information

Article

Medical Support Vehicle Location and Deployment at Mass Casualty Incidents

Miguel Medina-Perez , Giovanni Guzmán *, Magdalena Saldana-Perez * and Valeria Karina Legaria-Santiago

Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional (IPN), Unidad Profesional Adolfo López Mateos (UPALM)-Zacatenco, Mexico City 07320, Mexico; jmedinap2022@cic.ipn.mx (M.M.-P.); vlegarias2019@cic.ipn.mx (V.K.L.-S.)
* Correspondence: jguzmanl@ipn.mx (G.G.); amsaldanap@ipn.mx (M.S.-P.);
Tel.: +52-(55)-5729-6000 (ext. 56617) (G.G.)

Abstract: Anticipating and planning for the urgent response to large-scale disasters is critical to increase the probability of survival at these events. These incidents present various challenges that complicate the response, such as unfavorable weather conditions, difficulties in accessing affected areas, and the geographical spread of the victims. Furthermore, local socioeconomic factors, such as inadequate prevention education, limited disaster resources, and insufficient coordination between public and private emergency services, can complicate these situations. In large-scale emergencies, multiple demand points (DPs) are generally observed, which requires efforts to coordinate the strategic allocation of human and material resources in different geographical areas. Therefore, the precise management of these resources based on the specific needs of each area becomes fundamental. To address these complexities, this paper proposes a methodology that models these scenarios as a multi-objective optimization problem, focusing on the location-allocation problem of resources in Mass Casualty Incidents (MCIs). The proposed case study is Mexico City in a earthquake post-disaster scenario, using voluntary geographic information, open government data, and historical data from the 19 September 2017 earthquake. It is assumed that the resources that require optimal location and allocation are ambulances, which focus on medical issues that affect the survival of victims. The designed solution involves the use of a metaheuristic optimization technique, along with a parameter tuning technique, to find configurations that perform at different instances of the problem, i.e., different hypothetical scenarios that can be used as a reference for future possible situations. Finally, the objective is to present the different solutions graphically, accompanied by relevant information to facilitate the decision-making process of the authorities responsible for the practical implementation of these solutions.

Keywords: multi-objective optimization; mass casualty incidents; location-allocation problem; p-median problem; maximum coverage problem.

MSC: 90C29

check for updates

Citation: Medina-Perez, M.; Guzmán, G.; Saldana-Perez, M.; Legaria-Santiago, V.K. Medical Support Vehicle Location and Deployment at Mass Casualty Incidents. *Information* **2024**, *15*, 260. https://doi.org/10.3390/info15050260

Academic Editor: Tieling Zhang

Received: 19 April 2024
Accepted: 25 April 2024
Published: 3 May 2024

1. Introduction

The International Federation of Red Cross (IFRC) defines a disaster as "an event that disrupts the normal functioning conditions of a community and exceeds its ability to deal with the effects with its own resources [1]". The term natural risks refers to "natural phenomena that have the potential to cause a disaster, such as climatologist, meteorologist, hydrologist, biological or geophysical phenomena". Given the natural conditions of the planet, all geographic regions are exposed to different types of natural risks that have the potential to cause disasters that are inevitable and sometimes unpredictable. However, disasters that could occur from these natural phenomena are also avoidable and it is the responsibility of the population to prepare to reduce its impact in three different stages: before the event (proactive), during the event (active), and after the event (reactive). The World Health Organization (WHO) defines four phases for disaster management: prevention, to minimize the effects of the incident; preparation, to plan the actions to take in each dangerous situation and to educate the population on the subject; response, to act immediately after the incident and provide medical assistance and resources; and finally recovery, to return to normal once the tasks associated with the response phase are completed [2].

If a disaster leads to a large number of casualties beyond the capabilities of healthcare providers, it is considered a Mass Casualty Incident (MCI). Medical problems resulting from an MCI are those defined by the National Association of Emergency Medical Technicians (NAEMT) in the Manual for Prehospital Trauma Life Support (PHTLS) [3] that include the following five elements:

- Search and Rescue: This activity involves identifying and removing victims from a dangerous situation.
- Triage and Initial Stabilization: This is the process of classifying victims according to the severity of their injuries and the primary medical care they require.
- Patient Monitoring: This refers to continuously monitoring a patient's health status from identification to evacuation.
- Definitive Medical Care: It is the component in which specialized care is provided to the patient according to his needs until his recovery is complete.
- Evacuation: This refers to the transfer of the patient from the disaster area to the area of definitive medical care.

These elements describe the tasks carried out in the disaster response phase from a medical perspective, although there are other aspects to consider when participating in an MCI. One of the most widely used risk management tools is the Incident Command System (ICS), which is designed to facilitate the coordination of agencies in different jurisdictions to work effectively together [3,4]. This system is based on an organizational structure for managing the response to any small or large incident. The ICS describes best practices for trained personnel to address the five medically relevant elements safely and efficiently; however, various factors can make the scenario more complex, such as the dynamic nature of a disaster, the uncertainty of the damage caused, and the location of the victims, among others.

