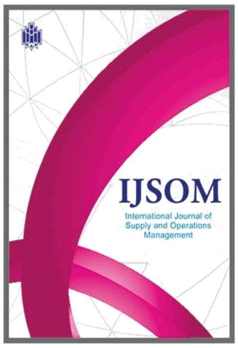
International Journal of Supply and Operations Management

IJSOM

2023, Volume 10, Issue 2, pp. 151-173
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



Implementing Solution Algorithms for a Bi-level Optimization to the Emergency Warehouse Location-allocation Problem

Ehsan Saghehei[a], Azizollah Memariani [b,*], Ali Bozorgi-Amiri [c]

[a] *Department of Industrial Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran*
[b] *Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran*
[c] *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*

Abstract

The aim of this paper was to develop a binary bi-level optimization model for the emergency warehouse location-allocation problem in terms of national and regional levels. This type of modeling is suitable for countries where the design of the disaster emergency network is decentralized. The upper-level decision-maker makes a decision regarding the location and allocation of national warehouses through considering the location of regional warehouses and allocating them to the demand cities. Each regional warehouse can provide a service for the demand cities within a specified distance threshold, ultimately affecting the efficiency of the solution algorithms. The optimization model parameters were calculated in terms of the real data in Iran. To solve the small size problem, an exact method was proposed from the explicit complete enumeration. Due to the complexity of the model with the large size, two innovative hybrid genetic algorithms, namely HG-ES-1 and HG-ES-2, were suggested. The results obtained from solving the problems showed that the HG-ES-1 algorithm outperformed HG-ES-2. The findings further indicated the proper functioning of the solution approaches.

Keywords: Bi pre-positioning; Disaster management; Bi-level programming; Hybrid genetic algorithms; Location-allocation problem.

1. **Introduction and Literature review**

Over the past two decades, many crises such as natural and man-made disasters have occurred with a huge amount of human and financial losses. In 2017, 335 natural disasters took place, killing more than 9697 people worldwide, impacting over 95.6 million others, and causing US$ 335 billion of economic damages(CRED, 2018). Disasters ensue a condition where a large number of people require immediate relief supplies. It is necessary to pre-position relief supplies in the disaster emergency network so as to quickly and efficiently respond to demand. Among the objectives of this network is to procure and store relief supplies and distribute them in the time of disaster. Locating and allocating warehouses and distribution centers are regarded as key elements for designing disaster emergency networks. Boonmee et al. (2017) summarized and categorized facility location optimization models in the relief supply chain field.

In some country, when a national decision-maker (upper-level) locates and allocates emergency facilities based on single-level optimization, the regional decision-maker (lower-level) may, by leveraging, allocate facilities to areas under his/her management. This leads to unfair allocation of facilities and the reduced effectiveness of planning. Using bi-level programming is one approach to preventing the disruption of the upper-level decision-makers (central government) by the lower-level decision-makers (regional managers) related to the emergency facility location problem. In this paper, a

*Corresponding author email address: memariani@khu.ac.ir
DOI: 10.22034/ijsom.2022.108930.2065

bi-level programming was used to model the location and allocation of emergency warehouses at national and regional levels. The optimization model of this research has been developed based on Saghehei et al. (2021). In the present model, the distance threshold constraint is considered in both levels for warehouses. In the following, some studies focused on the location and allocation of emergency facilities are categorized and presented in two sections: single-level and bi-level programming.

1.1. Single-level emergency warehouses location

Balcik and Beamon (2008) developed a model that specified the location of distribution centers and the amount of relief supplies inventory at each distribution center in the international humanitarian network. In their innovative work, Manopiniwes et al. (2014) proposed a mix integer model with capacity constraints and time restrictions. Based on a real flood case in Thailand, the optimization model minimized the total cost of relief tasks. Mokhtarinejad et al. (2015) developed the P-center optimization model for locating emergency warehouses. Their proposed model included certain features such as population distribution, economic condition, and transportation. Moreover, real data were gathered, and the number of Chinese national emergency warehouses was specified. Regarding emergency warehouse location problem, some studies have considered uncertainty in modeling. Bozorgi-Amiri et al. (2013) introduced a stochastic model comprised of two stages: the first one was set to locate relief distribution centers and inventory quantities for each item type while the second aimed for the number of items transported from distribution centers to demand points. In another study, Bozorgi-Amiri and Khorsi (2016) developed a multi-objective stochastic programming model that considered the minimization of the unsatisfied demand, travel time, and costs as objective functions of the model. Paul and Hariharan (2012) addressed three groups of delays to receive and allocate supplies in the SNS network. A new framework was further presented to increase the efficiency of the disaster preparedness plan based on the characteristics of disasters. Paul and MacDonald (2016b) developed a stochastic optimization model to specify the location and capacities of medical supplies; their model was designed for storm conditions using historical hurricane data. Paul and MacDonald (2016a) used the framework during the earthquake disaster. The model considered various uncertainties such as facility damage and casualty losses according to their severity and remaining survivability time as a function of earthquake magnitude.

1.2. Bi-level emergency warehouse location

Camacho-Vallejo et al. (2015) proposed a bi-level programming model to optimize the location of distribution centers in humanitarian logistics. The affected country and non-profit international organizations were considered to be the upper-level and the lower-level, respectively. Upper and lower-level's objectives were to minimize response time and the shipping cost. They considered the 2010 earthquake in Chile as their case study. Gutjahr and Dzubur (2016) developed a multi-objective bi-level optimization model for locating the distribution centers in a relief supply chain. The aid-providing organization was considered as the leader and the beneficiaries as followers. The leader's objectives were to minimize the total opening cost for the distribution centers and total uncovered demand, and the followers' objective was to provide user equilibrium related to the leader. Xu et al. (2016) proposed a multi-objective bi-level programming for the location-routing problem in post-earthquake phase. The leader (Rescue Control Center) decided on the location of distribution centers, and the follower (Logistics Company) selected an optimal rout to achieving relief supplies from distribution center. Road conditions were considered as a source of uncertainty (random fuzzy variables) in this optimization model. Safaei et al. (2018) utilized a bi-level programming model to locate distribution centers and select suppliers. The leader's objective function was to minimize operational costs and uncoated demands, and that of the lower-level was to minimize the risk of supplier choice. Haeri et al. (2020) developed a bi-level multi-objective programming approach for pre-positioning relief items. The DEA method was used to determine candidate places for locating relief centers. They utilized a fuzzy goal programming algorithm to solve the bi-level model.

1.3. Contribution of the research

In the suggested model, location and allocation of facilities are done at both levels in which the location and allocation of the upper-level depend on the location and allocation of the lower-level and vice versa. Hence, the reciprocal dependency of decision variables of the levels makes the model more complex to solve. With this end in view, two innovative hybrid evolutionary approaches, namely HG-ES-1 and HG-ES-2 with different initial solution generation and allocation methods have been developed. The main contributions of this research can be summarized as follows:

- Developing a new binary bi-level model based on minimizing the total weight of distances in order to locate and allocate national and regional warehouses in the disaster emergency network
- Proposing an exact algorithm based on full enumeration method to solve the binary bi-level location-allocation models for small size problem
- Designing the genetic algorithms to solve the binary bi-level location-allocation models with a large number of variables
- Comparing evolutionary algorithms to bi-level programming by innovative evaluation criteria

1.4. Manuscript organization

The rest of the paper is organized in the following order: Section 2 reviews the general structure and the solutions to discrete bi-level programming model. In Section 3, the problem is described, and a bi-level optimization model is developed for the emergency warehouse location-allocation problem. Section 4 provides the details of the algorithms proposed to solve the optimization model. Section 5 describes the computational results obtained from the solution approaches. The final section concludes this paper and offers future recommendations.

2. The Discrete Bi-level Programming

Bi-level programming is a specific type of multi-level programming for decentralized planning in a hierarchical organization with several decision-makers. This programming is closely associated with Stackelberg games, consisting of two levels (upper and lower level): the decision-makers at the upper-level and the lower-level are called leader and follower, respectively (J. Bard, 1998). In bi-level programming, the leader primarily attempts to optimize his/her objectives by considering the objective of the follower; the follower then decides to optimize his/her objectives while being aware of the decision taken by the leader. This process continues until an equilibrium point is reached (Bracken & McGill, 1973). A subset of this group of optimization problems is discrete bi-level programming, which formulation is the same as that of bi-level programming along with one or more variables at either of the levels being integer. The discrete bi-level model is presented as follows:

Leader:

$$\begin{aligned}
&\min_{x\in X} F(x,y)\\
&s.t \quad G(x,y)\le 0\\
&\textit{Where, for every } x \textit{ fixed, } y \textit{ solves} \qquad\qquad (1)
\end{aligned}$$

Follower:

$$\begin{aligned}
&\min_{y\in Y} f(x,y)\\
&s.t \quad g(x,y)\le 0\\
&x, y \in \text{int}
\end{aligned}$$

, where $F(x, y)$ and $f(x, y)$ are objective functions and $G(x, y)$ and $g(x, y)$ denote constraints for the leader and the follower. Solution methods of bi-level programming are classified into classical and evolutionary algorithms. Single-level reduction, descent methods, penalty function methods, and trust-region methods are among the known classical methods. Evolutionary algorithms are categorized into nested, meta-modeling, and single-level reduction problems for solving bi-level problems (Sinha et al., 2017). The following includes some studies on the classical and evolutionary algorithms for discrete bi-level programming.

2.1. Classical algorithms for discrete bi-level programming

Some of the early studies on solving discrete bi-level models by classical algorithms are in (Dempe, 1996; Vicente et al., 1996). Furthermore, J. F. Bard and Moore (1992) employed an implicit enumeration for pure binary bi-level problem. Dempe (2002) analyzed the feasible region of the discrete bi-level programming and presented the Cutting-Plan algorithm to solve the discrete bi-level programming. Branch-and-bound and branch-and-cut are two commonly employed techniques for solving this type of problem (Sinha et al., 2017). These methods were used to solve the discrete bi-level optimization models. Benders decomposition is one of the methods used in solving bi-level models with mixed-integer variables (Caramia & Mari, 2016; Fontaine & Minner, 2014; Kheirkhah et al., 2016; Yue et al., 2019).

