗	✓	✗	✗	✗	✓	✓	Network flow problem	Sum of logistics and penalty costs	✗	✗
[72]	✓	✓	✓	✓	✓	✓	✓	✓	Network flow problem	Logistics costs; Time; Sum of penalty costs	✓	✓
[73]	✗	✗	✓	✗	✗	✓	✓	✓	Routing problem	Logistics costs; Met demand	✗	✗
[76]	-	✗	✓	✓	✗	✗	✓	✓	Network flow	Demand-weighted distance	✓	✗
[77]	✗	✗	✗	✗	✗	✓	✗	✓	Routing problem	Sum of logistics costs and coverage reliability	✗	✗
This article	✓	✓	✓	✓	✓	✓	✓	✓	Assignment problem	Met demand; Time	✓	✓

Next to the underlying problem assumptions mentioned, the mathematical models that are used to describe the pre-positioning problem vary greatly with respect to the formulation of the aid distribution sub-problem (Table 1). The aid distribution is commonly modelled as a network flow problem per commodity, most probably due to existence of efficient algorithms that solve it, e.g., [26, 29, 57]. Such formulation over-simplifies the distribution problem as it does not allow to easily take into account the capacity nor the number of vehicles needed to transport different commodities. In most cases, the general model used only provides the flow amounts between vertices without specifying the destined path of the flow, making the solution difficult to implement in a real-world system (Figure 1). Furthermore, serving a demand location from multiple facilities is operationally overly complex and might pose significant risks in a chaotic setting after a disaster, e.g., carrying out a plan where 17% (i.e., any percentage in the interval $(0, 100)$) of demand for one commodity and 54% of demand for another commodity at an affected location are served by one facility (and at one point in time), and the remainder by another (or more) facilities. The latter is the reason why we also did not choose to formulate the aid distribution as a transportation problem.

Commodity $k = 1$ Commodity $k = 2$

Fig. 1 The aid distribution sub-problem is commonly modelled using a network flow formulation per commodity, with the graph vertices corresponding to facility or (zero) demand locations, and the graph edges corresponding to the flows of goods over the respective transportation links. The general network flow formulation makes it difficult to implement the solution in the real-world. How many vehicles need to leave vertex $i = 1$ to distribute the commodity $k = 1$? How is the flow of 40 on the edge $(1, 3)$ distributed between these vehicles? Does the flow of 10 on the edge $(2, 5)$ belong to the path $1-2-5$, $1-3-2-5$, or $1-3-4-2-5$? If there is commodity $k = 1$ pre-positioned at a facility open at vertex $i = 3$, the flow on the edge $(2, 5)$ might also belong to the path $3-2-5$ or $3-4-2-5$. The demand of a vertex $i = 2$ is served from a number of different places, and often a non-empty vehicle leaves the vertex without meeting its demand; this can pose a security risk, since the aid is being distributed to the next location, while a part of the population has not received the needed assistance. If the network flow problem is solved as a series of single-commodity problems, the distribution scheme becomes even more chaotic, as different commodities might have to be distributed from different facilities, and using very different transportation links.

On the other hand, the formulation of the aid distribution sub-problem as a routing problem is a waste of computational effort. Indeed, in the preparedness phase before a disaster, one is only interested in deciding where to open the facilities and what to store there; the aid distribution sub-problem is only solved to provide an evaluation of the quality of the pre-positioning facility and inventory decisions. Once a disaster happens, it is highly unlikely that it will completely match one of the considered disaster scenarios, implying that the optimized routing schemes would be of no use. In our formulation, we consider the aid distribution as an assignment problem, deciding which demand locations are served by which open facilities.

Furthermore, although the objective of humanitarian relief is to minimize human suffering [36] (what is an important distinction from commercial supply chains, as mentioned earlier), cost minimization is the common objective in the pre-positioning problem formulations. Since meeting all demand after an emergency is rarely possible, the objective function is usually defined to be the sum of logistics costs and different types of penalty costs, e.g., costs for unmet demand or delayed service (Table 1). Finding reasonable values for these penalty costs is not an easy task, while the quality of the emergency plan is extremely sensitive to these intangible coefficients [4, 59, 67, 78], that are also controversial as they assign a price to human suffering. In [74], we discuss the appropriate choice of the objective function in the mathematical models that describe humanitarian logistics problems, and show that an alternative formulation that directly minimizes unmet demand and response time in lexicographic order is able to circumvent all the issues of the cost-minimizing model, without any performance loss. This objective directly reflects the priorities of disaster relief: provide assistance to the greatest number of people possible, as soon as possible. In addition, in [74] we illustrate how the alternative model also offers practitioners the flexibility to explore a number of diverse emergency plans in a straightforward manner, which helps to provide insights into the problem and can be of use in many aspects of disaster response planning.

Independent from the particular choice of the objective function(s), the pre-positioning problem is multi-objective in nature, and the multiple objectives can be tackled in a number of different ways which all require solving a problem many times if they aim to provide appropriate support in the decision-making process [74]. In addition, multiple (nondominated, or good) solutions can provide more flexibility to the decision makers by allowing them to incorporate some planning components that are difficult to include in the mathematical model, such as political concerns. Despite this obvious need for efficient heuristics, most of the articles do not consider heuristic techniques to solve the pre-positioning problem (Table 1).

In Table 1, there are four articles which consider the same assumptions as in this paper: [30, 60, 61, 72]. No solution procedure is introduced in [30] (rather, the authors propose a methodological tool kit for humanitarian logistics to support practitioners to identify, compare, and apply suitable OR model), and CPLEX is used to find the best pre-positioning strategy for the case study focused on hurricane threat in the Gulf Coast area of the US in [61]. Heuristics algorithms are introduced in the remaining two studies, but these rely on a number of complex techniques. In [60], L-shaped method is used to solve the master problem, Lagrangian relaxation is employed to tackle the sub-problem (standard techniques to solve this class of problems, as elaborated in Section 2.1), and a network simplex algorithm solves the minimum cost flow problems, for the same Gulf Coast hurricane case study. In [72], a (inefficient) population-based differential evolution metaheuristic is used to design an earthquake humanitarian relief chain in Tehran, together with a weighted augmented ε-constraint method, fuzzy chance constrained programming approach and fuzzy ranking method. In addition, as indicated in Table 1, these articles differ from this paper with respect to the aid distribution sub-problem and objective function formulation.

Finally, in most of the papers, the authors use only a single, often very small, case study to confirm their findings (Table 1), without much statistical confidence. The few articles that employ more than a single problem instance rarely comment on the diversity of the instances, e.g., with respect to the network topology or the relationship between the instance parameters. For example, 17 instances that are used in [2] are all based on the nominal data with little variation, and are therefore all focused on the disaster of the same type, scale, that occurred in the same region. In [21], 15 different instances are used, but they are all randomly generated and there is little evidence provided that these instance resemble any realistic disaster. In this paper, we employ 30 diverse case studies and 10 random instances introduced in [75] to show good performance of the matheuristic. The instances and complete solutions are made publicly available to allow comparison with other heuristics and to foster further research on the pre-positioning problem.

3 Problem definition

In this section, we briefly describe an instance, a solution, and the mathematical formulation of the particular stochastic facility location problem under study. An example of an instance can be found in the Appendix, and more details and further argumentation behind the choice of such a problem definition can be found in [74].

As already mentioned, we assume a number of uncertainties, and model them as a random vector with a finite number of possible realizations, called scenarios $s \in \mathrm{S}$, which occur with probabilities P^s. The transportation network in every scenario $s \in \mathrm{S}$ is represented by a directed graph $\mathrm{G}_s = (\mathrm{V}, \mathrm{E}_s)$. Every vertex $i \in \mathrm{V}$ represents a city or village that has a demand for D_i^{ks} units of commodity $k \in \mathrm{K}$ in scenario $s \in \mathrm{S}$ (which can be zero). The set of edges E_s represents the transportation links, with the weight of an edge (i, j) being the distance L_{ij}^s from vertex $i \in \mathrm{V}$ to vertex $j \in \mathrm{V}$ in scenario $s \in \mathrm{S}$. If there is no edge between vertices $i \in \mathrm{V}$ and $j \in \mathrm{V}$ in scenario $s \in \mathrm{S}$, we set $L_{ij}^s = -1$.

A facility of any category $q \in \mathrm{Q}$ might be open at any location $i \in \mathrm{V}'$ ($\mathrm{V}' \subseteq \mathrm{V}$), if the facility budget A permits. The facility categories differ in volume M_q and opening cost A_q. At any open facility, commodities of different types $k \in \mathrm{K}$ (such as food, water, or medicine) can be stored, if the facility capacity and acquisition budget B constraints are respected. The commodity types differ in unit volume M^k, unit acquisition cost B^k and unit transportation cost C^k. A percentage R_i^{ks} of stored commodity type $k \in \mathrm{K}$ at vertex $i \in \mathrm{V}$ remains usable in a scenario $s \in \mathrm{S}$, and can be distributed via the traversable edges to the demand locations, as long as the transportation budget C is not violated.

Given a problem instance described above, we want to determine the best possible strategy which determines:

- the location and category of storage facilities to open, represented by binary variables $\mathbf{x} = [x_{iq}]$ that indicate whether a facility of category $q \in \mathrm{Q}$ is open at vertex $i \in \mathrm{V}'$,
- the amounts $\mathbf{y} = [y_i^k]$ of commodity $k \in \mathrm{K}$ to store at a facility open at vertex $i \in \mathrm{V}'$, and
- the assignments of demand locations to open facilities, represented by binary variables $\mathbf{z} = [z_{ij}^s]$ that indicate whether a facility open at vertex $i \in \mathrm{V}'$ serves the demands of vertex $j \in \mathrm{V}$ in scenario $s \in \mathrm{S}$.

These strategies are developed before any scenario outcome when the logisticians only care to know where to open the facilities and what to store there, so that the distribution decisions $\mathbf{z}$ are not truly a part of a solution. However, the decisions about the transportation of the stored goods from the storage facilities to the demand locations in every possible scenario are made in order to ensure that the facility capacity, transportation network availability and transportation budget are respected, and to evaluate the quality of the facility and inventory decisions we consider implementing. The notation for an instance and a solution of the problem is summarized in Table 2.

Deciding on the best strategy is formulated as a two-stage stochastic mixed-integer programming problem, with the facility and inventory decisions $\mathbf{x} = [x_{iq}]$ and $\mathbf{y} = [y_i^k]$ as the first-stage variables (made before there is knowledge of any specific scenario outcome), and the scenario-specific assignment decisions $\mathbf{z} = [z_{ij}^s]$ as the

Table 2 Notation for the instance and solution of the problem.

Sets	
Q	set of facility categories
K	set of commodities
S	set of scenarios
V	set of vertices
V′	set of potential facility locations, $\mathrm{V}' \subseteq \mathrm{V}$
E_s	set of edges in scenario $s \in \mathrm{S}$
Coefficients	
M_q	volume capacity of a facility of category $q \in \mathrm{Q}$ (m^3)
A_q	opening cost of a facility of category $q \in \mathrm{Q}$ (€)
M^k	unit volume of commodity $k \in \mathrm{K}$ (m^3)
B^k	unit acquisition cost of commodity $k \in \mathrm{K}$ (€)
C^k	unit transportation cost of commodity $k \in \mathrm{K}$ (€)
W	average speed (km/h)
P^s	probability of scenario $s \in \mathrm{S}$
D_i^{ks}	demand for commodity $k \in \mathrm{K}$ at vertex $i \in \mathrm{V}$ in scenario $s \in \mathrm{S}$
R_i^{ks}	percentage of stored commodity $k \in \mathrm{K}$ that remains usable at vertex $i \in \mathrm{V}'$ in scenario $s \in \mathrm{S}$
L_{ij}^s	$\begin{cases} \text{distance from vertex } i \in \mathrm{V} \text{ to vertex } j \in \mathrm{V} \text{ in scenario } s \in \mathrm{S} \text{ (km)}, & (i,j) \in \mathrm{E}_s \\ -1, & \text{otherwise} \end{cases}$
A	total budget for opening the facilities (€)
B	total budget for acquisition of supplies (€)
C	total budget for transportation (€)
Decision variables	
x_{iq}	$\begin{cases} 1, & \text{if a facility of category } q \in \mathrm{Q} \text{ is open at vertex } i \in \mathrm{V}' \\ 0, & \text{otherwise} \end{cases}$
y_i^k	amount of commodity $k \in \mathrm{K}$ stored at vertex $i \in \mathrm{V}'$
z_{ij}^s	$\begin{cases} 1, & \text{if the facility open at vertex } i \in \mathrm{V}' \text{ fully meets the demands of vertex } j \in \mathrm{V} \text{ in scenario } s \in \mathrm{S} \\ 0, & \text{otherwise} \end{cases}$

Table 3 Auxiliary coefficients that can be derived from a problem instance.