A scenario that can occur in large-scale disasters is when a large geographic area is affected (originated from fires, floods, earthquakes, and pandemics), where there is a multitude of demand points and the victims are not concentrated in a single area, which means that medical personnel cannot move between them in a reasonable amount of time. In this scenario, the process of assigning the available resources needs to be analyzed. The ICS recommends that "to avoid the problems that might be caused by the massive convergence of resources to the scene and to manage them effectively, the Incident Commander (IC) may establish the required holding areas".

In these types of situations, units are typically sent to a staging area rather than going directly to the incident location [3]. A staging area, or preparation area, is a location near the incident where multiple units can be held in reserve, and dinamically assigned as required. With this in mind, a hypothetical ambulance allocation proposal is to allocate only those ambulances necessary to care for confirmed victims at each care center, then place the rest of the available ambulances in one (or more) staging areas, and then allocate them to nearby care centers as the presence of victims is confirmed. This assignment means that an ambulance will go to a demand point, then paramedics will evacuate the victim to a hospital and return to the staging area, thus distributing the workload among all available units. However, this requires the selection and activation of staging zones, ensuring the following:

- They are close to the centers of attention to which ambulances will be directed.
- The number of activated zones does not create an unnecessary logistical problem.
- They have easy access to the roads and do not represent an obstacle for vehicle traffic.

For this reason, it is recommended to identify the candidate staging areas (candidate bases) in advance. From these candidates, the areas to be activated can be determined according to the disaster conditions. According to the state-of-the-art, various authors consider hospitals and ambulance bases as starting points that can be used as candidate stating areas. However, this approach is not applicable to all countries and cities considering the local characteristics of the medical service infrastructure. Alternatively, road transport infrastructure can be used to temporarily locate this type of resources, as long as it satisfies the previous requirements. In summary, there are three major tasks to solve this problem:

1. Define a methodology to find candidate areas to reduce the search space, given the number of road segments that can be considered as staging areas.
2. Find the best staging areas to activate from the set of candidate zones, taking into account the travel time to the attention centers and the number of activated zones.
3. For each enabled staging area, assign the focus of attention to cover, considering proximity and capacity.

Taking these aspects into account, there is an opportunity to improve the ability to respond to disasters by optimizing the use of medical assistance resources, reducing the impact on the number of human losses. In the present work, these three tasks are addressed in a specific study area, adapting to local protocols and regulations; however, the methodology developed and the mathematical model of the problem can be adapted to other regions, when relevant information is available.

This article is organized as follows: Section 2 presents related work; Section 3 presents materials and methods; Section 4 presents the obtained results; and Section 5 describes the conclusions and future work. Finally, references are listed.

2. Related Work

This section presents the most relevant research related to the issues analyzed in this paper. The studies include proposals for the location of ambulances and analysis of disaster situations such as earthquakes. Others, although not directly related (such as the permanent or temporary location of medical aid facilities or centers), offer solutions that have inspired the formulation of the strategy proposed in this work.

The problem of facility location and resource allocation (location-allocation problem) has been addressed by many authors from different perspectives. For example, Halper et al. [5] considered the Mobile Facility Location Problem (MFLP) to minimize the cost of moving existing facilities and the cost of traveling to the assigned facility. Obtaining satisfactory results using a dynamic model, the authors proposed to analyze the problem by decomposing it into two subproblems: the first to assign facilities and the second to assign customers. Both are solved in polynomial time using local search neighborhoods. On the other hand, Barojas-Payán et al. [6] considered two problems in the same form: one to determine the optimal locations of warehouses to provide humanitarian assistance and the other to determine the levels of stock in these facilities. In addition, the authors proposed solving two problems: determining the optimal location of temporary medical stations and allocating the minimum number of stations to optimally cover the entire demand area. Similarly, in this paper, the optimal location of temporary ambulance stations and the assignment of the smallest number of stations are studied to optimally cover the entire demand area.

In the location-allocation research area of medical support vehicles (ambulances), Zaffar and colleagues [7] found that the main objective of most state-of-the-art work is related to finding the best distribution of ambulances to maximize coverage and minimize response times. Mohri and Haghshenas [8] analyzed the problem using the Ambulance Location Problem (ALP) as an extension of the Set Coverage Problem introduced by Toregas [9]; Kavhe and Mesgari combined the Maximum Coverage Location Problem (MCLP) and metaheuristics [10]; Hashtarkhani et al. proposed the use of MCLP and combinatorial optimization techniques [11]; while Barojas-Payán et al. [12] addressed the problem using the *p*-median problem.

However, in the particular case of disasters, in addition to maintaining an adequate response time as the primary objective, the location of emergency medical services (both facilities and vehicles that are temporarily or permanently located at a site) frequently must also attempt to satisfy constraints associated with the number of available resources (sometimes insufficient for the demand points), the heterogeneity of demand (that changes over time), and other uncertain or indeterminate factors [13]. For this reason, and following earlier contributions, this paper models this request using a multi-objective function that combines the *p*-median and MCLP (the model description is detailed in Section 3.