2.2. Evolutionary algorithms for discrete bi-level programming

Moore and Bard (1990) proved that the mixed-integer bi-level programming problems were NP-hard. Despite the attempts made to develop classical methods for discrete bi-level programs, the research is still open to new methods and ideas as none of the proposed techniques would work well for problems with larger number of variables. Different types of evolutionary algorithms have also been used to solve discrete bi-level programming models (Aksen & Aras, 2013; Camacho-Vallejo et al., 2014; Hecheng & Yuping, 2008; Kheirkhah et al., 2015; Miandoabchi & Farahani, 2011). Huang and Liu (2004) developed an interactive evolutionary framework for the mixed integer bi-level programming in the location-allocation problem. They employed genetic algorithm for the lower-level model and enumeration vertex method for the upper-level model. Sinha et al. (2014) designed nested evolutionary approach to solve multi-period multi-leader-follower Stackelberg game. Calvete et al. (2013) utilized a nested genetic algorithm to solve a binary bi-level programming model in the ring star problem. Chen et al. (2017) developed an improved differential evolution algorithm

(IDE) to solve a binary bi-level model for emergency warehouse location-allocation problem. The computational results of IDE were compared with the results of conventional differential evolution algorithms (CoDE, SaDE, JADE, and JDE).

3. Problem description

In this paper, the disaster emergency network was assumed to comprise three stages (see, Fig. 1). The first stage includes a set of national warehouses, the second contains regional warehouses, and the last is comprised of demand cities. The regional warehouses receive their relief supplies from national warehouses. Demand cities are serviced only from regional warehouses, and the direct shipment of goods from the national warehouses to demand cities is prohibited. Bi-level programming is considered as a modeling framework. Upper-level and lower-level problems are associated with the national decision-makers and regional decision-makers, respectively. In our proposed model, the upper-level model made a decision regarding the location and allocation of national warehouses through considering the location of regional warehouses and allocating them to demand cities. As shown in Fig.1, each of the national and regional warehouse candidates, at a road distance threshold, is capable of serving a lower level.

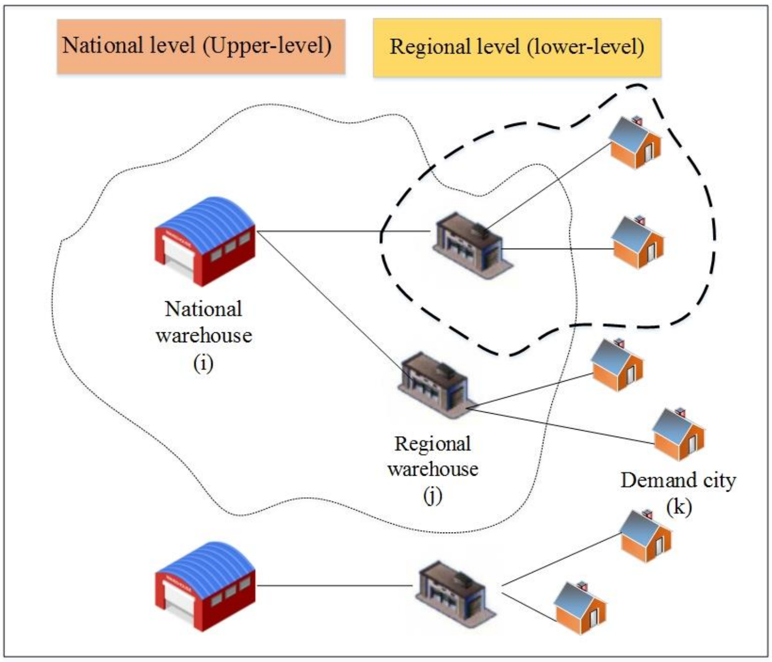


Figure 1. Bi-level emergency warehouse location-allocation problem (BL-EW-LAP)

3.1. The BL-EW-LAP optimization model:

Index and sets:

I set of candidate cities for national warehouses $I = \{1, \ldots, m\}$

J set of candidate cities for regional warehouses $J = \{1, \ldots, n\}$

K set of demand cities $k = \{1, \ldots, k\}$

parameters:

urd_{ij} road distance of national warehouse i from regional warehouse j

lrd_{jk} road distance of regional warehouse j from demand city k

ldw_k demand weight associated to demand city k

$ucap_i$ capacity of national warehouse i

$lcap_j$ capacity of regional warehouse j

$umax$ number of national warehouses to establish

$lmax$ number of regional warehouses to establish

utr_i the maximum distance that national warehouse i can serve regional warehouses

ltr_j the maximum distance that regional warehouse j can serve demand cities

Decision variable:

$Z_i = 1$ if a national warehouse is established at candidate city i, and 0 otherwise.

$X_{ij} = 1$ if regional warehouse j is allocated to national warehouse i, and 0 otherwise.

$\overline{Z}_j = 1$ if a regional warehouse is established at candidate city j, and 0 otherwise.

$Y_{jk} = 1$ if demand city k is allocated to regional warehouse j and 0 otherwise.

$udw_j = \sum_k ldw_k * Y_{jk}$ total demand weight associated to regional warehouse j

Upper-level model (ULM):

Min $$\sum_i \sum_j udw_j urd_{ij} X_{ij} \tag{2}$$

Subject to: $$\sum_i Z_i \le umax \tag{3}$$

$$\sum_i X_{ij} = 1 \quad \forall j \tag{4}$$

$$\sum_j udw_j X_{ij} \le ucap_i Z_i \quad \forall i \tag{5}$$

$$\sum_k ldw_k Y_{jk} = \sum_i udw_j X_{ij} \quad \forall j \tag{6}$$

$$urd_{ij} X_{ij} \le utr_i \quad \forall i, j \tag{7}$$

$$Z_i, X_{ij} \in \{0,1\}, \quad \forall i, j \tag{8}$$

Lower-level model (LLM):

Min $$\sum_j \sum_k ldw_k lrd_{jk} Y_{jk} \tag{9}$$

Subject to: $$\sum_j \overline{Z}_j \le lmax \tag{10}$$

$$\sum_j Y_{jk} = 1 \quad \forall k \tag{11}$$

$$\sum_k ldw_k Y_{jk} \le lcap_j \overline{Z}_j \quad \forall j \tag{12}$$

$$\sum_k ldw_k Y_{jk} = \left(\sum_i udw_j X_{ij}\right) \overline{Z}_j \quad \forall j \tag{13}$$

$$lrd_{jk} Y_{jk} \le ltr_j \quad \forall j, k \tag{14}$$

$$\overline{Z}_j, Y_{jk} \in \{0,1\}, \forall j, k \tag{15}$$

Equations (2-8) and (9-15) are related to ULM and LLM, respectively. The objective functions of the ULM and LLM are to minimize the total weighted distances between national and regional warehouses and regional warehouses and demand cities, respectively (equations 2 and 9). The maximum number of national warehouses that can be established on the candidate sites is equal to $umax$ (equation 3). From equation 4, it is guaranteed that each located regional warehouse j is likely to be allocated to national warehouse i. Equation (5) is to ensure that the regional warehouse allocated to the national warehouse will not exceed the capacity. Equation (6) is a balance constraint for each regional warehouse; more specifically, the amount of demand weights of cities allocated to a regional warehouse should be equal to the value allocated from the national warehouse. Moreover, equation 6 is the coupling constraint of the upper-level model to the lower-level. Equation (7) applies to the distance threshold for each national warehouse. If the regional warehouse is not within the coverage area of the national warehouse, that warehouse cannot be allocated to the national warehouse. The interpretation of equations (10-14) is similar to that of equations (3-7). Equations (8) and (15) set binary conditions and domain of decision variables for the bi-level model.

3.2. The relaxed of BL-EW-LAP (RXBL)

The relaxed version of a BL-EW-LAP can be defined as an optimization model comprised of the objective function of the upper-level and the constraints of both levels. In this model, the objective function of the lower-level model is eliminated, and the single-level model is used. Due to the minimization of the objective function of the bi-level model, the optimal solution of the relaxed version of the bi-level model is considered as the lower-bound of the original bi-level problem (Talbi, 2013). Formulation of the relaxed version of the bi-level model is shown as follows:

$$\text{Min} \quad \sum_i \sum_j udw_j urd_{ij} X_{ij} \tag{16}$$

Subject to:

$$\sum_i Z_i \le umax \tag{17}$$

$$\sum_i X_{ij} = 1 \qquad \forall j \tag{18}$$

$$\sum_j udw_j X_{ij} \le ucap_i Z_i \qquad \forall i \tag{19}$$

$$\sum_k ldw_k Y_{jk} = \sum_i udw_j X_{ij} \qquad \forall j \tag{20}$$

$$urd_{ij} X_{ij} \le utr_i \qquad \forall i, j \tag{21}$$

$$\sum_j \overline{Z}_j \le lmax \tag{22}$$

$$\sum_j Y_{jk} = 1 \qquad \forall k \tag{23}$$

$$\sum_k ldw_k Y_{jk} \le lcap_j \overline{Z}_j \qquad \forall j \tag{24}$$

$$\sum_k ldw_k Y_{jk} = \left(\sum_i udw_j X_{ij} \right) \overline{Z}_j \qquad \forall j \tag{25}$$

$$lrd_{jk} Y_{jk} \le ltr_j \qquad \forall j, k \tag{26}$$

$$Z_i, X_{ij}, \overline{Z}_j, Y_{jk} \in \{0,1\}, \qquad \forall i, j, k \tag{27}$$

4. Proposed solution approaches

As shown in Table 1, three approaches were presented to solve the BL-EW-LAP. An optimal solution method (FE-EA) was proposed based on the explicit complete enumeration. In this algorithm, both levels of the problem are solved by an exact algorithm. Two hybrid evolutionary approaches were introduced based on the genetic algorithm. In this paper, these approaches were designed as a combination of genetic algorithms in the ULM and an exact method in the LLM. The two hybrid genetic algorithms in this paper were named HG-ES-1 and HG-ES-2. According to Table 1, the facilities allocation approaches at the ULM and LLM model are the exact method for FE-EA and HG-ES-1 algorithm and the heuristic method for the HG-ES-2 algorithm, respectively. In three proposed solution approaches, GAMS software was used to implement exact algorithms. The CPLEX solver and the branch-bound algorithm were used In GAMS software. The main difference between the two genetic algorithms is the initial solution generation and the allocation method. Table 2 presents the summary tasks of solution generation regarding the two hybrid genetic algorithms.