Auxiliary coefficients	
$\mathcal{D}_i^s$	average percentage of the total demand in scenario $s \in \mathrm{S}$ that is needed at vertex $i \in \mathrm{V}$, $\mathcal{D}_i^s = \frac{1}{\lvert K \rvert} \sum_{k \in \mathrm{K}} \frac{D_i^{ks}}{\sum_{j \in \mathrm{V}} D_j^{ks}}$
$\mathcal{L}_{ij}^s$	$\begin{cases} \text{shortest path distance from vertex } i \in \mathrm{V} \text{ to vertex } j \in \mathrm{V} \text{ in scenario } s \in \mathrm{S} \text{ (km)}, \\ \text{if there is a path from vertex } i \in \mathrm{V} \text{ to vertex } j \in \mathrm{V} \text{ in graph } \mathrm{G}_s = (\mathrm{V}, \mathrm{E}_s) \\ -1, \quad \text{otherwise} \end{cases}$
$\mathcal{C}_{ij}^s$	$\begin{cases} \text{cost of transporting the demands of vertex } j \in \mathrm{V} \text{ via the shortest path from vertex} \\ i \in \mathrm{V} \text{ to vertex } j \in \mathrm{V} \text{ in scenario } s \in \mathrm{S} \text{ (€)}, \mathcal{C}_{ij}^s = \mathcal{L}_{ij}^s \sum_{k \in \mathrm{K}} C^k D_j^{ks}, \\ \text{if there is a path from vertex } i \in \mathrm{V} \text{ to vertex } j \in \mathrm{V} \text{ in graph } \mathrm{G}_s = (\mathrm{V}, \mathrm{E}_s) \\ -1, \quad \text{otherwise} \end{cases}$
$\mathcal{T}_{ij}^s$	$\begin{cases} \text{shortest path travel time from vertex } i \in \mathrm{V} \text{ to vertex } j \in \mathrm{V} \text{ in scenario } s \in \mathrm{S} \text{ (h)}, \\ \text{if there is a path from vertex } i \in \mathrm{V} \text{ to vertex } j \in \mathrm{V} \text{ in graph } \mathrm{G}_s = (\mathrm{V}, \mathrm{E}_s) \\ -1, \quad \text{otherwise} \end{cases}$

second-stage (recourse) variables. The model aims to minimize unmet demand and response time in lexicographic order, under resource and capacity constraints. In the context of pre-positioning emergency supplies, this corresponds to providing assistance to the greatest number of people possible, as soon as possible. A detailed argumentation behind the choice of such a formulation and methods to easily adapt the definition of lexicographic order is given in [74].

In order to improve the readability of the model, we calculate a few auxiliary coefficients using the instance information (Table 3). The model is now as follows.

$$\max \quad \sum_{s\in \mathrm{S}}\sum_{i\in \mathrm{V}'}\sum_{j\in \mathrm{V}} P^s \mathcal{D}_j^s z_{ij}^s \tag{1}$$

$$\sum_{k\in \mathrm{K}} M^k y_i^k \leq \sum_{q\in \mathrm{Q}} M_q x_{iq} \qquad (i\in \mathrm{V}') \tag{2}$$

$$\sum_{i\in \mathrm{V}'} z_{ij}^s \leq 1 \qquad (j\in \mathrm{V})(s\in \mathrm{S}) \tag{3}$$

$$\sum_{j\in \mathrm{V}} D_j^{ks} z_{ij}^s \leq R_i^{ks} y_i^k \qquad (i\in \mathrm{V}')(k\in \mathrm{K})(s\in \mathrm{S}) \tag{4}$$

$$z_{ij}^s \leq 1 + \mathcal{L}_{ij}^s \qquad (i\in \mathrm{V}')(j\in \mathrm{V})(s\in \mathrm{S}) \tag{5}$$

$$\sum_{i\in \mathrm{V}'}\sum_{q\in \mathrm{Q}} A_q x_{iq} \leq A \tag{6}$$

$$\sum_{i\in \mathrm{V}'}\sum_{k\in \mathrm{K}} B^k y_i^k \leq B \tag{7}$$

$$\sum_{i\in \mathrm{V}'}\sum_{j\in \mathrm{V}} \mathcal{C}_{ij}^s z_{ij}^s \leq C \qquad (s\in \mathrm{S}) \tag{8}$$

$$x_{iq} \in \{0,1\} \qquad (i\in \mathrm{V}')(q\in \mathrm{Q}) \tag{9}$$

$$y_i^k \geq 0 \qquad (i\in \mathrm{V}')(k\in \mathrm{K}) \tag{10}$$

$$z_{ij}^s \in \{0,1\} \qquad (i\in \mathrm{V}')(j\in \mathrm{V})(s\in \mathrm{S}) \tag{11}$$

The objective of the model is to maximize the expected (over scenarios) average (over commodities) percentage of met demand (1), i.e., to minimize the percentage of unmet demand. The first set of constraints (2) limits the amount of stored supplies at a vertex to the capacity of a facility opened there. The second set of constraints (3), together with (11), ensures that every vertex is served by at most one facility, and only if there is enough of stored goods that remained usable to serve the demands of that vertex (4), while (5) make sure that a demand vertex is only assigned to a facility if it can be reached from it. The constraints (6)-(8) represent the facility, acquisition and transportation budget limitations, respectively. Finally, constraints (9)-(11) are integrality and positivity constraints.

Let $\mathcal{D}^*$ be the optimal percentage of met demand, i.e., the objective function value of the optimal solution obtained by solving the model above. To further optimize response time in lexicographic order, an additional model needs to be solved that minimizes total expected response time subject to the same constraints (2)-(11), with an additional constraint that guarantees that the met demand is (greater than or) equal to the optimal value $\mathcal{D}^*$ obtained from (1)-(11).

$$\min \quad \sum_{s\in \mathrm{S}}\sum_{i\in \mathrm{V}'}\sum_{j\in \mathrm{V}} P^s \mathcal{T}_{ij}^s z_{ij}^s \tag{12}$$

$$(2)-(11) \tag{13}$$

$$\sum_{s\in \mathrm{S}}\sum_{i\in \mathrm{V}'}\sum_{j\in \mathrm{V}} P^s \mathcal{D}_j^s z_{ij}^s \geq \mathcal{D}^* \tag{14}$$

4 Matheuristic

As mentioned in first part of the paper, any reasonable planning of best facility strategy requires solving the facility location problem many times. For most realistic problem instances, this calls for an efficient and effective heuristic solution procedure. We start this section with a general description of the heuristic that we implemented for the problem, and proceed to illustrate each of the heuristic components in greater detail in the subsequent subsections.

To build a solution means to determine the values of decision variables $\mathbf{x} = [x_{iq}]$, $\mathbf{y} = [y_i^k]$ and $\mathbf{z} = [z_{ij}^s]$, i.e., to determine where to open the storage facilities, what to store there, and which demand vertices will be served by each facility in each scenario. If possible, it seems natural to try to resort to an exact solver, such as CPLEX, to make the optimal choices for one of the three sets of decisions, and optimize the remaining two sets of decision variables heuristically.

Designing a matheuristic that employs an exact solver to optimize the facility decisions $\mathbf{x}$ is not straightforward. Indeed, if $\mathbf{y}$ and $\mathbf{z}$ decisions are made heuristically so that there exist feasible $\mathbf{x}$, constraints (2) make it trivial to find $\mathbf{x}$: at each vertex $i \in \mathrm{V}$ where some goods are stored ($y_i^k > 0$ for some commodity $k \in \mathrm{K}$), there must be a facility open ($x_{iq} = 1$ for some facility category $q \in \mathrm{Q}$ that is able to store the sum of volumes of y_i^k, across all $k \in \mathrm{K}$).

The same holds for using a solver to optimize inventory decisions $\mathbf{y}$, although it might seem the most reasonable to optimize $\mathbf{y}$ using an LP solver, as they are continuous variables. Indeed, if $\mathbf{x}$ and $\mathbf{z}$ are optimized heuristically so that there exist feasible $\mathbf{y}$, constraints (4) imply that the usable amount of each commodity stored at vertex $i \in \mathrm{V}$ must be sufficient to meet the demands of all the vertices that are assigned to it, so that it becomes trivial to calculate $\mathbf{y}$:

$$y_i^k = \max_{s \in \mathrm{S}} \sum_{j \in \mathrm{V}} \frac{D_j^{ks}}{R_i^{ks}} z_{ij}^s$$

Therefore, using an exact solver to optimize $\mathbf{y}$ would also be a waste of efforts.

It is important to note here that the feasibility of the decisions made in each scenario does not imply the feasibility of a solution. Let us consider a small example with 1 facility category, 2 commodity types (water and food) and 2 scenarios, with the facility budget A that allows only one facility to be opened. A solution with two open facilities (possibly in disconnected regions), with each of them serving a number of demand vertices in one of the scenarios, can be feasible for each scenario, but such a solution violates the facility budget constraint. Consider further a solution with a single open facility, with 1000 units of water and 50 units of food transported from the facility to some demand vertices in scenario $s = 1$, and 300 units of water and 80 units of food in scenario $s = 2$. In order to be able to serve the demand of selected vertices, we would have to store 1000 units of water and 80 units of food, which can easily violate the facility capacity or inventory budget constraints. In general,

$$\max_{s \in \mathrm{S}} \sum_{k \in \mathrm{K}} B^k \frac{1}{R_i^{ks}} \sum_{j \in \mathrm{V}} D_j^{ks} z_{ij}^s \leq B \quad \not\Rightarrow \quad \sum_{k \in \mathrm{K}} \max_{s \in \mathrm{S}} B^k \sum_{j \in \mathrm{V}} \frac{D_j^{ks}}{R_i^{ks}} z_{ij}^s \leq B.$$

For these reasons, it is not straightforward to divide the optimization of the facility or inventory decisions into sub-problems corresponding to different scenarios.

On the other hand, if the (feasible) facility and inventory decisions $\mathbf{x}$ and $\mathbf{y}$ are made heuristically, the assignments $\mathbf{z}$ can immediately be optimized using an exact solver. Moreover, the assignment problem can be solved separately for each scenario: the only equations that include $\mathbf{z}$ in the mathematical model, equations (3), (4), (5), (8), (11), consist of a separate constraint for each scenario.

In our solution procedure, we therefore employ CPLEX to search for good assignments, but we note that the importance of CPLEX is less crucial than it might appear. Indeed, the inventory decisions $\mathbf{y}$ are continuous and thus difficult to optimize heuristically. However, the amounts of goods to be stored can be naturally derived from the assignments and it is for this reason that we always let the assignments $\mathbf{z}$ guide the inventory decisions $\mathbf{y}$. As already noted, the assignment decisions $\mathbf{z} = [z_{ij}^s]$ are made for each scenario, while the inventory decisions $\mathbf{y} = [y_i^k]$ need to be universal across all scenarios. Assigning vertices per scenario and deducing the inventory scheme would yield $|\mathrm{S}|$ different inventory configurations. In order to obtain a single inventory plan, for each open facility and each commodity we would have to take the maximum amount stored across all scenarios, but such an inventory plan is most likely infeasible due to a violation of the acquisition budget or facility capacity constraints. Therefore, we adjust the inventory scheme after every single assignment of a vertex in a scenario. It is only after the heuristic optimization of assignments (that is used to guide the inventory optimization) that we use CPLEX to further optimize the assignments. Since heuristic efficiency is a priority, CPLEX is used to optimize the assignments only of a limited number of selected solutions. This is later explained in greater detail. As noted earlier, the assignments are not actually part of the solution, but are only used to evaluate the quality of the facility and inventory decisions. However, using CPLEX to improve the assignments can help us identify the promising facility and inventory configurations that could be missed if the assignments were made heuristically.

The solution procedure that we implemented is therefore a matheuristic that optimizes the facility and inventory decisions $\mathbf{x}$ and $\mathbf{y}$ heuristically, and employs CPLEX to further improve the assignment decisions $\mathbf{z}$. In the remainder of this section, we first describe each of the matheuristic elements separately and then explain how we combined them together in the final subsection.

4.1 Initial solution

The initial solution is built using a greedy heuristic (Figure 2). Firstly, we must decide on the location and category of open facilities $\mathbf{x} = [x_{iq}]$. It seems reasonable to open facilities in areas where the demand is high, although we aim to avoid opening facilities at vertices where a significant percentage of the stored goods would not remain usable across a number of scenarios. For this reason, the facility location decisions are guided by the total demand of a certain number of neighbouring vertices and the percentage of goods that remains usable. We therefore start by ordering vertices from the greatest to the lowest expected average percentage of demand of neighbouring vertices, taking the percentage of goods that remains usable at the given vertex into account:

$$\mathcal{X}_i(m) = \sum_{s\in \mathrm{S}} P^s \sum_{k\in \mathrm{K}} R_i^{ks} \sum_{j\in \mathrm{V}(i,m)} \frac{D_j^{ks}}{\sum_{v\in \mathrm{V}} D_v^{ks}},$$

where $\mathrm{V}(i,m) \subseteq V$ is the set of m vertices that are the closest to vertex $i \in \mathrm{V}$ (including vertex i itself). We consider all the neighbouring vertices that can be reached from vertex $i \in \mathrm{V}$, rather than only the demand vertices, as this helps to additionally minimize the response time as it prefers to open facilities that have demand vertices within a shorter distance. The number m of neighbouring vertices is a parameter of the constructive heuristic.