Zheng and colleagues [13] presented a collection of algorithms designed to solve combinatorial optimization problems in disaster situations. The algorithms were classified by problem type, such as transportation planning, location of the facility, route, or integrated (combining multiple problems). The proposed algorithms formulate functions with one or more objectives and some are considered uncertain or random factors. In the collection, Genetic Algorithms (GAs) were the most used, followed by Particle Swarm Optimization (PSO), and other heuristic techniques such as Simulated Annealing (SA), Tabu Search (TS), Harmony Search (HS), Variable Neighborhood Search (VNS), as well as hybrid models that combine different metaheuristics to solve the problem. The NGSA-II algorithm was used for evacuation planning and transportation costs by Saadseresht et al. [14]; for relief planning by Nolz et al. [15]; and for dynamically adjusting supply distribution plans according to demand requirements, delivery time, and transportation costs by Chang et al. [16].

In this work, the behavior of the NGSA-II algorithm for finding the temporary locations of emergency vehicles is studied, which is supported by the results of Schjølberg et al. [17] and Saeidian et al. [18]. They found in their experiments that for the problem of finding the location of emergency vehicles, the Genetic Algorithm achieves better results when comparing different metaheuristics, achieving a better response time, which is more pronounced in scenarios with a high demand for emergency vehicles.

Particularly in earthquake disasters, the following research proposed resource management models with different considerations. Geng et al. [19] focused on the perception of victims' distress to analyze optimization of storage location and the allocation of supplies for emergency rescue. Alinaghian et al. [20] used harmony and taboo metaheuristics in combination with variable neighborhood metaheuristics to suggest the location of permanent health centers and temporary relief centers under normal and critical conditions, considering the characteristics before and after a disaster. To improve preparedness and response in the first 72 h after an earthquake, Oksuz and Satoglu [21] introduced a multi-objective stochastic programming model to optimize allocation of resources, temporary health centers, and medical personnel. Similarly to [21], where the model considers possible damage to roads and hospitals, including distance restrictions, in this research the characteristics of roads are evaluated.

Real-time decision-making has been studied in several contexts, but it is still challenging [22,23]. Location problems (including the *p*-median and MCLPs discussed in this paper) are NP-hard problems, meaning that they are extremely difficult to solve in terms of computational complexity and cannot be solved optimally for large input data. Moreover, there is no known efficient algorithm that can solve all instances of the problem in polynomial time in an NP-hard problem. To reduce computational complexity, some proposals require the list of possible candidate locations in advance or otherwise include in the methodology an approach to decrease the search space [11,21]. Saeidian et al. [18] used a GIS to select candidate locations that satisfied the initial conditions, followed by applying a metaheuristic to calculate the optimal location of temporary help centers. However, even with a list of candidate locations, conditions can change in disaster situations. Therefore, this article uses the proposal detailed in [24] to find possible locations based on the initial conditions of the problem and to reduce the search space according to the disaster scenario.

Finally, one of the works that is most closely related to the present proposal is described in [25]. Karpova et al. presented a technique for dynamically relocating ambulances that are available at a specific time, attempting to maintain complete coverage that adapts when an ambulance is occupied. Their proposal did not consider large-scale emergency scenarios, but rather everyday medical emergencies in which ambulances are individually assigned to care locations. Likewise, Caglayan and Satoglu [26] proposed a multi-objective model with three objectives: minimize the number of victims unable to be transported to hospitals after the disaster, reduce the number of additional ambulances needed in the response phase, and reduce the total transport time. Caglayan and Satoglu considered the assignment of triage points to emergency stations prior to the event and decided the number of ambulances that may be needed in advance. The number of unattended victims and the time spent in transport were minimized based on their health status.

On the other hand, Wang et al. [27] proposed a model for ambulance location-allocation in disaster management, addressing the identification of both temporary bases and medical centers with funding restrictions. It is a combination of SA, GA, and PSO. The main contribution of this research is the consideration of relocating temporary ambulance stations as the demand rate changes. The work of [28] analyzed the problem of ambulance location-allocation considering the station preference order to dispatch ambulances, the temporal variation in demand and travel time, the probabilities of station-specific ambulance occupancy, and possible everyday ambulance relocation. Their proposal used a particle swarm approach to obtain the number of ambulances assigned to each station. The adaptive VNS metaheuristic was used to solve the problem. However, both [27,28] recommended for future work to extend model experiments with different travel times from temporary stations to demand points, considering different times of day, since in a disaster situation the closest temporary station will not always be the first to arrive. To address this issue, the model proposed in the present paper consulted the traffic data corresponding to the time of day of each scenario analyzed. Thus, the actual travel time (taking into account the level of congestion of different roads) was considered to select the optimal locations to reach the demand points.

3. Materials and Methods

The proposed methodology is shown in Figure 1 and consists of eight stages: Problem Model, Historical Data, Incident Data, Data Integration, Algorithm Selection, Optimization Process, Solution Visualization, and Implementation Plan. The purpose of each stage and its components are described in the following.