Table 1. Proposed solution algorithms for BL-EW-LAP

Name of Algorithm	Type of approach	ULM solution	LLM solution	Facility allocation method
FE-EA	explicit complete enumeration	exact algorithm	exact algorithm	exact algorithm
HG-ES-1	nested evolutionary	genetic algorithm	exact algorithm	exact algorithm
HG-ES-2	nested evolutionary	genetic algorithm	exact algorithm	heuristic algorithm

Table 2. Solution generation of HG-ES-1 and HG-ES-2 algorithms

Name of approach	Steps	Decision variable	Decision type	Level	Methods
HG-ES-1	Step1	$Z_i, \overline{Z}_j$	Location	National and Regional	Randomly and sub-routine
	Step2	X_{ij}	Allocation	National	Nearest distance
	Step3	Y_{jk}	Allocation	Regional	Exact algorithm
	Step4	X_{ij}	Allocation	National	Exact algorithm
HG-ES-2	Step1	$Z_i, \overline{Z}_j$	Location	National and Regional	Randomly
	Step2	Y_{jk}	Allocation	Regional	Heuristic algorithm
	Step3	X_{ij}	Allocation	National	Heuristic algorithm
	Step4	Y_{jk}	Allocation	Regional	Exact algorithm
	Step5	X_{ij}	Allocation	National	Heuristic algorithm

4.1. Full Enumeration and Exact Algorithm (FE-EA)

According to Fig. 2, in the first step, the FE-EA algorithm identifies all the allocation modes of national warehouses to regional warehouses (X_{ij}). In the second and third steps, (X_{ij}) are sent to the LLM and the optimal solutions of the LLM $(\bar{Z}_j{}^*, Y_{jk}{}^*)$ are calculated by the exact algorithm. In steps 4 and 5, $(Y_{jk}{}^*)$ are sent to the ULM, and the optimal solutions of the ULM $(Z_i{}^*, X_{ij}{}^*)$ are calculated for all cases through the exact algorithm. Ultimately, the best solution of the ULM is considered to be the optimal solution for the bi-level problem.

1: **Construct** all allocation modes of the ULM X_{ij}
2: **Let** X_{ij} be as a parameters into the LLM
3: **Calculate** $\overline{Z}_j^*, Y_{jk}^*$ by an exact algorithm
4: **Let** Y_{jk}^* be as a parameters into the ULM
5: **Calculate** $Z_i^*, X_{ij}^*, OF_{ULM}^*$ for each Y_{jk}^* by an exact algorithm
6: **Find** Min{ OF_{ULM}^* }

Figure 2. Pseudo-code of FE-EA algorithm

4.2. HG-ES-1

Generally, this algorithm consists of two phases, namely the initial solution generation and the evaluation phase. The pseudo-code of the HG-ES-1 algorithm is presented in Fig. 3.

4.2.1. Solution representation and initialization

In the initial solution generation, an initial population of chromosomes(POP_0) are produced. As shown in Fig. 4, the chromosome structure is composed of two parts, the first one assigned to locating the national warehouse and the second dedicated to locating the regional warehouse. The length of the array for both parts is equal to the maximum number of warehouses which can be placed at the national and regional levels $(umax, lmax)$. A solution is made by unique integers between 1 and m for part1 and unique integers between 1 and n for part2.

According to Fig. 5, the initial population (POP_0) is generated through the sub-routine procedure. In the first step, $POP_0(1)$ is achieved from the optimal solution of the RXBL model, which was obtained from solving the relaxed model (RXBL) by GAMS. The remaining solutions of the initial population are randomly generated. In the next step, all chromosomes of (POP_0) are examined by the repair procedure. The steps of the repair procedure are as follows; first, the demand cities uncovered by any regional warehouses are searched and stored as set (NC). Afterwards, the regional warehouses able to serve NC are listed and stored as set (RE). In the final step, the RE index is inserted in the chromosome. These steps are repeated until all the demand cities are covered by at least one regional warehouse candidate. Following the generation of chromosomes, the national warehouses are allocated to regional warehouses based on the nearest distance, and X_{ij} is calculated. Next, (X_{ij}) is sent to the LLM as a parameter. Then $({Y_{jk}}^*)$is calculated by the exact algorithm and sent to the ULM, and $({X_{ij}}^*)$ is once again calculated via the exact algorithm.

4.2.2. Evaluation phase of genetic algorithm

In the evaluation phase of the algorithm, crossover and mutation operators were applied to the existing solutions. A two-point crossover operator was designed according to Fig. 6. In the crossover operation, one of the two parts of the chromosome is randomly selected with the same chance (probability of selection = 0.5), and the two-point crossover operator is applied to that part. If each chromosome part contains a duplicate index, indicating an infeasible solution, it will be modified as observed in Fig.6. Similar to the crossover operator, the mutation operation randomly selects one part of the chromosome with the same chance. Afterwards, one cell is randomly selected and replaced by an index that does not exist in the parent. In this way, the mutation operator produces an offspring from one parent chromosome (Fig.7). After performing operators, all offspring are once again checked by the repair steps of the sub-routine procedure.

4.3. HG-ES-2

The overall structure of this algorithm is similar to the HG-ES-1 whereas the solution generation and allocation method are different. The solution generation steps of the HG-ES-2 algorithm are illustrated in Fig. 8, and other steps of HG-ES-2 algorithm are the same as HG-ES-1. In the initial solution phase of the HG-ES-2 algorithm, POP_0 is generated randomly. In this algorithm, the allocation of national warehouses to regional warehouses and regional warehouses to demand cities are performed through the five heuristic allocation algorithms separately, presented in (Saranwong & Likasiri, 2016). In all the heuristic allocation algorithms, the assignment of facilities is based on the nearest distances. The difference between heuristic methods is the sorting criteria of nodes and their priority determination in the assignment. The sorting criteria of each algorithm are in accordance with Table 3. In HG-ES-2, demand cities are allocated to regional warehouses and (Y_{jk}) is calculated by one of the heuristic algorithms. In the next step, the weight associated to each regional warehouse (udw_j) is calculated via equation $(\sum_k ldwY_{jk})$. Once again, through one of the five allocation algorithms, the regional warehouses are allocated to the national warehouse and (X_{ij}) is calculated. After that, (X_{ij}) is sent to the LLM as the parameter. In the next step, the LLM is solved by use of the exact algorithm. Afterwards, the optimal solution of the LLM (Y_{jk}^*) is sent for the ULM. Regarding(Y_{jk}^*), (X_{ij}) is modified and replaced with the previous solution.

1: Input
2: **Let** p_c be the percentage of Crossovers population
3: **Let** p_m be the percentage of Mutation population
4: **Let** $npop$ be the size of population
5: **Let** spr be the percentage of Selection pressure rate
6: **Let** CON be a counter that initially set to 0
7: **Let** $MaxCON$ the maximum number of trials that the algorithm is not improved
8: Selection method of parent /*Roulette wheel selection*/
Initial Solution generation
9: **Call sub-routine** for generate an initial population of chromosomes (pop_0) /*integer string*/
10: **For** all chromosomes
11: **Calculate** X_{ij} based nearest distance between warehouses
12: **Let** X_{ij} be as a parameters into the LLM
13: **Calculate** Y_{jk}^* by an exact method
14: **Let** Y_{jk}^* be as a parameters into the ULM
15: **Calculate** X_{ij}^* by an exact method
16: **Calculate** OF_{ULM} of each chromosome
Evaluation phase:
17: While $CON < MaxCON$
18: $n_c = npop * p_c$ / * number of chromosomes that will be generated by crossover*/
19: $i = 1$;
20: While $i \le n_c / 2$
21: **Select** two chromosomes as parents based on Roulette wheel selection
22: **Generate** two offspring chromosomes by crossover
23: $n_m = npop * p_m$; / * number of chromosome that will be mutated*/
24: $j = 1$;
25: **While** $j \le n_m$
26: **Select** a random chromosome
27: **Mutate** chromosome
28: **For** each new chromosome
29: **Use repair steps of sub-routine** and update chromosome
30: **Repeat** 11 to 16 steps
31: **Calculate** OF_{ULM} of each new chromosome
32: Merge and sort population
Update stop condition
33: **If** the best solution not improve
34: $CON = CON + 1$
35: Else
36: $CON = 0$
37: End
38: **Output** best solution

Figure 3. Pseudo-code of HG-ES-1

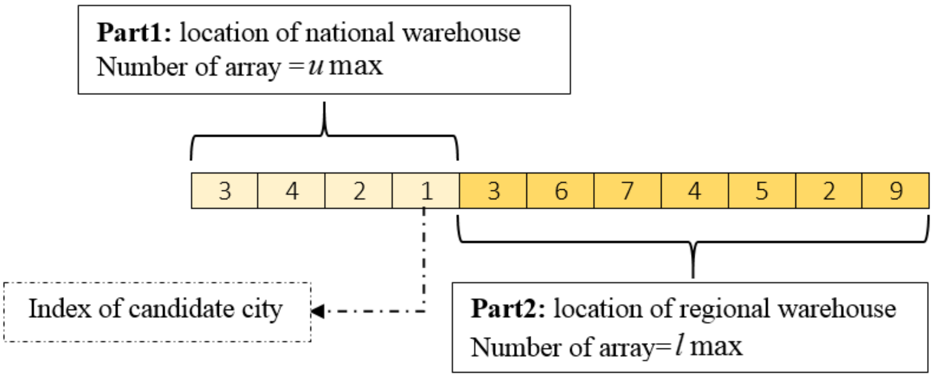


Figure 4. Solution representation

Generate pop_0 (1) from the RXM
Generate $pop_0(2{:}npop)$ randomly
For all chromosome
    Repair:
        Repeat
        Find $NC = \{k, \boldsymbol{lrd_{jk}} \leq \boldsymbol{ltr_j}\}$
        Find $RE = \{j, \boldsymbol{lrd_{jk}} \geq \boldsymbol{ltr_j} \quad \forall NC\}$
        Replace RE into cells of chromosome randomly
        Until NC= ∅
End for

Figure 5. Pseudo-code of the sub-routine procedure (HG-ES-1)

Table 3. Heuristic algorithms for allocation in HG-ES-2

Algorithm Code	Name of heuristic algorithm	Allocation criterion for ULM	Allocation criterion for LLM
RA	Random Algorithm	Random assignment of regional warehouses	Random assignment of demand points
DE	Demand Algorithm	Regional warehouses with higher weights (w_j) have higher priorities	Demand points with higher weights $(\overline{w}_k)$ have higher priorities
AD	Average Distance Algorithm	The regional warehouse with a higher average distance to the national warehouse has a higher priority	The demand point with a higher average distance to the regional warehouse has a higher priority
DG	Distance Gap Algorithm	The regional warehouse with a larger gap between the closest and second closest distances to the national warehouse has a higher priority	The demand point with a larger gap between the closest and second closest distances to the regional warehouse has a higher priority
STD	Standard Deviation Algorithm	The regional warehouse with a higher standard deviation (SD) of distances between a regional warehouse and each national warehouse has a higher priority	The demand point with a higher standard deviation (SD) of distances between a demand point and each regional warehouse has a higher priority