Following this order, we open as many new facilities as the facility budget permits. We note here that these facility opening decisions are not deterministic, as a quick sort procedure is used to sort the neighbours according to distance, and the vertices according to demand measure $\mathcal{X}_i(m)$ of m neighbouring vertices. Indeed, quick sort takes a random pivot when sorting, and if there is a few vertices that are at the same distance from a given vertex $i \in \mathrm{V}$, the set of neighbours $\mathrm{V}(i,m)$ does not always include the same vertices. These neighbours that are at the same distance from $i \in \mathrm{V}$ probably have different demands, so that the total demand measure of the neighbourhoud varies according to the choice of neighbours. In addition, it can also happen that a few potential facility locations have the same $\mathcal{X}_i$; quick sort would not rank these candidates always in the same order.

To decide on the category of the facility to be opened, it seems reasonable to aim to be able to store sufficient amounts of goods to meet the demand of the neighbouring vertices in each scenario. Since a percentage of goods might be destroyed in some of the outcomes, the amounts of commodity $k \in \mathrm{K}$ that need to be stored at vertex $i \in \mathrm{V}$ to meet the demand of m neighbouring vertices in scenario $s \in \mathrm{S}$, are

$$\mathcal{Y}_i^{ks}(m) = \left\lceil \sum_{j\in \mathrm{V}(i,m)} \frac{D_j^{ks}}{R_i^{ks}} \right\rceil.$$

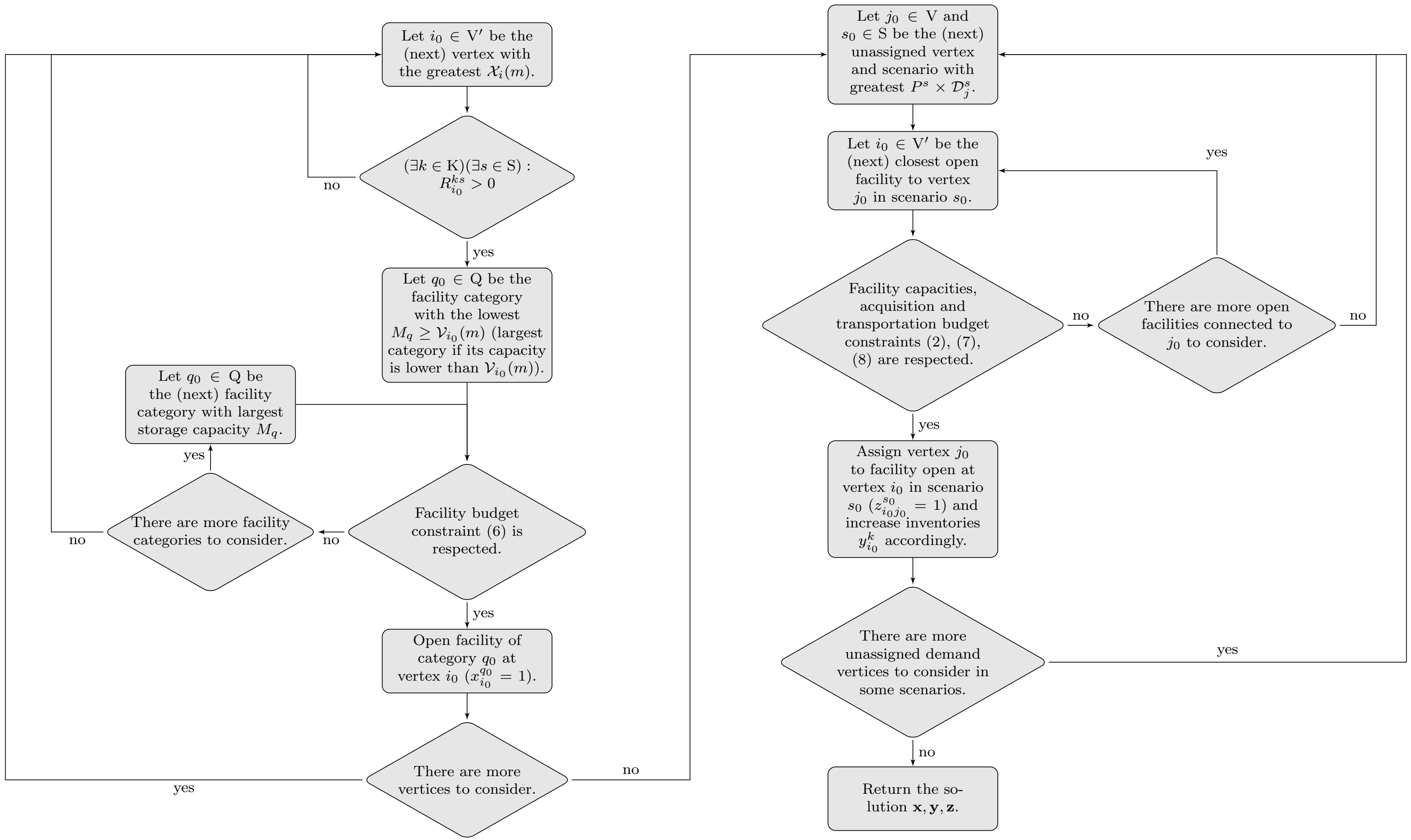


Fig. 2 greedy_constructive_heuristic(*instance*, m). Starting from an empty solution $\mathbf{x} = \mathbf{y} = \mathbf{z} = 0$, facility decisions are made in a greedy way and vary according to the number m of neighbouring vertices (left), and next the assignment of demand locations to open facilities across scenarios is done in a greedy fashion, with simultaneous increase of inventory (right). The initial solution is chosen as the best solution constructed for different values of parameter m.

In the case that the percentage R_i^{ks} of supplies that remains usable is zero, no amount of stored goods would be sufficient to meet the demands of neighbouring vertices and therefore we set $\mathcal{Y}_i^{ks}(m)$ to zero (although facilities will not be open at these vertices due to the first step, if there are better options, otherwise the solution would meet no demand). In order to be able to meet the demands of neighbouring vertices in all scenarios, the amounts to be stored at vertex $i \in \mathrm{V}$ would be

$$\mathcal{Y}_i^k(m) = \max_{s \in \mathrm{S}} \mathcal{Y}_i^{ks}(m).$$

The category of facility that we aim to open is therefore the category with minimum capacity that is able to store the volume of goods above:

$$\mathcal{V}_i(m) = \sum_{k \in \mathrm{K}} M^k \mathcal{Y}_i^k(m).$$

In case there is no such category as the total volume of supplies is too large, or if the facility budget would be violated, we open a facility of the largest possible capacity. Facilities are opened in this manner as long as the facility budget A is respected, or we have exploited all potential facility locations $i \in \mathrm{V}'$ (Figure 2, left).

Secondly, we make the inventory and assignments decisions $\mathbf{y} = [y_i^k]$ and $\mathbf{z} = [z_{ij}^s]$ simultaneously, due to their natural inter-connectedness that was elaborated earlier in this section. In every step, we perform a single assignment of a vertex in a scenario and increase the inventory accordingly, if necessary. We do this also in a greedy way, starting from the vertex $i \in \mathrm{V}$ and scenario $s \in \mathrm{S}$ with the greatest weighted percentage of demand, multiplied by the scenario probability, $P^s \times \mathcal{D}_i^s$, as this assignment will increase the primary objective the most. Every vertex is assigned to the closest facility possible, in order to increase the secondary objective the least, and to keep the transportation costs below the budget limit (Figure 2, right).

Instead of tuning the parameter m of the greedy heuristic, we run it for different values of m and choose the best found solution as the initial solution. The first reason for doing so is that for different instances the m that produces the best solution varies greatly, hence it becomes highly likely we would often miss out on good solutions by running the heuristic for a single value of m returned by the parameter tuning. Secondly, it might be of interest to practitioners to already produce a number of diverse solutions that can immediately offer insights into the structure of good solutions, which is an asset of such a constructive procedure. For instances of reasonable size (such as, e.g., all the case studies used in this paper), we run the greedy heuristic for any number of neighbours $m \in \{1, 2, ..., |V|\}$, as this takes less than one second of computation time. For large instances, we limit the calculations to only a few values of the parameter m not to waste too much computation time on finding an initial solution, especially since the quality of the solution rarely varies significantly with small changes of m. The computation time necessary to construct a solution for any m depends greatly on the number of assignments that have to be made after the empty facilities are open, in order to make inventory and assignment decisions (Figure 2, right). The maximum number of assignments that can be made to construct a single solution is the total number of nonzero demand pairs (i, s) of vertices $i \in \mathrm{V}$ and scenarios $s \in \mathrm{S}$. Therefore, the number of different solutions constructed, i.e., the number of different evenly distributed values of $m \in \{1, 2, ..., |V|\}$ for which the constructive heuristic is run, is such that the total number of assignments to be made is lower than 25 000, unless more assignments need to be made to construct a single solution.

4.2 Optimization of facility configuration $\mathbf{x}$

The facility configuration is optimized using the iterated local search metaheuristic [50]. In the local search, we make small changes to the facility configuration to find the local optimum, and then perform a perturbation in order to explore the solution space, i.e., to find the best out of many local optima.

The facility local search move closes a facility, proceeds to closing all empty facilities in order to increase the available facility budget, and opens a new facility. The new facility must be open at a different location or of a different category. In order to evaluate the quality of the new facility neighbour, we unassign all vertices, empty the facilities and perform a greedy assignment of vertices with simultaneous inventory increase (Figure 2, right).

Ideally, we would like to optimize the solution (i.e., the inventory and assignment decisions) after each such move to properly evaluate the quality of the new facility configuration, but that would be computationally too demanding. We thus resort to the greedy assignment with simultaneous inventory increase as a proxy evaluation of a facility neighbour. This often provides a good estimate of the quality of a facility configuration. If, however, the inventory optimization has a considerable probability of significantly improving the quality of the solution, we also perform the inventory optimization after the greedy assignment in order not to miss a promising facility neighbour with a poor evaluation. The inventory optimization can provide a much better idea of the quality of a facility configuration if the total number of assigned vertices is not too large (we limit it to 50), as the inventory local search changes the inventory according to only a single unassignment of a vertex, followed by one or a few assignments (see the description of the inventory optimization in Section 4.3); it is therefore limited to only 1 second of computation time.

The facility perturbation starts with closing a number of random facilities. This number is not a given percentage of open facilities, as this percentage would be difficult to tune since it does not give a very good idea of the measure of the change in a solution. For example, for a solution with 3 open facilities, the perturbation percentage 5, 10, 20 and 30 would all yield closing a single facility. It seems reasonable to close 2 random facilities each time, as making a random move in a neighbourhood of higher order than the one used by local search is a common perturbation that produces satisfactory results [50]. However, if we wish to be able to explore the solution space completely, we must also allow that closing a number of smaller facilities can ensure opening a facility with larger capacity. In order to do this, we need to enable closing a number of facilities such that the savings in the total facility cost can be sufficient to open a facility of a larger capacity. For this reason, the number of facilities to be closed is a random number drawn from a uniform distribution from the following set

$$\left\{2, 3, \ldots, \max_{q \in \{1,\ldots,|\mathrm{Q}|-1\}} \left\lceil \frac{A_{q+1}}{A_q} \right\rceil \right\}.$$

Next we open some facilities at random vertices and of a random category, until all vertices and categories are checked for opening or the facility budget constraint would not allow to open even the facility of the smallest category. New facilities cannot be open at the same locations where facilities were closed. After unassigning all vertices and emptying all facilities, the greedy assignment of vertices with simultaneous inventory increase (Figure 2, right) is performed to make reasonable assignment and inventory decisions for the new facility configuration. Of course, in order to allow a fair comparison of this solution to the facility neighbours in the next facility local search, we also perform an inventory local search if it is highly likely that it can significantly improve the solution (see above).

The facilities that are closed in the perturbation are blocked for opening in the next facility local search, to avoid returning to the same local optimum. To diversify the search even further, the facility perturbation is not performed on the best found solution, but rather on the last facility local optimum.

4.3 Optimization of inventory configuration $\mathbf{y}$

The inventory configuration is optimized also using the iterated local search metaheuristic, guided always by the changes in the assignments. The reason for this is that the inventory decisions are continuous and thus difficult to otherwise optimize heuristically. Besides, it seems natural to only store the amounts of goods that are necessary to cover the demands of vertices that we plan to serve. In the local search, we make small changes to the inventory configuration to find a local optimum, and then perturb the solution in order to explore the solution space, i.e., find the best out of many local optima.

The inventory(-assignments) local search move unassigns a vertex and decreases the inventory accordingly, then assigns a vertex and increases the inventory accordingly, and then performs a greedy assignment of vertices with simultaneous inventory increase (Figure 2, right). The new assignment must be an assignment of either a new vertex, or to a new facility, or in a new scenario (e.g., we can perform an assignment of the same vertex in the same scenario that has been unassigned, if we assign it to another open facility).