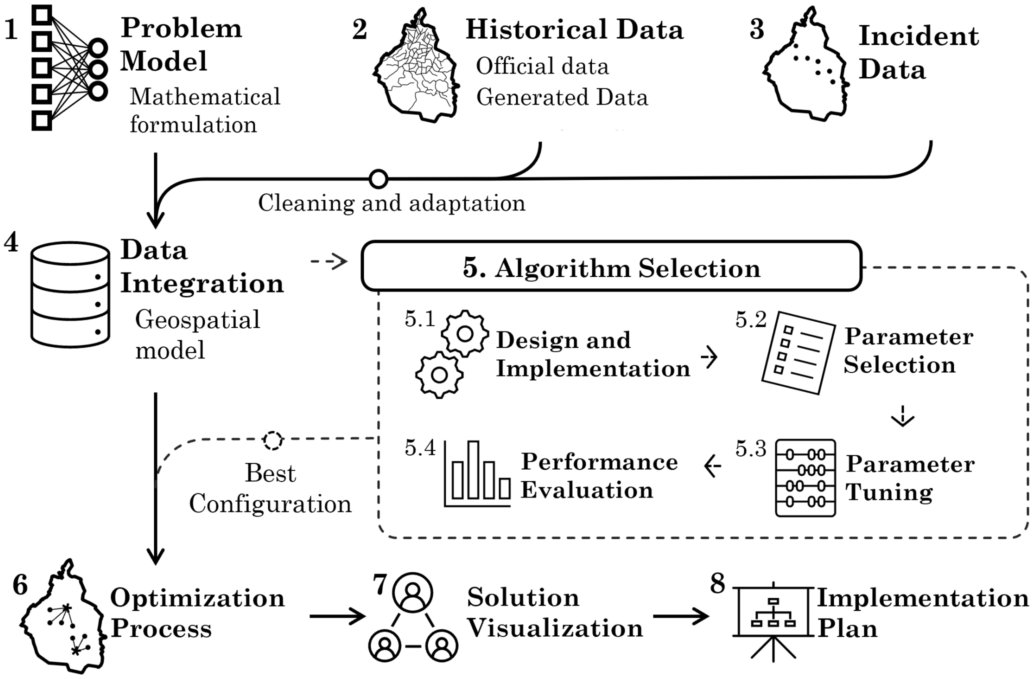


Figure 1. Proposed methodology.

3.1. Problem Model

The problem can be described as a combination of the p-median problem and the Maximum Coverage Problem, with two objectives. Formally, use the following representations:

- b_i is the i-th candidate base (decision variable).
- p_j is the j-th demand point.
- I is the set of candidate bases.
- J is the set of demand points.
- U is the set of urgencies of the demand points.
- a_{ij} is the coverage from b_i to p_j (decision variable).
- C is the cost matrix of $|I| \times |J|$.
- c_{ij} is the cost of travel from b_i to p_j.
- l_i is the capacity of b_i.
- d_j is the demand for p_j.
- u_j is the urgency of p_j.
- α is the maximum cost that a route can have.

The first objective consists of minimizing the cost (distance) between the candidate bases and their respective assigned demand points (see Equation (1)), and the second is to activate the smallest possible number of candidate bases (see Equation (2)). In consequence, the two objective functions are as follows:

$$\text{minimize } f_1 = \sum_{i \in I} \sum_{j \in J} c_{ij} a_{ij} \tag{1}$$

$$\text{minimize } f_2 = \sum_{i \in I} b_i \tag{2}$$

Subject to the restrictions described by the Equations (3)–(10).

$$a_{ij} = \begin{cases} 1, & \text{if } b_i \text{ attend } p_j \\ 0, & \text{other} \end{cases} \tag{3}$$

$$\sum_{i \in I} a_{ij} = 1, \; \forall j \in J \tag{4}$$

$$\sum_{j \in J} a_{ij} \geq b_i, \; \forall i \in I \tag{5}$$

$$a_{ij} \leq b_i, \; \forall i \in I, j \in J \tag{6}$$

$$\sum_{j \in J} d_j a_{ij} \leq l_i b_i, \; \forall i \in I \tag{7}$$

$$max_{j \in J}(c_{ij} a_{ij}) \leq \alpha, \; \forall i \in I \tag{8}$$

$$\sum_{i \in I} b_i \leq \frac{|J|}{2} \tag{9}$$

$$b_i \in \{0,1\}, a_{ij} \in \{0,1\}, \; \forall i \in I, j \in J \tag{10}$$

Taking into account the observation in [29] on the p-median problem, it is established that "if you have a solution with p installations, increasing another installation in any of the candidate nodes that do not have an installation will reduce the total weighted cost. Therefore, the objective function will decrease". However, in the proposed model, adding an installation p deteriorates the second objective, so there is a conflict between the two objectives. Figure 2 shows an example of the relationship between the bipartite graph and the location of the candidate bases and the demand points. In this figure, the cost matrix is represented by straight gray dotted lines, and the values of $a_{ij} = 1$ are represented by red lines. On the other hand, Figure 3 contains the matrices that describe this scenario.