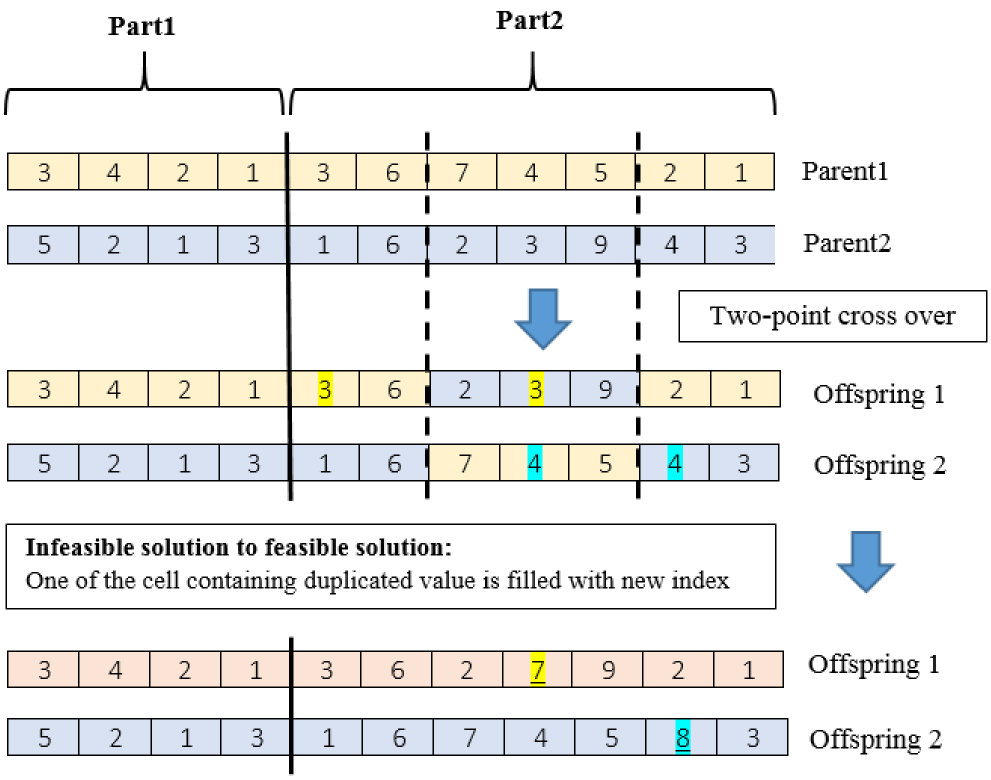


Figure 6. Crossover operator

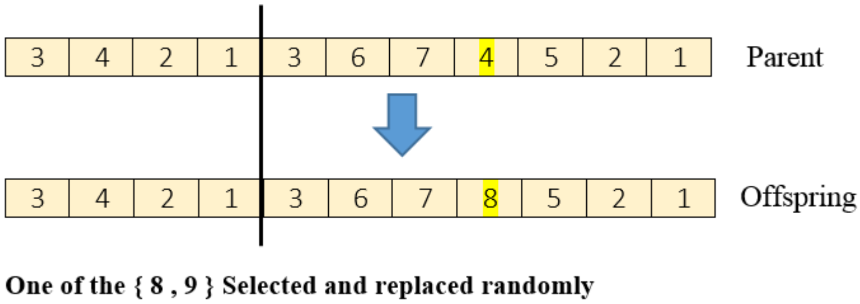


Figure 7. Mutation operator

Initial solution generation

Generate an initial population of chromosomes (pop_0) /*integer string*/

For all chromosomes

  Calculate Y_{jk} by the allocation heuristic algorithm

  Calculate $udw_j = \sum_k ldw_k Y_{jk}$

  Calculate X_{ij} by the allocation heuristic algorithm

  Let X_{ij} be as a parameters into the LLM

  Calculate, Y^*_{jk} by an exact algorithm

  Let Y^*_{jk} be as a parameters into the ULM

  Update w_j and calculate X_{ij} by the allocation heuristic algorithm based on chromosome

Figure 8. Initial solution generation of the HG-ES-2 algorithm

Note: For the sake of simplicity, we renamed each HG-ES-2 in the abbreviated form of heuristic algorithm names. For instance, HG-ES-2 (STD) means that the allocation of the upper and lower levels in HG-ES-2 is performed by the STD algorithm.

5. Computational results

5.1. Parameters setting for the BL-EW-LAP model

Parameters of the BL-EW-LAP model were generated based on the data in Iran, one of the high-risk countries regarding various man-made and natural disasters in the Middle East. Due to the specific geographical location of Iran, earthquake is more burdensome than other disasters in the country. Over the past forty years, there have been major earthquakes such as Tabas (1978), Manjil-Rudbar (1990), Bam (2003), Azerbaijan Sharghi (2012), and Kermansh (2017). These earthquakes killed more than 100000 people and imposed a lot of costs on the country. The steps of parameter setting based on earthquake disaster are described in the following.

5.1.1. Weight associated with each demand city (ldw_k)

In this paper, cities of Iran were considered to be warehouse candidates and demand cities. According to equation (28), ldw_k was calculated based on earthquake risk (ER) and the population of the city (PP). The population statistic was extracted from the Statistical Center of Iran website according to the Population and Housing Census (2016). Iranian code of practice for seismic resistant design of building, standard No.2800, was employed to calculate the earthquake risk of each city.

$$ldw_k = ER_k * PP_k \tag{28}$$

5.1.2. Distance between facilities (urd_{ij}, lrd_{jk})

To calculate the matrix of the distances between nodes (urd_{ij}, lrd_{jk}), we considered road transportation and used Google Maps Platform and Distance Matrix API service. A code was written using Python programming language to determine the distance matrix of nodes (urd_{ij}, lrd_{jk}).

5.1.3. Distance threshold (utr_i, ltr_j)

The population in Iran has a heterogeneous distribution. Population density is high in certain regions of the country such as the center and north. On the contrary, in the southeastern regions of Iran, the population density is low. Therefore, the threshold distance of each regional warehouse was calculated based on the average distance between that candidate city and all demand cities (equation 29). Similarly, the threshold distance of each national warehouse was calculated by the average distance between each candidate city and all the cities of the regional warehouse candidates (equation 30).

$$utr_i = \sum_{j=1}^{n} urd_{ij} / n \tag{29}$$

$$ltr_j = \sum_{k=1}^{k} lrd_{jk} / k \tag{30}$$

5.1.4. Warehouses capacity ($ucap, lcap_j$)

The capacity of the warehouses ($ucap, lcap_j$) was determined based on the proportion of the covered demand by candidate cities. According to equations 31 and 32, the covered demand (cov_i, cov_j) was calculated from the sum of the weight associated to each demand city where the distance between the demand cities and the candidate city was less than the distance threshold. Finally, the capacities of national and regional warehouse ($ucap_i, lcap_j$) were calculated with multiplication α in the covered demand (equation 33 and 34). The value of α depends on the size of the candidate city and its management infrastructure.

$$cov_i = \sum_{j\in s} udw_j \quad S = \{j, urd_{ij} \leq utr_i\} \tag{31}$$

$$cov_j = \sum_{k\in T} ldw_k \quad T = \{k, lrd_{ij} \leq ltr_j\} \tag{32}$$

$$ucap_i = \alpha * cov_i \tag{33}$$

$$lcap_j = \alpha * cov_j \tag{34}$$

5.2. Parameter tuning of genetic algorithms

Parameter tuning is one of the main determinants of the efficiency of evolutionary algorithms such as genetic algorithm. The inappropriate selection of an efficient algorithm parameter leads to weak performance (Zandieh et al., 2009). In this research, the Taguchi method was employed to calibrate the developed HG-ES algorithm parameters. This method has been extensively applied in tuning evolutionary algorithm parameters in (Akbari Kaasgari et al., 2017; M Hajiaghaei-Keshteli & Aminnayeri, 2014; M. Hajiaghaei-Keshteli et al., 2010; Mokhtarinejad et al., 2015). In this method, factors affecting the performance of the genetic algorithm were firstly specified based on the literature review. Population size, crossover rate, mutation rate, and selection pressure were selected as the effective factors for calibration. The factors and their levels are shown in Table 4. In this method, the signal-to-noise(S/N) ratio was maximized to find the optimal levels of factors according to equation 35; y_i and n define the solution value and the number of replications, respectively.

$$S / N = -10 \log \left(\sum_{i=1}^{n} y_i^2 / n \right) \tag{35}$$

Table 4. Levels of effective factors for genetic algorithm

factors	level	Values
Size of population	3	50,75,100
Crossover rate	3	0.7,0.8,0.9
Mutation rate	3	0.1,0.2,0.3
Selection pressure	3	1,2,3

Nine experiments (run four times) were designed in this research, and the average values were calculated. The response variable considered in this research was the robust parameter design (RPD), the maximization of which was the point of purpose, as defined in equation 36. OF_{Max} and OF_i were the best solutions obtained for a specific problem and the objective function for the ith example. The obtained values for S/N and average were inserted in Figs 9 to 14, according to which, the best levels of factors for the developed algorithms are mentioned in Table 5. The Roulette Wheel was used for the selection method. The stopping condition of the algorithm was set to 20 iterations, which not improve the best solution.