The inventory-assignments perturbation starts with the unassignment of the given percentage of random vertices with a simultaneous inventory decrease. Tuning the percentage of vertices to unassign is inconclusive, as the percentage of vertices to unassign that yields the best solutions varies greatly across different problem instances. We therefore let this percentage be a random number drawn from a uniform distribution from the interval [5,20]. The number of vertices that will be unassigned is the given percentage of the number of unassigned demand vertices, rather than the given percentage of the number of assigned vertices. We do this to avoid lacking vertices that are not blocked for assignment in the next iteration of assignments. In addition, if there are much more unassigned demand vertices than assigned vertices, this would mean we would have to unassign all vertices. This seems reasonable, as this actual number of vertices to unassign should indeed depend on the size of the pool of possibilities we have not tried out.

Next we start assigning random vertices to random facilities, increasing inventory accordingly. We stop when all vertices, scenarios, and facilities are checked for assignment. The combinations of vertices, facilities and scenarios that were unassigned are blocked for assignment in the local search to follow. To diversify the search even further, the inventory perturbation is also not performed on the best found solution, but rather on the last inventory local optimum.

The inventory local search changes the solution only slightly, so that the computation time for inventory optimization is divided in a way that enforces at least three iterations (i.e., perturbations) of inventory iterated local search, in order to examine at least a few different inventory configurations.

4.4 Optimization of assignments $\mathbf{z}$

Let X_{iq}, Y_i^k $(i \in \mathrm{V}', q \in \mathrm{Q}, k \in \mathrm{K})$ be a given facility-inventory configuration. In order to evaluate these facility and inventory decisions, and subsequently identify the best one, we must calculate the assignment of demand locations to the facilities. This allows us to assess the objective function value of a solution, i.e., to calculate its met demand and response time. The greedy assignment of vertices that is performed in the heuristic to guide the inventory optimization gives only an approximation of the met demand and response time that can be achieved with the given facility-inventory configuration, and as such can miss some promising configurations. To reduce the possibility of that happening, we use CPLEX to further optimize the assignments. In order to not compromise the heuristic efficiency, we optimize the assignments only of the local facility-inventory optima, rather than of every solution (Figure 3). The optimal assignment of demand vertices to open facilities for each scenario is calculated using CPLEX by solving the two models below.

$$\max \quad \sum_{i \in \mathrm{V}'} \sum_{j \in \mathrm{V}} \mathcal{D}_j^{s_0} z_{ij}^{s_0} \tag{15}$$

$$\sum_{i' \in \mathrm{V}} z_{ij}^{s_0} \leq 1 \qquad (j \in \mathrm{V}) \tag{16}$$

$$\sum_{j \in \mathrm{V}} D_j^{ks_0} z_{ij}^{s_0} \leq R_i^{ks_0} Y_i^k \qquad (i \in \mathrm{V}')(k \in \mathrm{K}) \tag{17}$$

$$z_{ij}^{s} \leq 1 + \mathcal{L}_{ij}^{s_0} \qquad (i \in \mathrm{V}')(j \in \mathrm{V}) \tag{18}$$

$$\sum_{i \in \mathrm{V}'} \sum_{j \in \mathrm{V}} \mathcal{C}_{ij}^{s_0} z_{ij}^{s_0} \leq C \tag{19}$$

$$z_{ij}^{s_0} \in \{0, 1\} \qquad (i \in \mathrm{V}')(j \in \mathrm{V}) \tag{20}$$

The objective of the model is to maximize the average percentage of met demand (15), similarly to the objective function (1) of the original model, but for a fixed scenario $s = s_0$. In the given scenario, the constraints (16), (17), (18) (19), (20) correspond to constraints (3), (4), (5), (8), and (11) respectively.

Let $\mathcal{D}^*(s_0)$ be the optimal met demand in scenario s_0, i.e., the objective function value of the optimal solution obtained by solving the model above. To further optimize the response time in lexicographic order, it remains to solve the following model.

$$\min \quad \sum_{i \in \mathrm{V}'} \sum_{j \in \mathrm{V}} \mathcal{T}_{ij}^{s_0} z_{ij}^{s_0} \tag{21}$$

$$(16) - (20) \tag{22}$$

$$\sum_{i \in \mathrm{V}'} \sum_{j \in \mathrm{V}} \mathcal{D}_j^{s_0} z_{ij}^{s_0} \geq \mathcal{D}^*(s_0) \tag{23}$$

The objective of the model is to minimize the response time (21) so that the standard constraints (22) are satisfied, together with the constraint that ensures that the met demand is (greater than or) equal to the previously obtained optimal met demand in scenario $s_0 \in \mathrm{S}$ (23).

4.5 Termination criterion

We choose a given maximum computation time as the termination criterion of the matheuristic, for a number of reasons. First of all, even for medium-sized problem instances, only the facility local search as the first heuristic step can take a very long time, as there can be many facility neighbours to explore. Most of the other termination criteria, such as the number of iterations (without improvement) could let the matheuristic

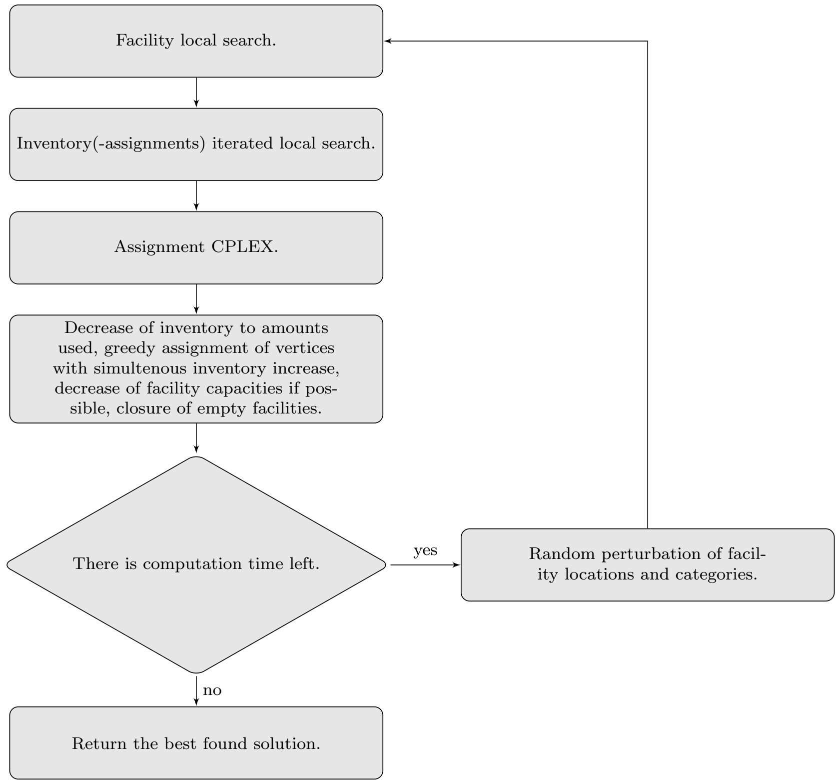


Fig. 3 iterated_local_search(*solution*). In the improvement phase, we optimize the facility and inventory configuration using an iterated local search metaheuristic, while the assignments are optimized using CPLEX.

run for a very long time (even for a very limited number of iterations). Secondly, computation time seems as the most straightforward stopping criterion since it does not require the user to have any understanding of algorithms in general or the matheuristic elements in particular (e.g., notion of an iteration). Finally, such a stopping criterion offers the most flexibility to the user as it allows them to run the heuristic as long as time permits. If, for example, the practitioners want to carry out a sensitivity analysis on the effect of budgets on the solution quality and solve the problem for thousands of budget combinations, they might want to allow only a few seconds per each run, whereas they can allow more computation time for finding a solution for the chosen budget combination.

Before we explain how the given maximum computation time is divided between different heuristic elements described in previous subsections, we take a look at how they are combined (Figure 3). As mentioned before, the optimization of assignments would ideally be enclosed within the inventory iterated local search that would further be enclosed within the facility iterated local search. In other words, every change in the inventory levels would be followed by finding the optimal assignments, and every change in the facility configuration would be followed by a complete inventory and assignments optimization. Since this would be computationally too demanding (and most often unnecessary), every change in the facility or inventory is only followed by a greedy assignment with simultaneous inventory increase (Figure 2, right), except for some small instances where facility configurations are evaluated by the greedy assignment followed by an inventory iterated local search (Section 4.2). The greedy procedure serves to approximately evaluate if the change in the facility or inventory yielded an improvement, or to reasonably complete a solution after a perturbation. It is for this reason that the inventory and assignments of the initial solution are not optimized: although better, the facility neighbours in the facility local search evaluated only with a greedy assignment (Figure 3, node 1) could hardly compete with the optimized initial solution.

However, in order to provide for a better evaluation of at least the promising facility and inventory configurations, every facility and inventory local optimum is further optimized by (inventory and) assignments optimization to better identify the global optimum. In order to improve the matheuristic efficiency, no computation efforts are wasted on inventory and assignment optimization of the poor facility local optima (with the gap from the current best solution greater than 20%). In this case, the facility perturbation is not performed on the (poor) last facility local optimum (Section 4.2), but rather on the best found solution.

In order to use the computation time allocated to each heuristic element effectively, some adaptations to these elements are made. If not given enough computation time, for most problem instances, the facility local search would not examine every facility neighbour. It is therefore a good idea to prune the search by starting with promising neighbours, and we do this by trying to close the facilities where the average percentage of unused goods across scenarios is the greatest, as this might allow for opening new facilities where the supplies could be exploited more effectively. Furthermore, the time allocated to optimizing the assignments is often not sufficient to find the optimal solution, but we let CPLEX find the best solution within the given time limit. We start by optimizing the assignments in the scenario with the greatest probability (in order to decrease the unmet demand and response time the most), letting CPLEX use the computation time designated for the optimization of assignments. The remaining time is used for optimizing other scenarios in the same order. In every scenario, the first half of the computation time is invested in minimizing unmet demand, and the other half in minimizing response time. Obviously, the assignment found by CPLEX is only accepted if it is better than the greedy assignment.

Lastly, a simple final step is performed after employing CPLEX to optimize the assignments. Indeed, even if the optimal assignments were found, the inventory (that remains unchanged during assignments optimization) is often not being used completely. Decreasing the inventory to the amounts used can gain some savings in the inventory costs, that can be utilized to store more supplies at some open facilities and thereby enable serving a few additional vertices. For every facility-inventory-assignments local optimum, we therefore decrease the inventory to the amounts used and perform the greedy assignment of vertices with a simultaneous inventory increase (Figure 3, node 4).

5 Numerical experiments

In this section, we carry out numerical experiments to tune the distribution of the total computation time between the facility, inventory and assignments optimization (Section 5.1), and to evaluate the performance of the matheuristic (Section 5.2) for the best identified computation time distribution.

The only instances available for the particular stochastic capacitated facility location problem we are aware of are the pre-positioning problem case studies (Figure 4) and random instances described in detail in [75]. The case studies were constructed from the instances originally introduced in [10], [6], [73] and [60], which focus on disasters of different type and scale that occurred in different parts of the world. To tune the computation time distribution, we employ 75% of each group of case studies and random instances, and use the remaining 25% of the instances to evaluate the matheuristic performance.