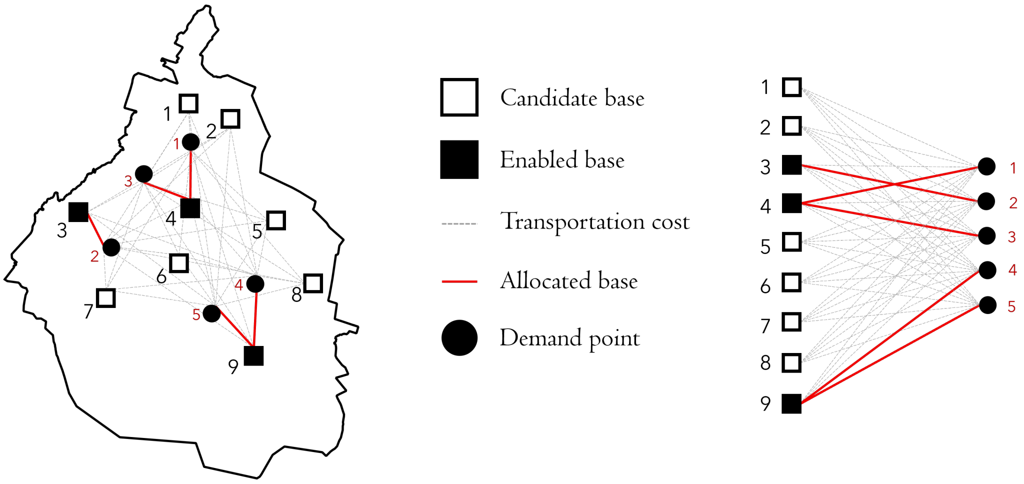


Figure 2. Representation of the problem using a bipartite graph.

Several definitions of the p-median problem specify $a_{ij} = 1 - c_{ij} < c_{ik}$, $i \neq k \; \forall \; i$, $k \in I, j \in J$. However, this is not always possible when the problem has limited capacity. For this reason, a criterion must be defined to process each $p_j \in J$, taking into account demand (d_j), capacity (l_i), distance (c_{ij}), or urgency (u_j) (several examples of urgency calculation can be consulted in [30]). For the processing of each j-th element, it is possible to apply one of the following allocation methods:

- Classic: A p_j is randomly chosen and assigned to the nearest b_i.
- Urgency-based: $u_j, \forall j \in J$ is calculated and processed in descending order of u_j.
- Demand–urgency-based: Elements are processed in descending order of d_j and later in descending order of u_j.

$$c = \begin{array}{|c|c|c|c|c|}\hline 1 & 5 & 6 & 7 & 4 \\ \hline 2 & 8 & 5 & 1 & 3 \\ \hline 6 & 8 & 4 & 2 & 3 \\ \hline 3 & 2 & 1 & 6 & 7 \\ \hline 2 & 1 & 6 & 8 & 5 \\ \hline 5 & 7 & 6 & 1 & 2 \\ \hline 1 & 4 & 3 & 5 & 8 \\ \hline 1 & 6 & 2 & 3 & 7 \\ \hline 3 & 1 & 1 & 7 & 8 \\ \hline \end{array} \qquad a = \begin{array}{|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 \\ \hline \end{array}$$

$$b_i = [0,\ 0,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 1]$$

$$f_1 = 3 + 8 + 1 + 7 + 8$$
$$f_1 = 27$$

$$f_2 = 0 + 0 + 1 + 1 + 0 + 0 + 0 + 0 + 1$$
$$f_2 = 3$$

Figure 3. Decision variables and fitness value of the solution for the scenario represented in Figure 2.

Figure 4 shows the results obtained using the classic, urgency-based, and demand–urgency-based allocation methods for the same set of vectors I and J, with their corresponding capacity and demand values. It is evident that the demand–urgency-based method is less expensive. However, this condition cannot be guaranteed in all scenarios, so it must be determined which method is more appropriate for a particular case and considered as a hyperparameter for further adjustment.

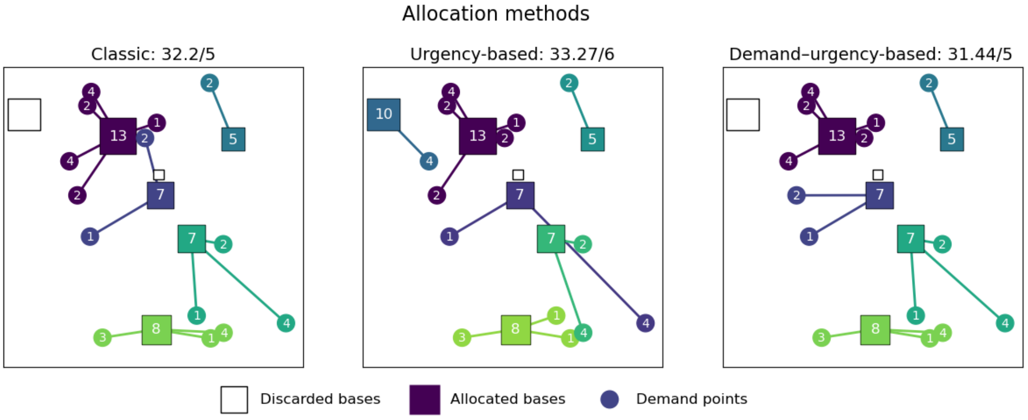


Figure 4. Example of allocation methods.