$$RPD = (OF_{Max} - OF_i) / OF_{Max} \tag{36}$$

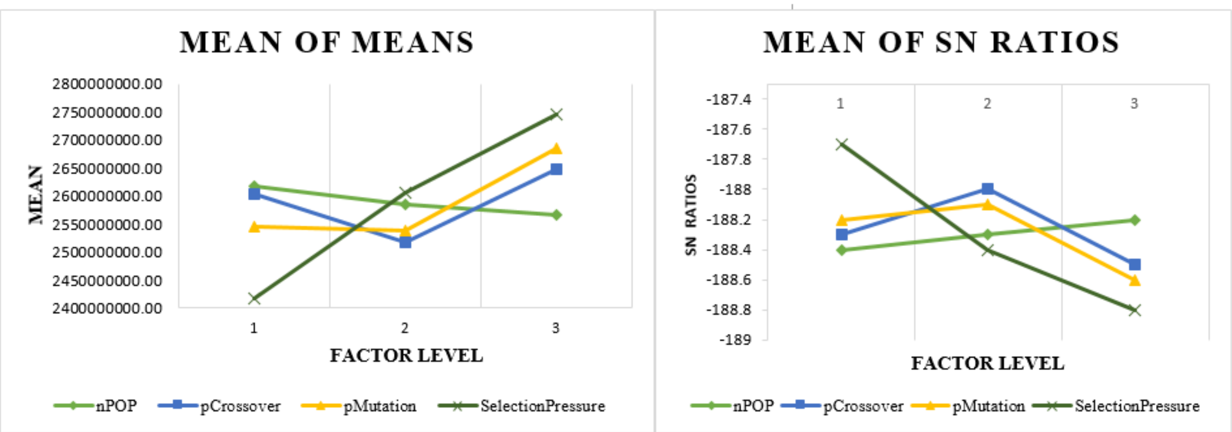


Figure 9. The average of means and S/N ratios for HG-ES-1 algorithm

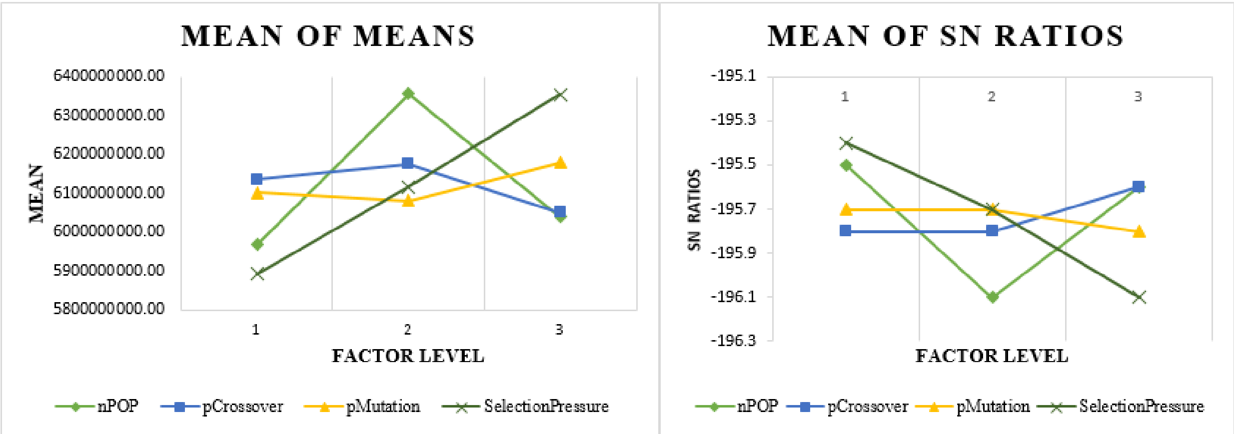


Figure 10. The average of means and S/N ratios for HG-ES-2(STD) algorithm

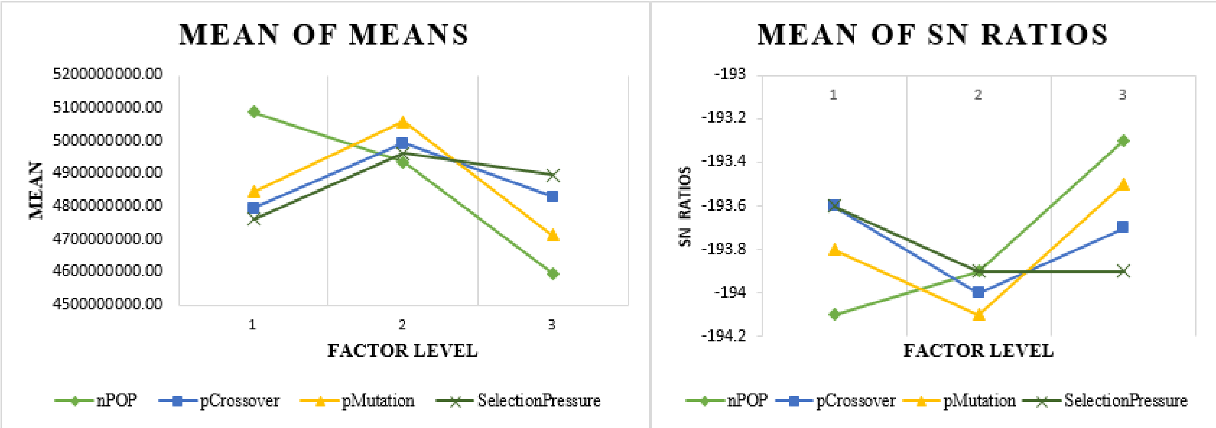


Figure 11. The average of means and S/N ratios for HG-ES-2(DG) algorithm

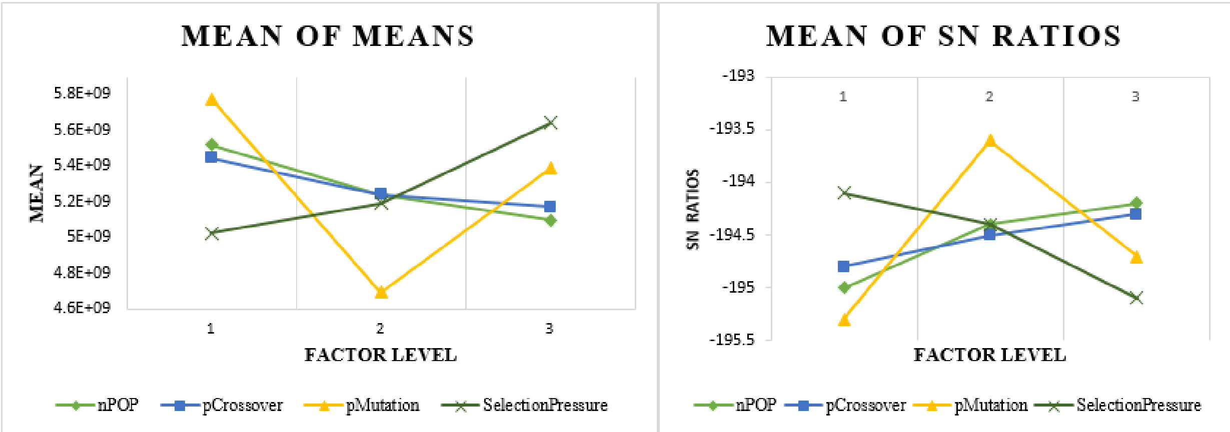


Figure 12. The average of means and S/N ratios for HG-ES-2(AD) algorithm

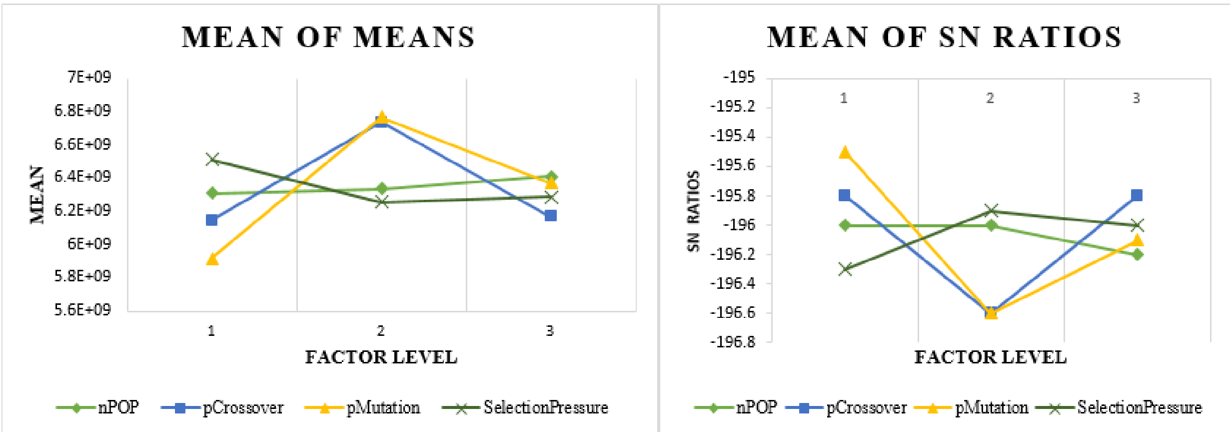


Figure 13. The average of means and S/N ratios for HG-ES-2(DE) algorithm

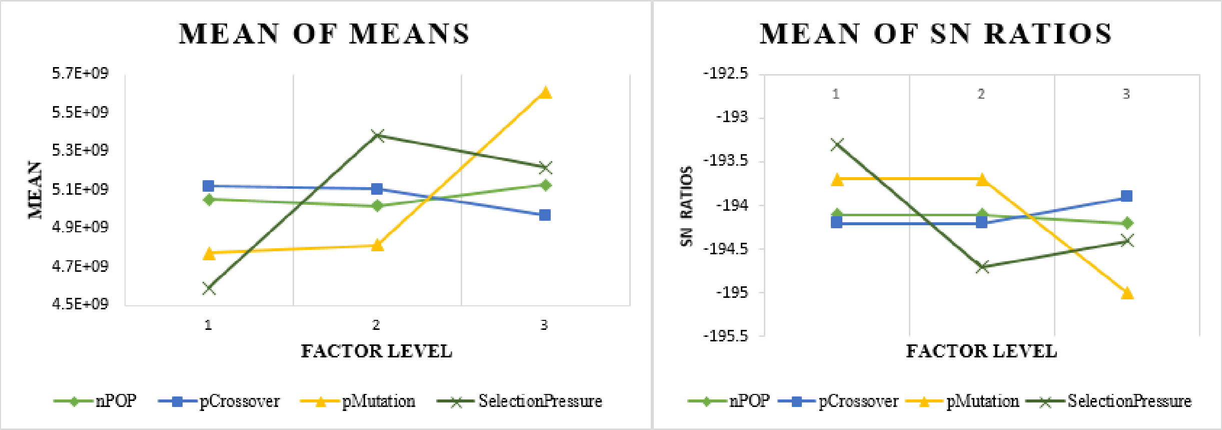


Figure 14. The average of means and S/N ratios for HG-ES-2(RA) algorithm

Table 5. The values of parameters selected for all types of the HG-ES algorithms

Algorithm	Pop size	Crossover rate	Mutation rate	Selection pressure
HG-ES-1	100	0.8	0.2	1
HG-ES-2(STD)	50	0.9	0.2	1
HG-ES-2(DG)	100	0.7	0.3	1
HG-ES-2(AD)	100	0.9	0.2	1
HG-ES-2(DE)	50	0.7	0.1	2
HG-ES-2(RA)	75	0.9	0.1	1