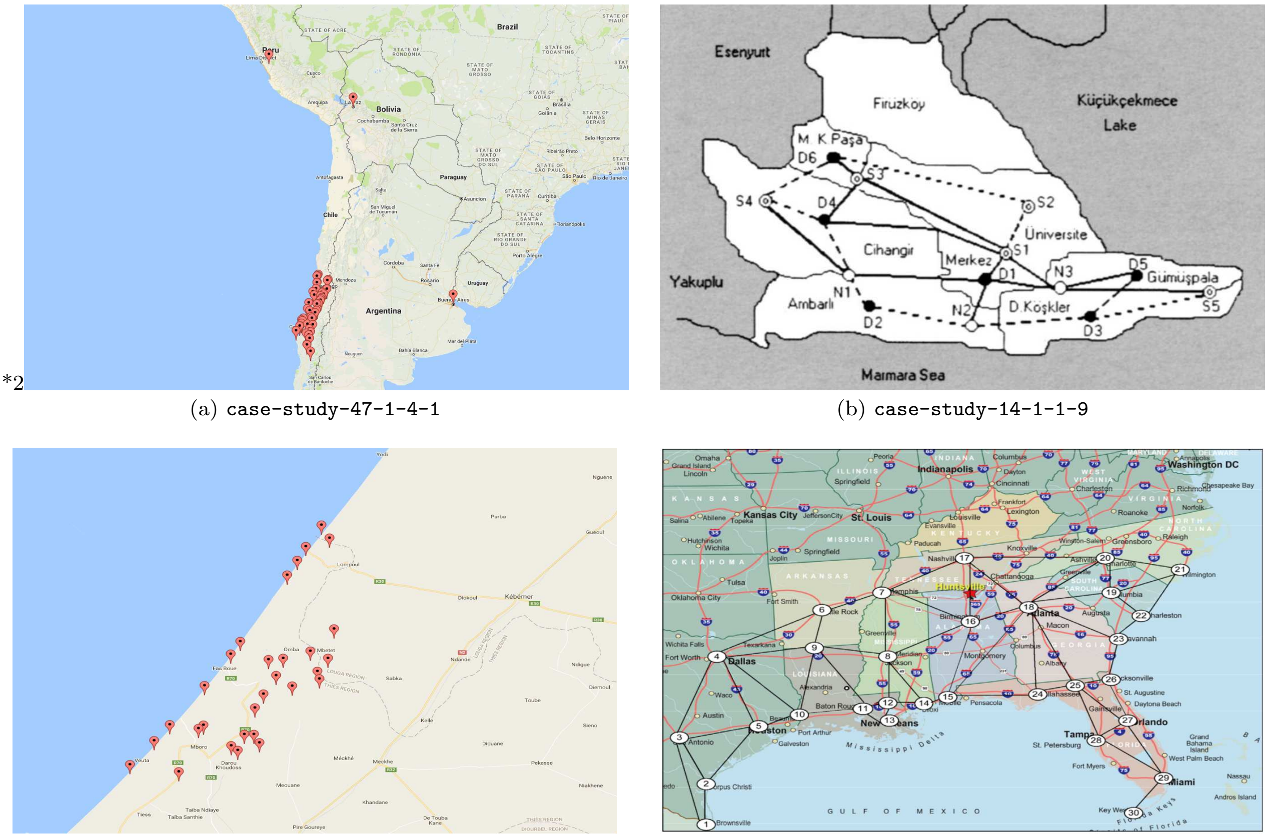
5.1 Tuning the distribution of the total computation time

The matheuristic termination criterion is the total computation time (Section 4.5). In order to determine how to distribute the total computation time between the time given to the optimization of facility, inventory and assignment decisions, we run the matheuristic for every selected pre-positioning problem instance according to 16 distributions of the total computation time:

- only one set of decisions is optimized (distributions 1-0-0, 0-1-0, 0-0-1),
- only two sets of decisions are optimized, with the total computation time equally distributed between them (distributions 0.5-0.5-0, 0.5-0-0.5, 0-0.5-0.5),
- every set of decisions is optimized, and
  - the total computation time is distributed equally (0.33-0.33-0.34),
  - one set of decisions is optimized for 50% of the total computation time, and the remaining computation time is equally distributed between the optimization of the other two sets of decisions (0.5-0.25-0.25, 0.25-0.5-0.25, 0.25-0.25-0.5),
  - the greatest computation time allocated for the optimization of one set of decisions is double the computation time for the other set of decisions, which is then double the computation time for the optimization of the remaining set of decisions (0.57-0.29-0.14, 0.57-0.14-0.29, 0.29-0.57-0.14, 0.14-0.57-0.29, 0.29-0.14-0.57, 0.14-0.29-0.57).

For example, the distribution 0.57-0.29-0.14 means that 57% of the computation time is allocated to facility optimization, 29% to inventory optimization, and the remaining 14% to the assignments optimization.

There are two issues that need to be addressed to properly assess the quality of each computation time distribution. Since the greedy heuristic is not deterministic and therefore might return different initial solutions across different runs (Section 4.1), we start the experiment by running the greedy heuristic once to find *an* initial solution (otherwise, good or poor performance of a particular matheuristic computation time distribution might



(a) `case-study-47-1-4-1` (b) `case-study-14-1-1-9`

(c) `case-study-30-1-1-10` (d) `case-study-30-3-3-51`

Fig. 4 Network graphs $G_1 = (V, E_1)$ of the 4 base case studies that inspired the generation of the 30 pre-positioning case studies: (a) Chile 2010 earthquake and tsunami (b) Turkey 1999 earthquake (c) Senegal Mboro region disaster threat (d) US Gulf Coast hurricane threat.

be accredited to a good or poor starting solution). For the next 5 minutes, the matheuristic tries to improve the initial solution according to the given computation time distribution.

Secondly, although the matheuristic employs CPLEX to optimize the assignment sub-problem (decisions $\mathbf{z}$), CPLEX is given a limited computation time and therefore the assignments are not necessarily optimal. If, for example, two solutions returned by the matheuristic are reported to have unmet demand 21% and 22%, it might be the case that the unmet demand (of the solutions, with optimal assignments) is 20.5% and 20% respectively, so that the latter solution is better. As already mentioned, solving the considered facility problem means deciding on the facility and inventory configurations, whereas the assignments are made only to evaluate the quality of those decisions. After running the matheuristic for 5 minutes (see above), we therefore give CPLEX additional 5 minutes to optimize the assignments decisions and thereby better assess the quality of the solutions returned by different computation time distributions. Since the improvement of the solution quality with respect to the initial solution varies across different problem instances, we evaluate the different computation time distributions with the unmet demand improvement, relative to the best improvement for a given problem instance.

Table 4 registers the performance of the average of 5 runs of the matheuristic for different distributions of the total computation time of 5 minutes, averaged across the 75% of selected instances introduced in [75]. As already mentioned, CPLEX is given additional 5 minutes to optimize the assignments of the solution returned by the matheuristic, in order to better assess the quality of the solution. There is a number of different total computation time distributions that perform similarly well, because the relative unmet demand improvements are averaged across various problem instances, and different matheuristic components are particularly important for different instances. However, it is clear that the worst-performing are the distributions that do not include any facility or inventory optimization. It is also important to note that further optimizing the assignments (the *math* part of our heuristic) in order to better identify promising facility-inventory local optima is shown to have an added value.

In order to properly interpret the tuning results, it is important to have in mind how the facility, inventory and assignment optimization are combined together (Section 4.5). For example, the distribution 0.57-0.29-0.14 means that at *maximum* 57% of total computation time is spent on facility *local search*, 29% is spent on inventory

Table 4 Tuning the distribution of the total computation time between different matheuristic components on 75% of the 30 case studies and 10 random instances shows that it is best to allocate 29% of the time to facility local search, 14% to the inventory iterated local search, and the remaining 57% to the further optimization of the assignments (or facility iterated local search iterations).

Facility optimization	0.29	0.57	0.25	0.14	0.33	0.14	0.57	0.50	0.29	0.25	0.50	0.50	1.00	0.00	0.00	0.00
Inventory optimization	0.14	0.14	0.25	0.29	0.33	0.57	0.29	0.25	0.57	0.50	0.50	0.00	0.00	0.50	1.00	0.00
Assignments optimization	0.57	0.29	0.50	0.57	0.34	0.29	0.14	0.25	0.14	0.25	0.00	0.50	0.00	0.50	0.00	1.00
Relative unmet demand improvement	0.90	0.89	0.89	0.88	0.87	0.87	0.86	0.86	0.86	0.84	0.82	0.77	0.75	0.68	0.60	0.33

(*and assignments*) iterated local search, and CPLEX is given at *maximum* 14% of the total computation time to further optimize the assignments of the facility-inventory local optima (in order to choose better between them). Since the inventory optimization is directly guided by the changes in the assignments (Section 4.3), even the matheuristic under 0.5-0.5-0 includes some (heuristic) optimization of the assignments. Since CPLEX often does not take a lot of time to optimize the assignments, the success of the distribution 0.14-0.29-0.57 should not necessarily be attributed to the assignments optimization - it is rather possible that additional facility perturbations are responsible for the improvements in the solution.

Note that this means that, although further optimization of assignments by CPLEX can improve the matheuristic performance, the simplified heuristic 0.5-0.5-0 without the exact solver is not much worse, compared to the best distribution of computation time. The iterated local search can thus replace the matheuristic if there is a need for a simple solution procedure.

Overall, we can conclude that the best strategy is to allocate a limited proportion of the total computation time to the inventory optimization, and use the remaining time to optimize the facility and assignments decisions. One of the best computation time distributions is 0.57-0.14-0.29, which uses the majority of the computation time for the facility local search. The other best performing distributions allocate the majority of the computation time to the assignment optimization, but as previously mentioned, CPLEX often uses less computation time to optimize the assignments, so that the remainder of the time can be used for further iterations of the facility iterated local search. The facility moves yield the greatest change in the solution, and it is therefore not surprising that facility optimization is crucial.

5.2 Matheuristic performance evaluation

In this section, we evaluate the performance of the matheuristic for the best identified computation time distribution 0.29-0.14-0.57, on the remaining 25% of the each group of case studies (Figure 4) and random instances introduced in [75].

An instance with |V| vertices, |Q| facility categories, |K| commodity types and |S| possible scenarios is named `instance`-|V|-|Q|-|K|-|S|. Before we describe the experimental set-up and results, we note here that the notion of instance size or complexity is not straightforward for the particular stochastic facility location problem. Some obvious numerical parameters that give an idea about the "size" of an instance are the aforementioned number of vertices, number of facility categories, number of commodity types or number of scenarios. However, there are other properties that can significantly influence the instance complexity. For example, the number of potential facility locations obviously influences the complexity of finding good facility configurations. Furthermore, in the set of instances with the same values of aforementioned numerical parameters, one instance might have only a few demand vertices in each scenario, while every vertex can be a demand location in every scenario for another instance. For instance, the group of case studies based on `case-study-30-1-1-10` and `case-study-30-3-3-51` both have 30 vertices, but each of them have non-zero demand in every disaster scenario for `case-study-30-1-1-10`, but there is only a few non-zero demand vertices in each scenario for `case-study-30-3-3-51`. The facility capacity can also influence the difficulty of finding near-optimal solutions, but also the strictness of the facility, inventory and transportation budgets A, B, and C. In the extreme cases where these budgets are all zero or all extremely large (i.e., more than sufficient to meet all demand), solving the instance (with any aforementioned numerical parameters) is trivial, and varying these budgets might take CPLEX between less than a second to a few days to solve to optimality (unless it runs out of memory).

Since the facility problem with the same assumptions has not yet been solved for the same problem instances in the literature, we cannot compare our solutions to the results from other articles. We therefore use CPLEX to help us evaluate the matheuristic performance. Table 5 gives an overview of the performance of the greedy constructive heuristic, and the best solution found by the matheuristic and by employing CPLEX to solve the models that minimize unmet demand (1)-(11), and minimize unmet demand and response time in lexicographic order (1)-(11), (12)-(14). The matheuristic is given 5 minutes (divided to different components according to the 0.29-0.14-0.57 distribution), and CPLEX is given a maximum computation time of 6 hours. As already mentioned, we evaluate the matheuristic performance on the remaining 25% of the each group of case studies (Figure 4) and random instances introduced in [75], but for the sake of completeness, we show the results for all problem instances. For every instance and every solution procedure, the averages across 5 runs for are reported. When employing CPLEX to find the solution that minimizes the two objectives in lexicographic order, solving the first model that minimizes unmet demand is limited to the first 3 hours, while solving the second model that minimizes the response time is limited to the remaining 3 hours. When solving the second model, we provide CPLEX with the solution with the best found unmet demand as the starting solution, as in this case even finding a feasible solution is difficult. We also supply CPLEX with an initial solution in the matheuristic: the greedy assignment is provided as the initial solution when CPLEX looks for the assignment with minimum unmet demand, and the assignment with the minimum unmet demand is provided to the model that looks for the assignment with minimum response time. Whenever CPLEX is used to minimize both objectives and solving the first model that minimizes unmet demand returns a solution with zero response time, the same solution is immediately returned by the second model that minimizes response time.

For the case studies (which are, on average, the smaller and simpler in the set of considered problem instances), CPLEX reports a small MIP gap (Table 5), implying that it is able to find a solution that is (close to) optimal within 6 hours of computation time. However, due to the numeric difficulties that CPLEX cannot handle, this gap is calculated to be 0% for some of the instances, although the matheuristic is able to find better solutions. For larger random instances, CPLEX is not even able to build the model within 6 hours of given computation time, or it runs out of memory, and therefore the best found solution is the trivial solution with unmet demand percentage of 100% and response time of 0 hours.

The experimental results demonstrate a promising matheuristic performance (Table 5). For the instances where CPLEX can find a near-optimal solution within 6 hours, the matheuristic finds solutions in only 5 minutes with the unmet demand that is the same or not much greater (e.g., on average 0.32% from the optimal solution for the first group of instances with 14 vertices, 1 facility type, 1 commodity type and 9 scenarios). As expected, the matheuristic becomes crucial for larger instances, as it is able to return a good solution for any instance in a limited computation time. The matheuristic does not face any numeric difficulties and can thus prove beneficial even for simple instances. Overall, for the highlighted validation set of problem instances listed in Table 5, the matheuristic yields solutions with the unmet demand that is, on average, 14.92% lower than the solutions calculated by CPLEX when minimizing unmet demand, and 23.03% lower than the solutions calculated by CPLEX when minimizing unmet demand and response time (and 18.18% and 34.14% respectively across the complete set of instances introduced in [75]).

The response time as the secondary objective is reported in Table 5, but any comparison of the solution procedures with respect to response time is senseless, as they yield solutions that do not have the same unmet demand.