3.2. Historical Data

These is a result of acquiring or generating information from the case study in which the problem is analyzed. Examples of required information include road transport infrastructure, hospital locations, reception centers, temporary refuges, and potential staging areas. Some of these data were obtained from open access databases and official sources. The remaining data set is generated specifically for the problem.

3.2.1. Official Data

These data were obtained from the Mexico City Open Data Portal (Portal de Datos Abiertos de la Ciudad de México, in Spanish) and the Risk Atlas Portal (Atlas de Riesgos, in Spanish). All layers were modified to homogenize common characteristics and reprojected to SRS EPSG:6369, Mexico ITRF2008/ UTM Zone 14N. The names of the layers and each attribute are written in lowercase, and a crop was applied using the boundary of the polygon of Mexico City. The layer list and related information are described in Table 1.

Table 1. Data retrieved from official data sources.

Layer	Description	Geometry Type	Number of Records	Features
mayorships [31]	Geographic boundaries of each mayorship	polygon	16	6
risk_zones [32]	Earthquake risk zones	polygon	4908	2
regions * [33]	Classification of urban areas according to the unit responsible for rescue operations	polygon	5	3
hospitals [34]	Public and private hospitals operating in the CDMX	point	584	59
gathering_centers [35]	Emergency collection centers available	point	67	10
refuges [36]	Temporary emergency shelters available	pint	67	10
collapses [32]	Collapsed buildings from the 2017 earthquake	point	38	1
damages [32]	Damaged buildings from the 2017 earthquake	point	38	1
blocks [37]	Geographic delimitation of blocks and 2020 population census data	polygon	1,215,066	18
seduvi [38]	Buildings Centroid and their characteristics, provided by SEDUVI	point	1,215,066	18
land_registry [39]	Geographical delimitation of the buildings and their characteristics, provided by Land Registry Department	polygon	1,607,109	7

* Data created from the description.

3.2.2. Generated Data

The set of geographic data required for the multi-objective optimization process is obtained and adapted in this stage. It consists of obtaining data on the road transport network and the set of possible locations for candidate bases.

Road Transport Network. The OSMNX Python library is applied to generate a graph using the Mexico City polygon as a reference. A multi-digraph object is obtained from NetworkX, where the edges represent the roads and the vertices describe their intersections. All components of the graph are georeferenced with the SRS EPSG:4326, so a reprojection is applied to the system EPSG:6369.

Candidate bases. Several approaches assume that the candidate locations where the ambulances will be placed are known in advance and are typically hospitals. In Mexico City, there are no official data, but in practice the same road transport infrastructure is generally used to place ambulances. Applying this alternative to other models, each road could be considered a candidate location, i.e., there would be 305,190 possible locations using the roads individually, or 128,825 considering only the intersections as starting points. However, the search space for the location problem would be $2^{128,825}$, which is not computationally efficient. Therefore, in this work, a simplification process based on [24] is proposed to segment the road transport network, described as a road transport graph $G_t = \{V_t, E_t\}$, and to obtain a filtered graph $G_f = \{V_f, E_f\}$, such that $G_f \subset G_t$, considering a limit representing the minimum capacity (μ) and a proximity factor (λ). The process used is described in the following:

1. Selection by characteristics: Roads with any of the following characteristics are discarded:
    - It is a highway, incorporation, residential street, or it has no classification;
    - It has fewer than 3 lanes;
    - It is represented with a bidirectional edge and has fewer than 4 lanes;
    - The number of lanes is unknown and it has a lower hierarchy than the type of secondary road.
2. Selection by capacity: The road segments are discarded where it is not possible to assign a minimum of μ ambulances. On average, the length of an ambulance is considered equal to 6 m, and the use of $\mu = 15$ removes all segments that are shorter than 90 m. Furthermore, isolated vertices are eliminated and the filtered graph G_f is updated with the capacity factor.

3. Elimination of redundancy: Roads are represented by their terminal vertex, i.e., the vertex v of the road (u, v), because this is the point from which an ambulance will start, and intersections of two or more roads can have a common terminal node. To avoid redundancy, roads that share a terminal node v are grouped, but their individual characteristics are preserved. As a result, a subset of vertices $V_s \in V_f$ is obtained, which stores the attributes of the edges $\{(u_1, v), (u_2, v) \rightarrow (v) \mid u_1, u_2, v \in V_f\}$ and contains two new attributes, length and total capacity, which are the sum of the lengths and capacities of the grouped edges, respectively.
4. Proximity grouping: The vertices of the set V_s that are "close" are grouped according to a proximity parameter λ. This proximity factor is measured by the path from point $a \in V_s$ to point $b \in V_s$ using a path in G_f. Using a clustering algorithm, information from multiple vertices $(w_1, w_2, ..., w_n)$ is added to a representative vertex v.