5.3. Evaluation of the proposed algorithms

To compare the proposed algorithms, three instances were prepared in small, medium, and large sizes. Table 6 shows the specifications of the instances. In this paper, nine criteria were reported as commotional results, six of which were considered in evaluating the algorithm performance (See Table 7). The lower values of the nine criteria exhibited higher algorithm performance. Legillon et al. (2012) introduced direct rationality (DR) to evaluate the bi-level solution, which counted the number of times the algorithm enhanced its response. Two criteria (INF and AIT) were calculated based on equations 37-41, respectively. According to equation 26, the deviation of lower-bound (DLB) is equal to the subtraction proportion of the best ULM solution from the lower-bound of bi-level model (LBBL). In the present paper, the lower-bound of the bi-level model (LBBL) was obtained from solving the relaxed model (RXBL) by an exact algorithm.

$$DLB = (BSU - LBBL) / LBBL \qquad (37)$$

$$INF_r = N\inf_r / Nso_r \qquad (38)$$

$$AINF = \sum_{r=1}^{r} INF_r / r \qquad (39)$$

$$IT_h = CPUTime_r / Nit_r \qquad (40)$$

$$AIT = \sum_{r=1}^{r} IT_h / r \qquad (41)$$

$r = number\ of\ algorihm\ runs$
$INF_r = proportion\ of\ infeseable\ solution\ for\ rth\ algorithm\ run$
$Nso_r = total\ number\ of\ generation\ solutions\ \ for\ rth\ algorithm\ run$
$Nit_r = number\ of\ iteration\ for\ rth\ algorithm\ run$
$IT_r = iteration\ time\ for\ rth\ algorithm\ run$

Table 6. Specification of the designed problems

Problem number	**Problem size**	**Number of national warehouse candidate (i)**	**Number of regional warehouse candidate (j)**	**Number of demand cities(k)**	$umax$	$lmax$
1	Small	5	6	5	2	3
2	Medium	9	19	38	3	6
3	Large	9	31	117	5	9

Table 7. Criteria for the proposed algorithm evaluation

Name of criteria	**Abbreviation**	**Apply to evaluate**	**Units of values**
Average objective function value of ULM	AOFU	√	weighted distances (W*D)
Standard deviation objective function value of ULM	SDOFU	√	weighted distances (W*D)
Best solution of ULM	BSU	√	weighted distances (W*D)
Best solution of LLM	BSL	×	weighted distances (W*D)
Average CPU Time	CPUT	×	Second (sec)
Average direct rationality	DR	√	Index∈[0-1]
Deviation from the lower-bound	DLB	√	Index∈[0-1]
Average iteration time	AIT	×	Second (sec)
Average infeasibility proportion	AINF	√	Index∈[0-1]

The small-size instance was solved in two modes, namely a and b .The BL-EW-LAP model was considered in Mode (a), and in Mode (b), the distance threshold constraint was removed from the BL-EW-LAP model. According to Table 8, four genetic algorithms calculated the optimal solution (best solution of FE-EA algorithm) for the small-size problem in both modes. As shown in Table 9, HG-ES-1 and HG-ES-2(STD) obtained the best solution of ULM in the medium-size instance. According to Fig 15-17 and Table 10, HG-ES-1 outperformed all the sub-groups of HG-ES-2 with AOFU, SDOFU, and BSU for large variables model. The HG-ES-1 algorithm obtained the best results for the DR and AINF (see Table 11). As illustrated in Fig. 19, HG-ES-1 achieved better DLB values regarding all problem sizes. Ultimately, HG-ES-1 was superior to the other five algorithms in terms of the six evaluation criteria; therefore, this algorithm was selected as the appropriate algorithm for this research. The BL-EW-LAP was a long-term problem, and CPU time and ATI were not used to choose the appropriate algorithm. Fig. 18 and Table 19 show that the HG-ES-1 algorithm ranked fourth among the six algorithms in terms of CPU Time and had the highest value for ATI index.

Table 8. The results of BL-EW-LAP (small-size)

Name of algorithm	**Mode(a)**		**Mode(b)**	
	BSU	**BSL**	**BSU**	**BSL**
FE-EA	1394908294.2	1521270482.2	912589320.2	2913096362.3
HG-ES-1	1394908294.2	1521270482.2	912589320.2	2913096362.3
HG-ES-2(STD)	1394908294.2	1521270482.2	912589320.2	2913096362.3
HG-ES-2(DG)	1396118600.9	1589158738.1	946221123	2709942155.5
HG-ES-2(AD)	1396118600.9	1589158738.1	946221123	2709942155.5
HG-ES-2(DE)	1394908294.2	1521270482.2	912589320.2	2913096362.3
HG-ES-2(RA)	1394908294.2	1521270482.2	912589320.2	2913096362.3

Table 9. The results of BL-EW-LAP (medium-size)

Algorithm code	AOFU	SDOFU	BSU	BSL	CPUT (sec)
HG-ES-1	1719427978	0	1719427978	4843577231	908
HG-ES-2(STD)	1908065773	50402335.98	1887489105	3964128160	986
HG-ES-2(DG)	2264293525	255366766.4	2065258980	4076310089	782
HG-ES-2(AD)	2087432395	320994009.1	1719427978	1719427978	999
HG-ES-2(DE)	2167416505	246822805.6	2010949110	4868108559	876
HG-ES-2(RA)	2432397178	413134182.3	1887489105	1887489105	901

Table 10. The results of BL-EW-LAP (large-size)

Algorithm code	AOFU	SDOFU	UBS	LBS	CPUT (sec)
HG-ES-1	2901798931	743578068.6	1905346704	14129090278	2432
HG-ES-2(STD)	6179698851	858247863.7	5144236733	15985743371	2857
HG-ES-2(DG)	5052407285	928165770.9	4232517664	15880775166	2036
HG-ES-2(AD)	5606144035	2180603340	2843806906	11877905344	1682
HG-ES-2(DE)	6000184435	1280348619	4834429230	13034504401	2653
HG-ES-2(RA)	5169373506	1656754817	3659385473	14568610874	1451

Table 11. Result of evaluation criteria for large-size instance

	Medium-size			Large-size		
Algorithm code	AINF	DR	AIT	AINF	DR	AIT
HG-ES-1	0.042	0.040	43.4	0.19	0.040	80.71
HG-ES-2(STD)	0.16	0.129	20.5	0.28	0.077	52.65
HG-ES-2(DG)	0.21	0.143	18.9	0.566	0.132	38.26
HG-ES-2(AD)	0.15	0.132	21.4	0.342	0.175	34.52
HG-ES-2(DE)	0.18	0.103	21.0	0.2095	0.132	36.22
HG-ES-2(RA)	0.26	0.196	21.1	0.818	0.167	17.15

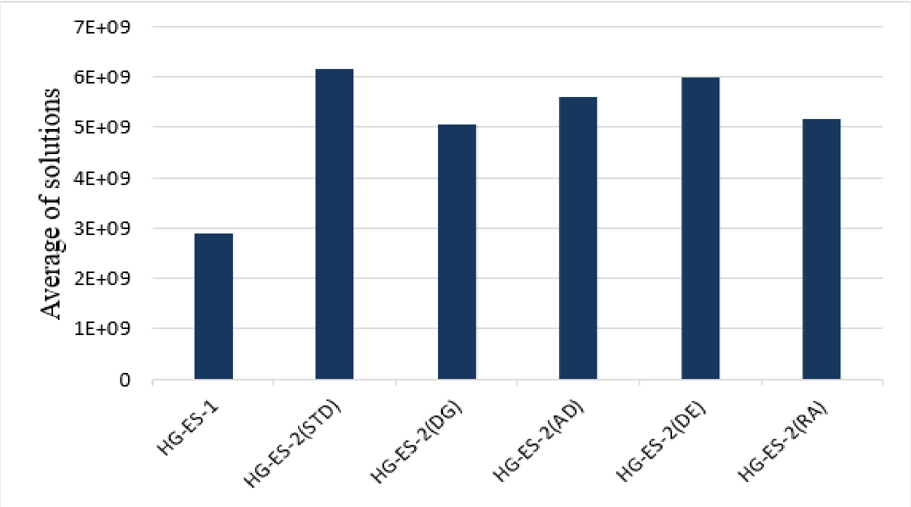


Figure 15. Result of the large-size instance (AOFU)

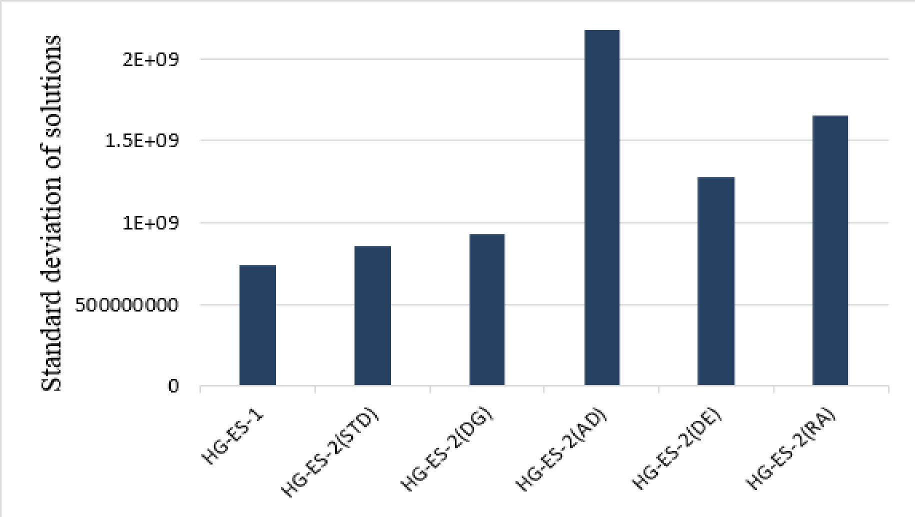


Figure 16. Result of the large-size instance (SDOFU)