The numerical results also show that the greedy constructive procedure itself builds a very good initial solution very fast. Actually, the facility-inventory configuration (the actual decisions of the stochastic facility problem, e.g., the actual pre-positioning decisions made during the disaster preparedness phase) of the initial solution is better than it might appear in Table 5. Indeed, the assignments of the initial solution are never optimized to more accurately evaluate the initial facility and inventory decisions, in order to allow a fair comparison with the facility-inventory configurations explored in the facility local search that are only evaluated with a greedy assignment (Section 4.2). Letting CPLEX optimize the assignments of the initial solution for a few seconds offers a better evaluation of the unmet demand, that is often significantly lower than the unmet demand of the initial solution given in the Table 5 (the same is true for the quality of the facility and inventory decisions yielded by the matheuristic).

We also compare the performance of the matheuristic to that of CPLEX, for the same amount of computation time. Figure 5 depicts the average across all instances (and over 5 runs) for 30, 300, 600, 1200, 2700 and 3600 seconds, and clearly shows that the matheuristic outperforms CPLEX for any of the given computation times.

Table 5 Given a maximum computation time of 5 minutes, the matheuristic yields solutions with the unmet demand that is 14.92% lower on average across the validation set of problem instances, compared to the strategies obtained by CPLEX within 6 hours. This gap is even greater (23.03%) if we compare the matheuristic results to the results obtained by CPLEX when minimizing both objectives in lexicographic order. The greedy constructive heuristic is very fast and produces solutions with the unmet demand that is on average 3.82% greater from the unmet demand of the best found solution the matheuristic returns after 5 minutes.

Instance	Greedy heuristic			Matheuristic		CPLEX min unmet demand				CPLEX min unmet demand and time				
	Unmet demand (%)	Response time (h)	Runtime (s)	Unmet demand (%)	Response time (h)	Unmet demand (%)	Unmet demand gap (%)	Response time (h)	Runtime (s)	Unmet demand (%)	Unmet demand gap (%)	Response time (h)	Response time gap (%)	Runtime (s)
case-study-14-1-1-9	29.45	0.33	0.01	28.83	0.44	28.71	0.01	0.40	6.77	28.71	0.01	0.40	0.00	16.72
case-study-14-1-1-9_1	27.73	0.51	0.01	25.80	0.69	25.40	0.01	0.72	6.15	25.40	0.01	0.71	0.00	28.86
case-study-14-1-1-9_2	56.02	0.18	0.01	41.84	0.21	41.84	0.00	0.21	1.78	41.84	0.00	0.21	0.00	2.08
case-study-14-1-1-9_3	17.66	0.37	0.01	17.30	0.38	17.39	0.01	0.40	3.14	17.38	0.01	0.37	0.00	6.14
case-study-14-1-1-9_4	58.38	0.18	0.01	46.66	0.25	45.18	0.01	0.27	2.25	45.18	0.01	0.27	0.00	3.53
case-study-14-1-1-9_5	15.02	0.79	0.01	14.39	0.59	14.39	0.01	0.68	3.13	14.39	0.01	0.59	0.00	5.98
case-study-30-1-1-10	34.85	5.46	0.05	34.81	7.95	34.81	0.01	7.28	6.52	100.00	–	0.00	–	–
case-study-30-1-1-10_1	39.13	4.97	0.05	39.07	7.80	39.07	0.00	7.49	2.26	100.00	–	0.00	–	–
case-study-30-1-1-10_2	12.06	10.92	0.05	12.01	9.98	12.02	0.01	13.29	28.92	100.00	–	0.00	–	–
case-study-30-1-1-10_3	17.70	13.59	0.07	17.67	11.85	17.67	0.01	18.45	473.06	100.00	–	0.00	–	–
case-study-30-1-1-10_4	42.34	5.31	0.06	37.44	10.12	37.44	0.14	11.39	21714.90	100.00	–	0.00	–	–
case-study-30-1-1-10_5	54.34	12.20	0.03	52.04	18.47	51.76	0.07	20.03	21715.50	51.76	0.07	19.23	7.58	20630.30
case-study-30-1-1-10_6	52.51	16.96	0.05	50.89	22.10	48.35	0.16	31.19	21756.20	54.97	0.19	7.91	6.19	15128.10
case-study-30-1-1-10_7	22.36	10.29	0.06	17.23	9.49	17.20	0.01	13.99	212.06	17.20	0.01	9.83	0.01	786.51
case-study-30-1-1-10_8	23.96	9.09	0.06	19.13	10.39	18.96	0.01	11.33	16156.50	100.00	–	0.00	–	–
case-study-30-3-3-51	14.38	77.91	1.51	14.10	72.54	13.76	1.09	96.04	21755.60	100.00	–	0.00	–	–
case-study-30-3-3-51_1	15.65	85.41	1.38	14.21	84.82	13.82	1.15	98.71	21777.90	100.00	–	0.00	–	–
case-study-30-3-3-51_2	18.74	92.22	0.71	18.46	95.71	18.10	2.26	94.37	21600.10	18.12	2.98	75.36	53.60	21605.90
case-study-30-3-3-51_3	29.51	64.82	0.55	28.41	70.64	27.43	0.01	67.35	1946.39	27.43	0.01	65.21	0.00	2662.23
case-study-30-3-3-51_4	51.11	0.61	0.49	39.36	2.24	36.69	0.00	1.96	27.10	36.69	0.00	1.51	0.00	79.24
case-study-30-1-3-51	39.24	0.49	0.41	39.24	0.49	39.24	0.00	1.13	23.02	39.24	0.00	0.45	0.00	70.06
case-study-47-1-4-1	21.49	1666.12	0.05	19.82	2907.64	100.00	0.00	0.00	0.40	100.00	0.00	0.00	–	0.42
case-study-47-1-4-6_1	31.68	1312.32	0.30	29.68	2587.19	37.73	0.00	3445.32	3.54	37.70	0.00	876.24	0.01	11.37
case-study-47-1-4-6_2	55.29	638.74	0.31	53.29	1918.59	100.00	0.00	0.00	2.53	100.00	0.00	0.00	–	2.53
case-study-47-1-4-6_3	28.40	1494.88	0.31	27.32	2202.40	100.00	0.00	0.00	1.54	100.00	0.00	0.00	–	1.97
case-study-47-1-4-6_4	53.27	766.67	0.31	52.44	1738.19	100.00	0.00	0.00	1.97	100.00	0.00	0.00	–	2.20
case-study-47-2-4-1_1	21.44	1667.28	0.06	19.76	2962.28	100.00	0.00	0.00	0.37	100.00	0.00	0.00	–	0.32
case-study-47-2-4-1_2	41.51	1062.20	0.05	41.47	2131.50	100.00	0.00	0.00	0.15	100.00	0.00	0.00	–	0.31
case-study-47-2-4-6_1	49.14	843.62	0.32	49.10	1241.62	54.39	0.00	2906.61	2.29	54.38	0.00	429.62	0.01	5.05
case-study-47-2-4-6_2	66.73	498.19	0.30	66.70	869.77	100.00	0.00	0.00	2.02	100.00	0.00	0.00	–	1.87
rand-instance-50-2-4-50	43.32	139.82	1.03	30.13	109.94	56.81	0.00	119.45	405.93	56.80	0.00	3.60	0.01	563.53
rand-instance-50-3-1-100	55.24	240.36	0.24	55.24	242.90	55.24	0.00	242.22	508.10	55.24	0.00	238.48	0.01	2741.61
rand-instance-50-3-3-100	45.92	226.12	2.33	40.99	266.16	100.00	0.00	0.00	553.94	100.00	0.00	0.00	–	1086.48
rand-instance-100-2-1-30	83.94	23.92	0.33	83.93	24.84	83.93	0.00	24.65	224.65	83.93	0.00	23.69	0.01	2362.43
rand-instance-100-2-3-200	47.71	195.67	9.49	41.41	234.86	100.00	–	0.00	21600.00	100.00	–	0.00	–	10800.00
rand-instance-100-3-3-50	28.98	117.18	6.04	22.40	108.13	19.86	0.00	386.83	4013.93	19.86	0.00	39.38	0.01	10023.60
rand-instance-100-3-3-100	79.93	72.70	1.27	79.87	73.42	80.66	0.00	84.84	8219.22	100.00	–	0.00	–	10800.00
rand-instance-200-2-1-50	72.37	266.23	0.49	53.14	365.16	100.00	–	0.00	21600.00	100.00	–	0.00	–	10800.00
rand-instance-200-3-3-100	18.94	116.93	36.06	18.94	116.93	100.00	–	0.00	21600.00	100.00	–	0.00	–	10800.50
rand-instance-200-3-5-200	70.09	257.28	15.82	66.09	341.89	100.00	–	0.00	21600.10	100.00	–	0.00	–	10800.10
Average validation set	41.50	173.75	5.35	37.68	259.37	52.60	0.00	293.60	5962.85	60.71	0.00	44.23	0.00	2505.87
Average	39.69	300.47	2.02	36.52	522.01	54.70	0.14	192.67	6239.00	70.66	0.12	44.83	3.55	4119.69

Next to the instances, the complete solutions for any solution procedure and computation time listed in Table 5 or Figure 5 can be found on the following website:

`http://antor.uantwerpen.be/members/renata-turkes/`.

Based on the experimental results, we can see that a very simple way to improve our matheuristic would be to let the matheuristic run for most of the given computation time, but to also allocate a very limited amount of time (only a few seconds) for CPLEX. The final solution would of course be chosen as the better of the two solutions, yielded by the matheuristic and by CPLEX. Such a heuristic has the best of both worlds: it will identify the optimal solution for small instances which CPLEX can solve to optimality, find good solutions even for large instances and detect CPLEX numeric difficulties for any instance.

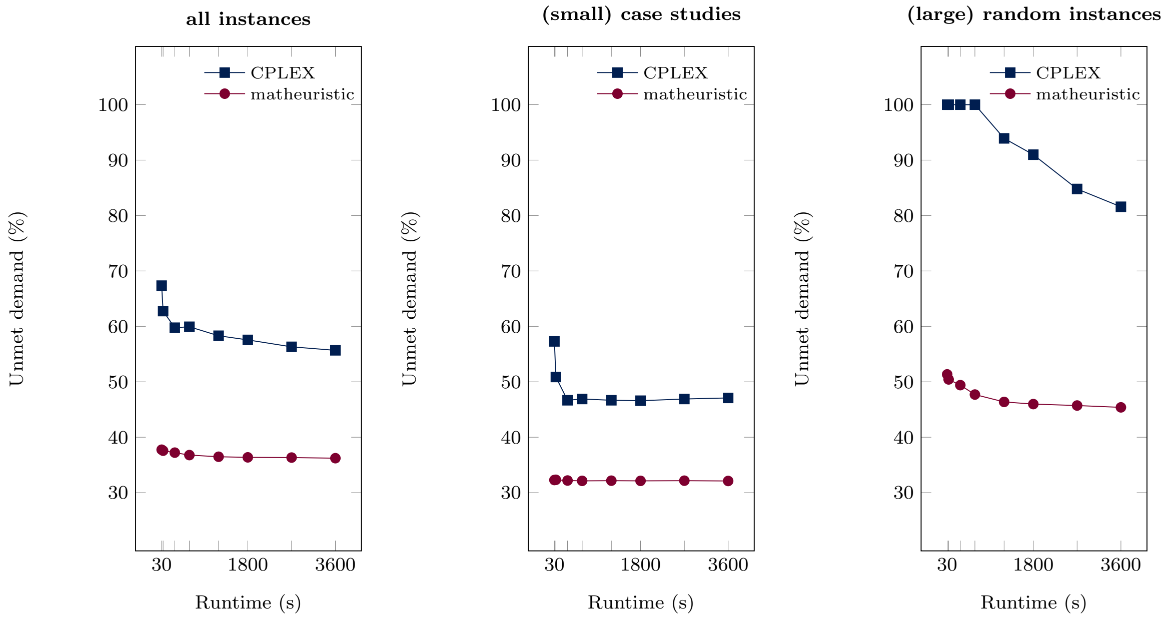


Fig. 5 The matheuristic outperforms CPLEX for any given computation time, especially for large instances.

6 Conclusions, limitations and future research

The main aim of this work has been to develop an effective and efficient procedure that can solve a realistic version of the stochastic facility location problem. With realistic, we mean both a sensible problem formulation, and problem instances of reasonable size and complexity.

In our problem formulation, we have incorporated a number of complex elements, such as both the facility and inventory decisions, multiple facility and commodity types, uncertainties related to the demands, survival of stored supplies and transportation network availability. The objective is to minimize the unmet demand and response time in lexicographic order.

To be able to solve instances of reasonable size and complexity that most often become intractable for exact methods, we have developed a matheuristic algorithm that is able to find solutions also for very large instances. In addition, even though the facility problem is strategic, it is also inherently multi-objective and thus the decision-making process requires solving the problem many times. For this reason, a heuristic procedure often becomes necessary even for smaller instances and the efficiency of the heuristic is a requisite.