A reduction is applied when obtaining the network using OSMNX, in which the edges representing connected roads without intersections are merged, i.e., the edges $(u, v), (v, w)$ are merged into (u, w) if the vertex v has no neighbors (either ancestors or predecessors) except u and w. For this reason, 21,454 edges in the graph represent more than one road, having multiple attribute values such as name, type, and number of lanes, among others. In the previous simplification process, the merged roads were considered a single road of the type of lower hierarchy present in the merged roads, i.e., if a primary road and a secondary road have been merged, the new edge is considered a secondary road but preserves the original attributes of the merged roads. To verify the coverage that the bases have throughout the city, the distance between each vertex of G_t and its nearest base is evaluated using the Voronoi diagram of the graph with the resulting candidate bases as control points, i.e., the set V. To detect possible changes in this coverage, different weights can be applied to the edges using historical traffic data.

3.3. Incident Data

In this stage, event information is collected from the location of the demand points, considering the confirmed or possible number of victims, as well as blocked roads, traffic data, and available hospital capacity. In a real scenario, these data are obtained from government sources as the information is being integrated. In an experimental scenario, historical data from previous incidents are used.

Location of demand points. The CDMX Open Data Portal provides georeferenced data on buildings that collapsed or suffered damage during the 19 September 2017 earthquake, which can be used to estimate the probability distribution of damage locations that could occur in future events. It is proposed to generate new hypothetical scenarios that follow this spatial distribution and enable an adjustment of the algorithm parameters based on different instances of the same problem, using the kernel density estimation of the buildings damaged in the 2017 earthquake, and using the library scikit learn and a Gaussian kernel. To determine the bandwidth of the kernel, a grid search is performed (in one dimension) with 1000 values of powers of 10, from 10^1 to 10^4. For each bandwidth value, the KDE is obtained using the leave one out validation method, and selecting the maximum likelihood logarithmic estimation.

In addition, the demand for each of them has to be estimated. In a real scenario, the responsible official reports the location of the incident and its characteristics to make an initial estimation of the damage per quadrant [40]. This estimation includes details of damaged buildings (does not change), as well as the number of victims (may change as more specific information becomes available). The official data reported in the September 2017 earthquake do not have this information; therefore, it is proposed to make an approximation of the number of victims by using the points generated with the kernel density, the data from the land census, and the SEDUVI, which requires the following modifications:

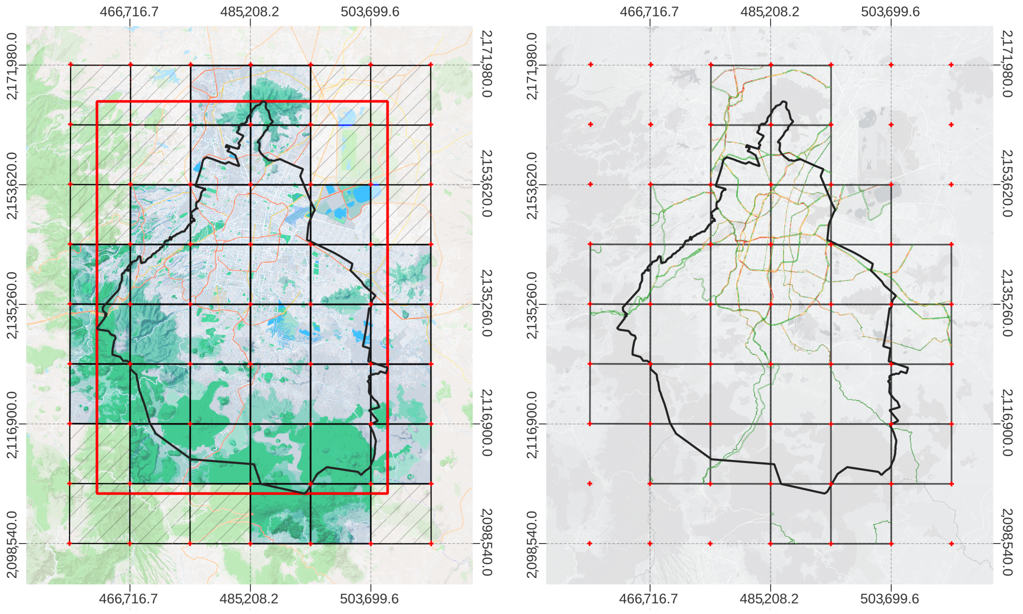
- Only residential buildings with information on the number of levels are considered.
- If a building does not have the number of levels but the height, then half of the height in meters is used as the number of levels.

- For blocks that do not have the average number of occupants, the average of the surrounding blocks is used.

To estimate the number of victims in each p_j, the following steps were performed:

- n points are sampled using the kernel with $15 \leq n \leq 60$;
- Search for the nearest building in the land records layer using Euclidean distance.
- The block in which each building is located is identified by calculating the Euclidean distance between the building and the centroids of the blocks.
- The average number of occupants per apartment p_{ocup} in the block is taken and rounded to the closest integer using the ceiling function.
- If the building has three floors or less, it is considered a single-family unit, so the number of victims of the block is equal to $n_{victims} = p_{occup}$.
- If the building has more than three levels, it is considered a multi-family building, and the average number of residents in the block is multiplied by the number of levels $n_{victims} = n_{levels} \times p_{occup}$.