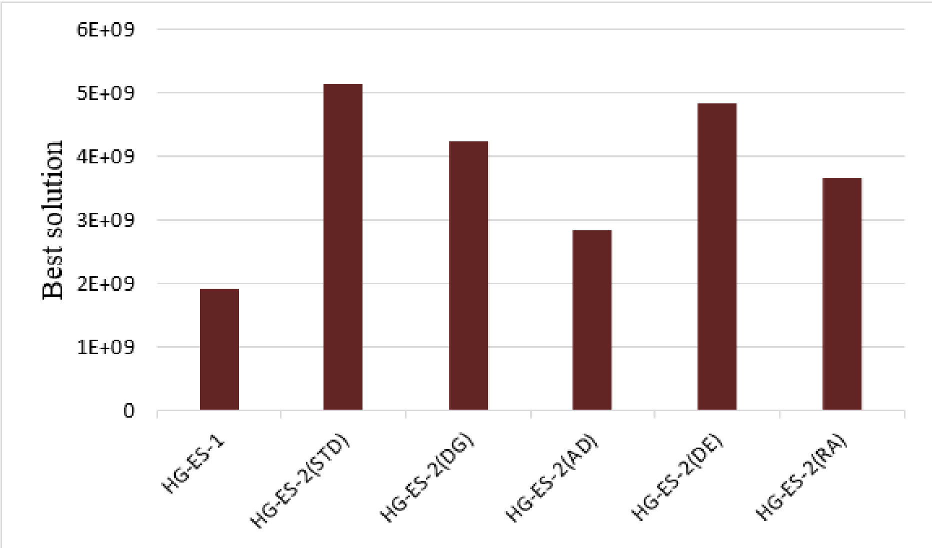


Figure 17. Result of the large-size instance (BSU)

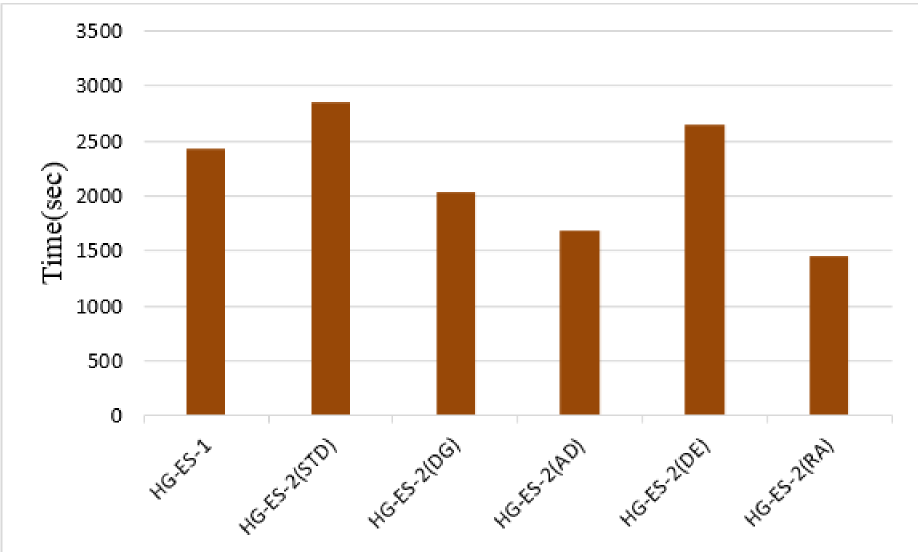


Figure 18. Result of the large-size instance (CPUT)

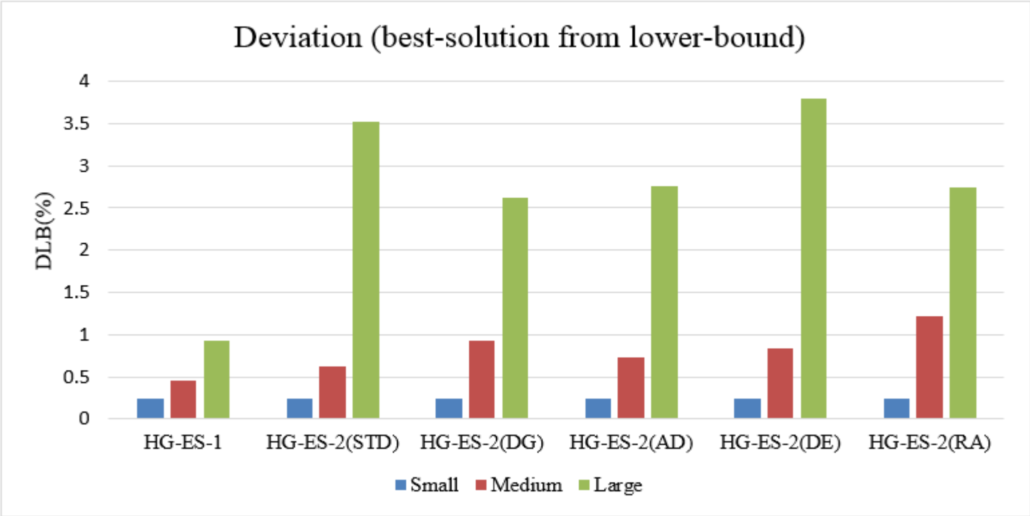


Figure 19. DLB of proposed algorithms for all size instance

5.4. Analysis

The following examines the effect of changing parameters on the objective function of the bi-level model. The maximal number of national and regional warehouses placed at ($umax,$ lmax) and the capacity of warehouses $(ucap_i, lcap_j)$ are the key parameters affecting the objective functions. As shown in Table 12, changing the maximal number of regional warehouses ($lmax$) did not impact the increase in the objective function values of both levels; however, with the increase in the maximal number of regional warehouses ($umax$), the value of the ULM objective function was reduced. This underscores the need for the modeler to accurately determine this parameter in the model. Moreover, according to Table 13, it was found that increasing the capacities of each national and regional warehouse could separately reduce the objective function of the ULM. Fig. 20, shows the behavior of the objective functions and the effect of the leader's response on the follower for different solutions. The leader is monotonically decreasing, while the follower has some variations.

Table 12. Impact of change $umax$, $lmax$ on objective functions

Instance	Objective function (ULM)	Objective function (LLM)	$Umax$	$Lmax$
1	3771031849	5323428965	2	4
2	3771031849	5323428965	2	5
3	3771031849	5323428965	2	6
4	3771031849	5323428965	2	7
5	1669882335	6397497550	3	4
6	1756957065	5337344728	3	5
7	1719427978	4843577231	3	6
8	1719427978	4843577231	3	7
9	1065013326	5189249823	4	4
10	1065013326	5189249823	4	5
11	1065013326	5189249823	4	6

Table 13. Impact of change c_i, α on objective functions

Instance	Capacity of national warehouse (c_i)	α (regional warehouse)	Objective function (ULM)	Objective function (LLM)
1	5500000	0.4	2709200368	4847162626
2	5500000	0.5	1719427978	4843577231
3	5500000	0.6	2262382669	5132827589
4	6000000	0.4	2203986298	5079818939
5	6000000	0.5	1719427978	4843577231
6	6000000	0.6	1349969708	5149054551
7	6500000	0.4	2203986298	5079818939
8	6500000	0.5	1719427978	4843577231
9	6500000	0.6	1683825138	4826802816
10	7000000	0.4	2170159579	4386446911
11	7000000	0.5	1719427978	4843577231
12	7000000	0.6	1349969708	5149054551

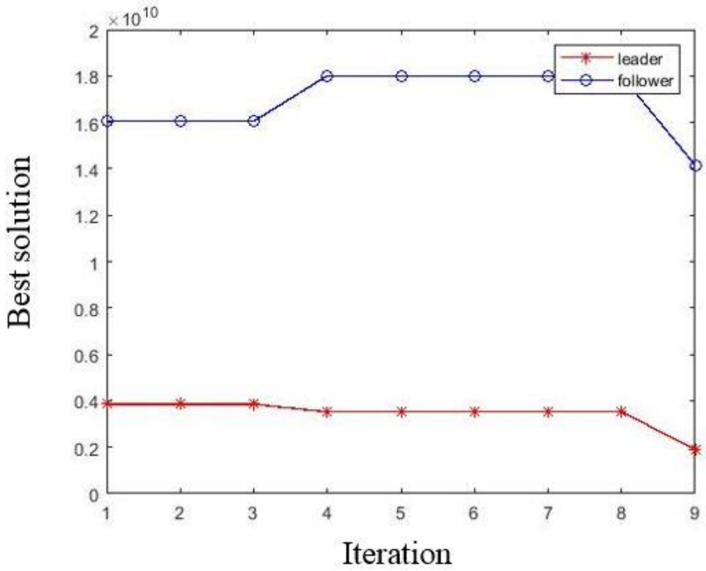


Figure 20. The changes in objective functions by HG-ES-1

6. Conclusion

The emergency warehouse location-allocation problem can be formulated by the bi-level programming when the decision-making structure is decentralized and hierarchical. In this paper, we presented a new binary bi-level programming model for this problem. The central government and the regional manager were respectively considered as the leader and the follower. The optimization model presented in this research is conducive to reducing the managerial conflicts at the national and the regional levels and provide consensus in decision-making for the central government and regional managers. In other words, the proposed framework can reduce management crises in disaster management.

The bi-level optimization models in a simple and linear form are complex and difficult to solve. Particularly, with the addition of integer and binary variables, this complexity increases (Sinha et al., 2017). Three approaches were developed to solve the bi-level emergency warehouse location-allocation problem (BL-EW-LAP). The first approach, FE-EA, was designed based on the explicit complete enumeration. According to the computational results, this algorithm is suitable for optimally solving small-size models. The size of real-world problems is usually large, and it is difficult to design optimal global algorithms. Evolutionary algorithms are among the approaches to solving these problems. The general framework of the second (HG-ES-1) and third (HG-ES-2) approach is based on Matheuristic approach.