The matheuristic we developed is based on the iterated local search procedure, with the assignment sub-problem intermittently solved with an exact solver. The second-stage assignment sub-problem assigns demand locations to the storage facilities that meet those demands, and is solved in order to evaluate the first-stage facility and inventory decisions. The efficiency of the heuristic is guaranteed by the maximum computation time that serves as the heuristic termination criterion. Given any amount of computation time that the user might have at their disposal, the heuristic is thus able to find a solution even for very large problem instances. The effectiveness of the heuristic is demonstrated by comparing its performance to that shown by CPLEX for a number of diverse case studies and random instances of different sizes. For any given computation time, the matheuristic produces solutions that are, on average, significantly better than solutions calculated by CPLEX; in particular for larger instances where CPLEX fails to find non-trivial solutions in any reasonable computation time. Since these problem instances were not yet solved in the literature, our numerical experiments offer benchmark solutions for this set of case studies and random instances.

In addition, the numerical results show that the greedy constructive procedure itself builds a very good initial solution very fast. The greedy heuristic is simple and intuitive and can be used to even manually calculate good solutions and thus provides a rule of thumb for facility planning. In addition, for different parameters of the constructive heuristic (the initial solution is chosen to be the best out of a number of solutions), different strategies are obtained, which allows experimentation with a pool of solutions that can incorporate, e.g., political or other factors that cannot be taken into account in a mathematical model. Also, the quality of any given facility configuration can be estimated extremely quickly by the greedy assignment with simultaneous inventory increase.

The stochastic facility problem under study corresponds to the formulation of the problem of pre-positioning emergency supplies. The first-stage decisions correspond to pre-disaster decisions about the location and category of storage facilities and amounts of different types of aid to be stored, and the second-stage distribution subproblem assigns people in need of assistance to the storage facilities that meet their needs. The objectives directly reflects the priorities of disaster relief: provide assistance to as many people as possible, as soon as possible. The matheuristic in this paper can be therefore use to further study the pre-positioning problem and to find good emergency strategies. Due to a lack of strong mathematical background or computational infrastructure common in humanitarian settings, the simplicity of the heuristic and the good performance of the greedy heuristic can be of particular importance as they can be directly used to guide disaster preparedness planning.

Although the heuristic can be used by practitioners for pre-disaster decision making, simpler rules of thumb are more easily integrated in practice; such straightforward guidelines are indeed what most practitioners rely on [82]. The matheuristic introduced in this paper can therefore be used to also derive general guidelines for practitioners trough carefully designed experiments that study the effects of some disaster properties on the solution structure. For example, it would be interesting to investigate how the demand network topology (i.e., if the disaster is localized or dispersed), the level of transportation network damage and/or the relationship between the capacities and costs of different facility types or different budgets influence the number, type and location of storage facilities to open. Paring down the models and solution techniques in such manner into simple guidelines that workers can use on the ground, is one way of responding to the challenge of carrying theory into practice that has been widely acknowledged in the humanitarian logistics literature.

Acknowledgments

We thank the Interuniversity Attraction Poles (IAP) Programme on Combinatorial Optimization: Metaheuristics and Exact Methods (COMEX), initiated and funded by the Belgian Science Policy Office (BELSPO), for supporting this research. We also thank Yasemin Arda and Sibel Salman for their suggestions that helped improve the numerical experiments in this paper. In particular, we are grateful to the anonymous referees for taking the time to go through our manuscript in detail, and for their constructive feedback that significantly improved this work.

Appendix: Matheuristic steps on a toy example

An example of a toy instance for the considered stochastic facility location problem with 3 cities, 2 facility categories, 2 commodities and 2 scenarios is given in Table 6 and Figure 6. We assume that vertices $i = 2$ and $i = 3$ are the potential facility locations, i.e., they belong to the set V'. The facility, acquisition and transportation budgets are $A = 18\,000$, $B = 500\,000$ and $C = 10\,000$ respectively. The average speed is $W = 50$, and the distances in the two disaster scenarios are the following:

$$L^1 = \begin{pmatrix} 0 & 300 & 400 \\ 350 & 0 & 410 \\ 400 & 400 & 0 \end{pmatrix}, \quad L^2 = \begin{pmatrix} 0 & 300 & 400 \\ 350 & 0 & -1 \\ 420 & -1 & 0 \end{pmatrix}.$$

The transportation networks are represented by directed graphs and we therefore do not impose any restrictions on the symmetry of the distance matrices. If the distance between vertices $i \in \mathrm{V}$ and $j \in \mathrm{V}$, and $j \in \mathrm{V}$ and $i \in \mathrm{V}$, are not the same (e.g., if there are multiple one-way roads of different lengths connecting the vertices, and/or one direction is not traversable due to debris or a flood), the distance matrices might be asymmetric.

Table 6 Toy instance facility categories that might be opened at potential facility locations differ in their volume capacity M_q and opening cost A_q. For the different types of goods that are in demand and might be stored at open facilities, their unit volume M^k, unit acquisition cost B^k and unit transportation cost C^k are given.

Facility category q	M_q	A_q
1 (small facility)	5 00	8 000
2 (big facility)	2 000	18 000

Commodity k	M^k	B^k	C^k
1 (water)	2	200	0.1
2 (food)	0.4	1250	0.08

The initial solution is chosen as the best of the number of solutions built using a greedy constructive heuristic, across different values of m, which is a parameter of this heuristic. To construct a solution for a given m, we first need to decide where to open the facilities, and of which category (Figure 2, left). For each potential facility location, i.e., for each vertex $i \in \mathrm{V}'$, we calculate $\mathcal{X}_i(m)$ - the expected average percentage of demand of m

Scenario $s = 1$, $P^1 = 0.9$ Scenario $s = 2$, $P^2 = 0.1$

Fig. 6 Toy instance graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ represent three cities and the road network that connects them in two possible scenarios, with potential facility locations denoted by a square. The scenarios occur with probabilities P^1 and P^2 respectively, and both are defined with the demand D_i^{ks} and percentage of inventory that remains usable R_i^{ks} for every commodity $k \in \mathrm{K}$ and every vertex $i \in \mathrm{V}$, together with the availability of every edge that is indicated in the graph. For example, the demand at vertex $i = 3$ for commodity $k = 1$ in scenario $s = 2$ is $D_3^{12} = 500$ units of that commodity; if any amount of commodity $k = 1$ would be stored at a facility open at vertex $i = 3$, only $R_3^{12} = 0.1 = 10\%$ of the stored goods would remain usable in scenario $s = 2$ and could be distributed to beneficiaries; the edge between vertices $i = 2$ and $i = 3$ is not traversable in scenario $s = 2$ ($L_{23}^2 = L_{32}^2 = -1$, because the transportation link is, e.g., flooded or covered in debris).

neighbouring vertices, taking the percentage of goods that remains usable at the given vertex into account. For $m = 1$, we have:

$$\begin{aligned}
\mathcal{X}_2(1) &= 0.9\Big(0.7 \times \frac{100}{70+100+0} + 0.6 \times \frac{45}{20+45+0}\Big) + 0.1\Big(0.8 \times \frac{900}{400+900+500} + 0.9 \times \frac{250}{130+250+90}\Big)\\
&= 0.83,\\
\mathcal{X}_3(1) &= 0.9\Big(0.9 \times \frac{0}{70+100+0} + 0.5 \times \frac{0}{20+45+0}\Big) + 0.1\Big(0.1 \times \frac{500}{400+900+500} + 0.2 \times \frac{90}{130+250+90}\Big)\\
&= 0.01,
\end{aligned}$$

so that we decide to open the first facility at vertex $i = 2$. To decide which category of facility to open here, we look at the amounts of commodity $k \in \mathrm{K}$ that need to be stored at vertex $i = 2$ to meet the demand of $m = 1$ neighbouring vertices in scenario $s \in \mathrm{S}$:

$$\begin{aligned}
\mathcal{Y}_2^{11}(1) &= \lceil 100/0.7 \rceil = 143,\\
\mathcal{Y}_2^{21}(1) &= \lceil 45/0.6 \lceil = 75,\\
\mathcal{Y}_2^{12}(1) &= \lceil 900/0.8 \lceil = 1\,125,\\
\mathcal{Y}_2^{22}(1) &= \lceil 250/0.9 \lceil = 278.
\end{aligned}$$

In order to be able to meet the demands of neighbouring vertices in all scenarios, the amounts of different commodities to be stored at vertex $i = 2$ would be

$$\begin{aligned}
\mathcal{Y}_2^1(1) &= \max\{\mathcal{Y}_2^{11}(1), \mathcal{Y}_2^{12}(1)\} = 1\,125,\\
\mathcal{Y}_2^2(1) &= \max\{\mathcal{Y}_2^{21}(1), \mathcal{Y}_2^{22}(1)\} = 278.
\end{aligned}$$

The category of facility that we aim to open is therefore the category with minimum capacity that is able to store the volume of goods above:

$$\mathcal{V}_2(1) = M^1 \mathcal{Y}_2^1(1) + M^2 \mathcal{Y}_2^2(1) = 2 \times 1\,125 + 0.4 \times 278 = 2361.2$$

Since there is no facility category with such a large capacity, we open the biggest facility available, thus $x_{22} = 1$. Opening this facility uses up all the available facility budget, so that no more facilities can be open.

Next, we need to decide how much inventory to store at the open facilities, and which demand vertices are allocated to which open facilities (Figure 2, right). The constructive heuristic performs this assignment also in a greedy fashion according to the demand (in order to increase the primary objective the most), simultaneously increasing the inventory after each assignment (if necessary). We therefore calculate the weighted percentage of demand, multiplied by the scenario probability, $P^s \times \mathcal{D}_i^s$ for each demand vertex $j \in \mathrm{V}$ in each scenario $s \in \mathrm{S}$:

$$P^1 \times \mathcal{D}_1^1 = 0.9 \times \Big(\frac{70}{70+100+0} + \frac{20}{20+45+0}\Big) = 0.65,$$
$$P^1 \times \mathcal{D}_2^1 = 0.9 \times \Big(\frac{100}{70+100+0} + \frac{45}{20+45+0}\Big) = 1.15,$$
$$P^2 \times \mathcal{D}_1^2 = 0.1 \times \Big(\frac{400}{400+900+500} + \frac{130}{130+250+90}\Big) = 0.05,$$
$$P^2 \times \mathcal{D}_2^2 = 0.1 \times \Big(\frac{900}{400+900+500} + \frac{250}{130+250+90}\Big) = 0.10,$$
$$P^2 \times \mathcal{D}_3^2 = 0.1 \times \Big(\frac{500}{400+900+500} + \frac{90}{130+250+90}\Big) = 0.05,$$

This means that we start by assigning vertex $j = 1$ in scenario $s = 1$. A vertex is assigned to the closest open facility where possible, in order to increase the secondary objective the least, and to keep the transportation costs below the budget limit. In the toy example, vertex $j = 1$ is assigned in scenario $s = 1$ to the only open facility at vertex $i = 2$ ($z_{21}^1 = 1$), and the inventory is increased accordingly, in order to enable the demands of the considered vertex in the considered scenario to be met:

$$y_2^1 = \lceil 70/0.7 \rceil = 100,$$
$$y_2^2 = \lceil 20/0.6 \rceil = 34.$$

It is easy to check that this solution is feasible, i.e., it also satisfies the facility capacity, and acquisition and transportation budget constraints. We continue in this manner, checking if it is possible to assign more vertices, proceeding from the vertex with the next greatest $P^s \times \mathcal{D}_i^s$.

After building a number of solutions for different values of the greedy heuristic parameter m, we choose the best one as our initial solution. This solution is further optimized using iterated local search procedure as described in detail in Sections 4.2, 4.3, 4.4 (closing and opening facilities, or assigning and unassigning vertices to explore the solution spaces), with the assignments also intermittently optimized using a solver.

References

1. Aikens, C.H.: Facility location models for distribution planning. European Journal of Operational Research **22**(3), 263–279 (1985)
2. Alem, D., Clark, A., Moreno, A.: Stochastic network models for logistics planning in disaster relief. European Journal of Operational Research **255**(1), 187–206 (2016)
3. Balcik, B., Beamon, B.M.: Facility location in humanitarian relief. International Journal of Logistics **11**(2), 101–121 (2008)
4. Balcik, B., Beamon, B.M., Smilowitz, K.: Last mile distribution in humanitarian relief. Journal of Intelligent Transportation Systems **12**(2), 51–63 (2008)
5. Balcik, B., Bozkir, C.D.C., Kundakcioglu, O.E.: A literature review on inventory management in humanitarian supply chains. Surveys in Operations Research and Management Science **21**(2), 101–116 (2016)
6. Barbarosoglu, G., Arda, Y.: A two-stage stochastic programming framework for transportation planning in disaster response. Journal of the Operational Research Society **55**(1), 43–53 (2004)
7. Barzinpour, F., Esmaeili, V.: A multi-objective relief chain location distribution model for urban disaster management. The International Journal of Advanced Manufacturing Technology **70**(5-8), 1291–1302 (2014)
8. Birge, J.R., Louveaux, F.: Introduction to stochastic programming. Springer Science & Business Media (2011)
9. de Brito Junior, I., Leiras, A., Yoshizaki, H.T.Y.: Stochastic optimization applied to the prepositioning of disaster relief supply decisions in Brazil. Tech. rep., POMS 24th Annual Conference (2013)
10. Camacho-Vallejo, J.F., González-Rodríguez, E., Almaguer, F.J., González-Ramírez, R.G.: A bi-level optimization model for aid distribution after the occurrence of a disaster. Journal of Cleaner Production **105**, 134–145 (2015)
11. Campbell, A.M., Jones, P.C.: Prepositioning supplies in preparation for disasters. European Journal of Operational Research **209**(2), 156–165 (2011)
12. Chan, Y., Carter, W.B., Burnes, M.D.: A multiple-depot, multiple-vehicle, location-routing problem with stochastically processed demands. Computers & Operations Research **28**(8), 803–826 (2001)
13. Chapman, J., Davis, L.B., Samanlioglu, F., Qu, X.: Evaluating the effectiveness of pre-positioning policies in response to natural disasters. International Journal of Operations Research and Information Systems **5**(2), 86–100 (2014)
14. Charles, A.: Improving the design and management of agile supply chains: feedback and application in the context of humanitarian aid. Ph.D. thesis, Institut National Polytechnique de Toulouse (2010)
15. Cooper, L.: Heuristic methods for location-allocation problems. SIAM review **6**(1), 37–53 (1964)
16. Cooper, L.: A random locational equilibrium problem. Journal of Regional Science **14**(1), 47–54 (1974)
17. Cornuéjols, G., Nemhauser, G., Wolsey, L.: The uncapicitated facility location problem. Tech. rep., Cornell University Operations Research and Industrial Engineering (1983)
18. Daskin, M.S.: Network and discrete location: models, algorithms, and applications. John Wiley & Sons (2011)
19. Daskin, M.S., Coullard, C.R., Shen, Z.J.M.: An inventory-location model: Formulation, solution algorithm and computational results. Annals of operations research **110**(1-4), 83–106 (2002)

20. Davis, L.B., Samanlioglu, F., Qu, X., Root, S.: Inventory planning and coordination in disaster relief efforts. International Journal of Production Economics **141**(2), 561–573 (2013)
21. Döyen, A., Aras, N., Barbarosoğlu, G.: A two-echelon stochastic facility location model for humanitarian relief logistics. Optimization Letters **6**(6), 1123–1145 (2012)
22. Duran, S., Gutierrez, M.A., Keskinocak, P.: Pre-positioning of emergency items for CARE international. Interfaces **41**(3), 223–237 (2011)
23. Erlenkotter, D.: A dual-based procedure for uncapacitated facility location. Operations Research **26**(6), 992–1009 (1978)
24. Farahani, R.Z., Hekmatfar, M.: Facility location: concepts, models, algorithms and case studies. Springer (2009)
25. Farahani, R.Z., SteadieSeifi, M., Asgari, N.: Multiple criteria facility location problems: A survey. Applied Mathematical Modelling **34**(7), 1689–1709 (2010)
26. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canadian Journal of Mathematics **8**(3), 399–404 (1956)
27. Francis, R.L., McGinnis, L.F., White, J.A.: Locational analysis. European Journal of Operational Research **12**(3), 220–252 (1983)
28. Gabor, A.F., van Ommeren, J.C.: An approximation algorithm for a facility location problem with stochastic demands and inventories. Operations Research Letters **34**(3), 257–263 (2006)
29. Goldberg, A.V., Tarjan, R.E.: A new approach to the maximum-flow problem. Journal of the ACM (JACM) **35**(4), 921–940 (1988)
30. Gösling, H., Geldermann, J.: Methodological tool kit for humanitarian logistics. In: ISCRAM (2014)
31. Grass, E., Fischer, K.: Two-stage stochastic programming in disaster management: A literature survey. Surveys in Operations Research and Management Science (2016)
32. Grass, E., Fischer, K., Rams, A.: An accelerated L-shaped method for solving two-stage stochastic programs in disaster management. Annals of Operations Research pp. 1–26 (2018)
33. Gutjahr, W.J., Dzubur, N.: Bi-objective bilevel optimization of distribution center locations considering user equilibria. Transportation Research Part E: Logistics and Transportation Review **85**, 1–22 (2016)
34. Hakimi, S.L.: Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research **12**(3), 450–459 (1964)
35. Holguín-Veras, J., Jaller, M., Van Wassenhove, L.N., Pérez, N., Wachtendorf, T.: On the unique features of post-disaster humanitarian logistics. Journal of Operations Management **30**(7), 494–506 (2012)
36. Holguín-Veras, J., Pérez, N., Jaller, M., Van Wassenhove, L.N., Aros-Vera, F.: On the appropriate objective function for post-disaster humanitarian logistics models. Journal of Operations Management **31**(5), 262–280 (2013)
37. Holguín-Veras, J., Pérez, N., Ukkusuri, S., Wachtendorf, T., Brown, B.: Emergency logistics issues affecting the response to Katrina: A synthesis and preliminary suggestions for improvement. Transportation Research Record: Journal of the Transportation Research Board (2022), 76–82 (2007)
38. Hong, X., Lejeune, M.A., Noyan, N.: Stochastic network design for disaster preparedness. IIE Transactions **47**(4), 329–357 (2015)
39. Jia, H., Ordóñez, F., Dessouky, M.M.: Solution approaches for facility location of medical supplies for large-scale emergencies. Computers & Industrial Engineering **52**(2), 257–276 (2007)
40. Kall, P., Wallace, S.W., Kall, P.: Stochastic programming. Springer (1994)
41. Katz, I.N., Cooper, L.: An always-convergent numerical scheme for a random locational equilibrium problem. SIAM Journal on numerical analysis **11**(4), 683–692 (1974)
42. Kelle, P., Schneider, H., Yi, H.: Decision alternatives between expected cost minimization and worst case scenario in emergency supply. International Journal of Production Economics **157**, 250–260 (2014)
43. Khayal, D., Pradhananga, R., Pokharel, S., Mutlu, F.: A model for planning locations of temporary distribution facilities for emergency response. Socio-Economic Planning Sciences **52**, 22–30 (2015)
44. Kınay, Ö.B., Saldanha-da Gama, F., Kara, B.Y.: On multi-criteria chance-constrained capacitated single-source discrete facility location problems. Omega (2018)
45. Krarup, J., Pruzan, P.: Selected families of location problems. In: Annals of Discrete Mathematics, vol. 5, pp. 327–333. Elsevier (1979)
46. Laporte, G., Louveaux, F.V., van Hamme, L.: Exact solution to a location problem with stochastic demands. Transportation Science **28**(2), 95–103 (1994)
47. Leonardi, G.: A unifying framework for public facility location problems—part 1: A critical overview and some unsolved problems. Environment and Planning a **13**(8), 1001–1028 (1981)
48. Li, L., Jin, M., Zhang, L.: Sheltering network planning and management with a case in the Gulf Coast region. International Journal of Production Economics **131**(2), 431–440 (2011)
49. Listeş, O., Dekker, R.: A stochastic approach to a case study for product recovery network design. European Journal of Operational Research **160**(1), 268–287 (2005)
50. Lourenço, H.R., Martin, O.C., Stutzle, T.: Iterated local search. International Series in Operations Research and Management Science pp. 321–354 (2003)
51. Louveaux, F.V.: Discrete stochastic location models. Annals of Operations research **6**(2), 21–34 (1986)
52. Louveaux, F.V., Peeters, D.: A dual-based procedure for stochastic facility location. Operations Research **40**(3), 564–573 (1992)
53. Manne, A.S.: Capacity expansion and probabilistic growth. Econometrica: Journal of the Econometric Society pp. 632–649 (1961)
54. Melo, M.T., Nickel, S., Saldanha-Da-Gama, F.: Facility location and supply chain management–a review. European journal of operational research **196**(2), 401–412 (2009)
55. Mirchandani, P.B., Odoni, A.R.: Locations of medians on stochastic networks. Transportation Science **13**(2), 85–97 (1979)
56. Mirchandani, P.B., Oudjit, A., Wong, R.T.: 'multidimensional'extensions and a nested dual approach for the m-median problem. European Journal of Operational Research **21**(1), 121–137 (1985)
57. Orlin, J.B.: Max flows in O(nm) time, or better. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pp. 765–774. ACM (2013)
58. Owen, S.H., Daskin, M.S.: Strategic facility location: A review. European journal of operational research **111**(3), 423–447 (1998)
59. Rabbani, M., Manavizadeh, N., Samavati, M., Jalali, M.: Proactive and reactive inventory policies in humanitarian operations. Uncertain Supply Chain Management **3**(3), 253–272 (2015)

60. Rawls, C.G., Turnquist, M.A.: Pre-positioning of emergency supplies for disaster response. Transportation Research Part B: Methodological **44**(4), 521–534 (2010)
61. Rawls, C.G., Turnquist, M.A.: Pre-positioning planning for emergency response with service quality constraints. OR Spectrum **33**(3), 481–498 (2011)
62. Rennemo, S.J., Rø, K.F., Hvattum, L.M., Tirado, G.: A three-stage stochastic facility routing model for disaster response planning. Transportation Research Part E: Logistics and Transportation Review **62**, 116–135 (2014)
63. Revelle, C., Marks, D., Liebman, J.C.: An analysis of private and public sector location models. Management Science **16**(11), 692–707 (1970)
64. ReVelle, C.S., Eiselt, H.A.: Location analysis: A synthesis and survey. European Journal of Operational Research **165**(1), 1–19 (2005)
65. Rezaei-Malek, M., Tavakkoli-Moghaddam, R.: Robust humanitarian relief logistics network planning. Uncertain Supply Chain Management **2**(2), 73–96 (2014)
66. Richardson, D.A., De Leeuw, S., Dullaert, W.: Factors affecting global inventory prepositioning locations in humanitarian operations - a Delphi study. Journal of Business Logistics **37**(1), 59–74 (2016)
67. Rottkemper, B., Fischer, K., Blecken, A., Danne, C.: Inventory relocation for overlapping disaster settings in humanitarian operations. OR Spectrum **33**(3), 721–749 (2011)
68. Shen, Z.J.M., Coullard, C., Daskin, M.S.: A joint location-inventory model. Transportation Science **37**(1), 40–55 (2003)
69. Snyder, L.V.: Facility location under uncertainty: a review. IIE Transactions **38**(7), 547–564 (2006)
70. Tansel, B.C., Francis, R.L., Lowe, T.J.: State of the art—location on networks: a survey. part i: the p-center and p-median problems. Management science **29**(4), 482–497 (1983)
71. Tansel, B.C., Francis, R.L., Lowe, T.J.: State of the art—location on networks: a survey. part ii: exploiting tree network structure. Management Science **29**(4), 498–511 (1983)
72. Tofighi, S., Torabi, S.A., Mansouri, S.A.: Humanitarian logistics network design under mixed uncertainty. European Journal of Operational Research **250**(1), 239–250 (2016)
73. Tricoire, F., Graf, A., Gutjahr, W.J.: The bi-objective stochastic covering tour problem. Computers & Operations Research **39**(7), 1582–1592 (2012)
74. Turkeš, R., Cuervo, D.P., Sörensen, K.: Pre-positioning of emergency supplies: does putting a price on human life help to save lives? Annals of Operations Research **283**(1-2), 865–895 (2019). DOI 10.1007/s10479-017-2702-1
75. Turkeš, R., Sörensen, K.: Instances for the problem of pre-positioning emergency supplies. Journal of Humanitarian Logistics and Supply Chain Management **9**(2), 172–195 (2019). DOI 10.1108/JHLSCM-02-2018-0016
76. Uichanco, J.: A robust model for pre-positioning emergency relief items before a typhoon with an uncertain trajectory. Submitted to Manufacturing & Service Operations Management
77. Ukkusuri, S., Yushimito, W.: Location routing approach for the humanitarian prepositioning problem. Transportation Research Record: Journal of the Transportation Research Board (2089), 18–25 (2008)
78. Van Hentenryck, P., Bent, R., Coffrin, C.: Strategic planning for disaster recovery with stochastic last mile distribution. In: International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming, pp. 318–333 (2010)
79. Van Slyke, R.M., Wets, R.: L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics **17**(4), 638–663 (1969)
80. Vasant, P.M.: Meta-heuristics optimization algorithms in engineering, business, economics, and finance. IGI Global (2012)
81. Vinck, P.: World disasters report: Focus on technology and the future of humanitarian action: International federation of red cross and red crescent societies. Tech. rep. (2013)
82. de Vries, H., Van Wassenhove, L.N.: Evidence-based vehicle planning for humanitarian field operations. Working Paper No. 2017/62/TOM/Social Innovation Centre, INSEAD (2017). `https://ssrn.com/abstract=3039320`
83. Weaver, J.R., Church, R.L.: Computational procedures for location problems on stochastic networks. Transportation Science **17**(2), 168–180 (1983)
84. Weber, A.: Ueber den standort der industrien. (1909)