The HG-ES-1 algorithm exhibited better results in this paper. With the addition of distance threshold constraints, the HG-ES-2 lost its effectiveness in solving the model. The main reasons behind the superiority of HG-ES-1 over HG-ES-2 are related to the allocation method and the solution generation procedure of the two algorithms. The HG-ES-1 allocated facilities at both levels through an exact method, but the allocation in the HG-ES-2 was performed by the heuristic algorithms. The use of the exact method for allocation enhanced the exploitation of the best observed solutions (intensification), a fact further highlighted by the low Direct Rationality (DR) of HG-ES-1. The solution generation of HG-ES-1 was performed by Sub-routine procedure, and the nearest distance algorithm with allocation in the upper-level (X_{ij}) was estimated without having to specify the allocation in the lower-level (Y_{jk}). The solution generation of the HG-ES-2 is a bottom-up approach (lower level model to upper level model), meaning that to calculate (X_{ij}), (Y_{jk}) must be calculated first. In certain cases, the generation of good solutions for the follower may exclude the generation of good solutions for the leader. This mechanism reduced the exploration of the search space (diversification), markedly impacting its performance. The solution repair through Sub-routine procedure of HG-ES-1 was avoided to generate infeasible solutions, leading to a low value for average infeasibility proportion (AINF). To improve the proposed algorithm, innovative design of chromosomes is recommended. For discrete bi-level optimization programming, future research can also consider other evolutionary approaches such as co-evolutionary approach with various evolutionary algorithms such as Ant colonies and Particle Swarm Optimization (PSO), and Artificial immune systems (AIS).

Refrences

Akbari Kaasgari, M., Imani, D. M., and Mahmoodjanloo, M. (2017). Optimizing a vendor managed inventory (VMI) supply chain for perishable products by considering discount: Two calibrated meta-heuristic algorithms. *Computers & Industrial Engineering*, Vol. 103, pp. 227-241.

Aksen, D., and Aras, N. (2013). A matheuristic for leader-follower games involving facility location-protection-interdiction decisions *Metaheuristics for Bi-level Optimization* (pp. 115-151): Springer.

Balcik, B., and Beamon, B. M. (2008). Facility location in humanitarian relief. *International Journal of Logistics*, Vol. 11(2), pp. 101-121.

Bard, J. (1998). Practical bilevel optimization: applications and algorithms *Series: Nonconvex Optimization and Its Applications* (Vol. 30, pp. 7-8): Springer.

Bard, J. F., and Moore, J. T. (1992). An algorithm for the discrete bilevel programming problem. *Naval Research Logistics*, Vol. 39(3), pp. 419-435.

Boonmee, C., Arimura, M., and Asada, T. (2017). Facility location optimization model for emergency humanitarian logistics. *International Journal of Disaster Risk Reduction,* Vol. 24, pp. 485-498.

Bozorgi-Amiri, A., Jabalameli, M., and Al-e-Hashem, S. M. (2013). A multi-objective robust stochastic programming model for disaster relief logistics under uncertainty. *OR spectrum,* Vol. 35(4), pp. 905-933.

Bozorgi-Amiri, A., and Khorsi, M. (2016). A dynamic multi-objective location–routing model for relief logistic planning under uncertainty on demand, travel time, and cost parameters. *The International Journal of Advanced Manufacturing Technology,* Vol. 85(5-8), pp. 1633-1648.

Bracken, J., and McGill, J. T. (1973). Mathematical programs with optimization problems in the constraints. *Operations Research,* Vol. 21(1), pp. 37-44.

Calvete, H. I., Galé, C., and Iranzo, J. A. (2013). An efficient evolutionary algorithm for the ring star problem. *European Journal of Operational Research*, Vol. 231(1), pp. 22-33.

Camacho-Vallejo, J.-F., Cordero-Franco, Á. E., and González-Ramírez, R. G. (2014). Solving the bilevel facility location problem under preferences by a Stackelberg-evolutionary algorithm. *Mathematical Problems in Engineering, 2014*.

Camacho-Vallejo, J.-F., González-Rodríguez, E., Almaguer, F.-J., and González-Ramírez, R. G. (2015). A bi-level optimization model for aid distribution after the occurrence of a disaster. *Journal of Cleaner Production,* Vol. 105, pp. 134-145.

Caramia, M., and Mari, R. (2016). A decomposition approach to solve a bilevel capacitated facility location problem with equity constraints. *Optimization Letters,* Vol. 10(5), pp. 997-1019.

Chen, Y.-x., Tadikamalla, P. R., Shang, J., and Song, Y. (2020). Supply allocation: bi-level programming and differential evolution algorithm for Natural Disaster Relief. *Cluster Computing*, Vol. 23(1), pp. 203-217.

CRED. (2018). *Natural Disasters 2017*. Retrieved from Brussels:

Dempe, S. (1996). *Discrete bilevel optimization problems*: Inst. für Wirtschaftsinformatik.

Dempe, S. (2002). *Foundations of bilevel programming*: Springer Science & Business Media.

Dempe, S., and Kue, F. M. (2017). Solving discrete linear bilevel optimization problems using the optimal value reformulation. *Journal of Global Optimization,* Vol. 68(2), pp. 255-277.

DeNegre, S. T., and Ralphs, T. K. (2009). A branch-and-cut algorithm for integer bilevel linear programs *Operations research and cyber-infrastructure* (pp. 65-78): Springer.

Fontaine, P., and Minner, S. (2014). Benders decomposition for discrete–continuous linear bilevel problems with application to traffic network design. *Transportation Research Part B: Methodological,* Vol. 70, pp. 163-172.

Gutjahr, W. J., and Dzubur, N. (2016). Bi-objective bilevel optimization of distribution center locations considering user equilibria. *Transportation Research Part E: Logistics and Transportation Review,* Vol. 85, pp. 1-22.

Haeri, A., Hosseini-Motlagh, S.-M., Samani, M. R. G., and Rezaei, M. (2020). A bi-level programming approach for improving relief logistics operations: A real case in Kermanshah earthquake. *Computers & Industrial Engineering,* Vol. 145, pp. 106532.

Hajiaghaei-Keshteli, M., and Aminnayeri, M. (2014). Solving the integrated scheduling of production and rail transportation problem by Keshtel algorithm. *Applied Soft Computing,* Vol. 25, pp. 184-203.

Hajiaghaei-Keshteli, M., Molla-Alizadeh-Zavardehi, S., and Tavakkoli-Moghaddam, R. (2010). Addressing a nonlinear fixed-charge transportation problem using a spanning tree-based genetic algorithm. *Computers & Industrial Engineering,* Vol. 59(2), pp. 259-271.

Hecheng, L., and Yuping, W. (2008). Exponential distribution-based genetic algorithm for solving mixed-integer bilevel programming problems. *Journal of Systems Engineering Electronics,* Vol. 19(6), pp. 1157-1164.

Huang, B., and Liu, N. (2004). Bilevel programming approach to optimizing a logistic distribution network with balancing requirements. *Transportation Research Record,* Vol. 1894(1), pp. 188-197.

Kheirkhah, A., Navidi, H., and Bidgoli, M. M. (2015). Dynamic facility layout problem: a new bilevel formulation and some metaheuristic solution methods. *IEEE Transactions on Engineering Management,* Vol. 62(3), pp. 396-410.

Kheirkhah, A., Navidi, H., and Bidgoli, M. M. (2016). An improved benders decomposition algorithm for an arc interdiction vehicle routing problem. *IEEE Transactions on Engineering Management,* Vol. 63(2), pp. 259-273.

Legillon, F., Liefooghe, A., and Talbi, E.-G. (2012). *Cobra: A cooperative coevolutionary algorithm for bi-level optimization.* Paper presented at the 2012 IEEE Congress on evolutionary computation.

Manopiniwes, W., Nagasawa, K., and Irohara, T. (2014). Humanitarian relief logistics with time restriction: thai flooding case study. *Industrial Engineering & Management Systems,* Vol. 13(4), pp. 398-407.

Miandoabchi, E., and Farahani, R. Z. (2011). Optimizing reserve capacity of urban road networks in a discrete network design problem. *Advances in Engineering Software,* Vol. 42(12), pp. 1041-1050.

Mokhtarinejad, M., Ahmadi, A., Karimi, B., and Rahmati, S. H. A. (2015). A novel learning based approach for a new integrated location-routing and scheduling problem within cross-docking considering direct shipment. *Applied Soft Computing,* Vol. 34, pp. 274-285.

Moore, J. T., and Bard, J. F. (1990). The mixed integer linear bilevel programming problem. *Operations research,* Vol. 38(5), pp. 911-921.

Paul, J. A., and Hariharan, G. (2012). Location-allocation planning of stockpiles for effective disaster mitigation. *Annals of Operations Research,* Vol. 196(1), pp. 469-490.

Paul, J. A., and MacDonald, L. (2016a). Location and capacity allocations decisions to mitigate the impacts of unexpected disasters. *European Journal of Operational Research,* Vol. 251(1), pp. 252-263.

Paul, J. A., and MacDonald, L. (2016b). Optimal location, capacity and timing of stockpiles for improved hurricane preparedness. *International Journal of Production Economics,* Vol. 174, pp. 11-28.

Safaei, A. S., Farsad, S., and Paydar, M. M. (2018). Robust bi-level optimization of relief logistics operations. *Applied Mathematical Modelling,* Vol. 56, pp. 359-380.

Saghehei, E., Memariani, A., and Bozorgi-Amiri, A. (2021). A Bi-level Programming Approach for Pre-positioning Emergency Warehouses. *International Journal of Engineering,* Vol. 34(1), pp. 128-139.

Saranwong, S., and Likasiri, C. (2016). Product distribution via a bi-level programming approach: Algorithms and a case study in municipal waste system. *Expert Systems with Applications,* Vol. 44, pp. 78-91.

Sinha, A., Malo, P., and Deb, K. (2017). A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*.

Sinha, A., Malo, P., Frantsev, A., and Deb, K. (2014). Finding optimal strategies in a multi-period multi-leader–follower Stackelberg game using an evolutionary algorithm. *Computers Operations Research,* Vol. 41, pp. 374-385.

Talbi, E.-G. (2013). A taxonomy of metaheuristics for bi-level optimization *Metaheuristics for bi-level optimization* (pp. 1-39): Springer.

Vicente, L., Savard, G., and Judice, J. (1996). Discrete linear bilevel programming problem. *Journal of optimization theory applications,* Vol. 89(3), pp. 597-614.

Xu, J., Wang, Z., Zhang, M., and Tu, Y. (2016). A new model for a 72-h post-earthquake emergency logistics location-routing problem under a random fuzzy environment. *Transportation Letters,* Vol. 8(5), pp. 270-285.

Yue, D., Gao, J., Zeng, B., and You, F. (2019). A projection-based reformulation and decomposition algorithm for global optimization of a class of mixed integer bilevel linear programs. *Journal of Global Optimization* Vol. 73(1), pp. 27-57.

Zandieh, M., Amiri, M., Vahdani, B., and Soltani, R. (2009). A robust parameter design for multi-response problems. *Journal of computational and applied mathematics,* Vol. 230(2), pp. 463-476.