Traffic data. Traffic data can be obtained in real time through TomTom's TrafficFlow API according to the specifications provided on the website, which is in raster format and segmented into fixed-size rectangular tiles, representing an exponential division of space [41], as is described in Figure 5a. To optimize the memory required to maintain all tiles, the data are converted to single-band rasters that preserve the different traffic classes by mapping the RGB components with a range of $[0, 255]$ to a class index of $[0, 6]$; see Figure 5b.



(**a**) (**b**)

Figure 5. Mexico City traffic on 20 October 2023 at 18:58 obtained with a zoom level of 12. (**a**) Tiles required to cover the bounding box in Mexico City (red box). (**b**) Final result of conversion to single-band raster.

When tiles are available as single-band raster data, a sampling is performed for each edge of the graph. This is performed by calculating the center of the sampling window, which is the end of the line segment perpendicular to the intermediate line segment of the edge geometry and crossing its center. The final result of the extraction of traffic data is a Pandas dataframe that associates the *id* of each edge (u, v, k) with an integer value for each color that corresponds to the most commonly used value in the sampling window.

However, if the sample window does not contain a number other than 0, it is assumed to represent free traffic segment and the number 1 is assigned.

3.4. Data Integration

At this stage, the Historical Data and Incident Data are concentrated and processed. Analysis is performed to ensure that all geographic data are in the same coordinate system and that all files are properly formatted. The incident data update the status of previous data, with features such as the possibility of traveling along the road axes, obtaining a geospatial model that describes the complete scenario of the incident. Specifically, to obtain the required Geospatial Model, it is necessary to apply the procedure described below.

3.4.1. Upload Information

The data previously obtained must be loaded, processing the following information:

- Candidate locations as temporary bases $I, l_i, \forall i \in I$. Since bases are a subset $V \in V_t$, they are identified by the *id* of the vertex in G_t, i.e., the *node* attribute is used as an index;
- The road transport graph $G_t = V_t, E_t$;
- The location of hospitals with more than 10 beds;
- The points where the victims are located are identified by an *id* denoted *num*;
- Traffic data associated with the incident.

3.4.2. Update Previous Data with Incident Data

For each particular scenario, the following data must be modified before performing an optimization technique, in order to consider the specific characteristics of each incident:

1. The traffic information is included in the traffic graph G_t as an attribute *(traffic)* of the edges. If historical traffic data are used, all available data collected on the same day of the week with the closest time to the event are obtained.
2. A version of the graph is obtained in which there are no vertices that cannot be reached from a candidate base. Since the simplification of G_t excludes unclassified roads, some points cannot be traveled to from a temporary base. This does not mean that they cannot be accessed, but only that they will not be considered as a destination point and the nearest available edge or vertex will be considered instead. To determine which points must be discarded, the Voronoi diagram of G_t is calculated using the candidate bases as control points. Vertexes that are not grouped in any cell are eliminated from G, resulting in a graph G_u.
3. The nearest edge to each demand point and to each hospital in the graph G_u is identified. These edges are stored with their identifier (u, v, k) in independent lists that will be used later.
4. Edges of G_t that are not traversable are identified, either because they have a value of 5 or 6 in the traffic classification, or if it is the closest edge to p_j and is found in the list of edges from the previous step. All edges of G_t are assigned a state attribute *locked* and marked *true* for those identified in this step.
5. The edge costs of G_t are calculated as the *weight* attribute as follows:
    - If the value of the *block* attribute is *false*, the value corresponding to the mapping *traffic→weight* is assigned as the value of the *weight* attribute:
        - 🟩 $1 \rightarrow 1 \times l$;
        - 🟨 $2 \rightarrow 2 \times l$;
        - 🟧 $3 \rightarrow 4 \times l$;
        - 🟥 $4 \rightarrow 6.7 \times l$.

      where l is the length of the edge and the coefficient is the inverse of the speed, expressed as the distance.
    - On the other hand, if the *lock* attribute is *true*, the maximum weight is 1×10^9.

6. Candidate bases that satisfy any of the following conditions will be identified:
   - It has no successors in the graph G_t (this can occur at the boundaries of the zone).
   - Some of the grouped edges are found in one of the lists in step 3, i.e., there is a demand point or a hospital on one of the streets it represents.
   - They have a traffic value greater than 2.

   These bases are discarded for use in the optimization problem, so they are not included in the calculation of the cost matrix.
7. The graph G_t is updated with the new edge attributes.

3.4.3. Computation of the Cost Matrix

The resulting set I is used to calculate the origin–destination matrix (matrix C). To define the targets (set J), we take the two endpoints of the nearest edge to each demand point and use the vertex with the highest cost to consider the worst-case scenario. If the most expensive vertex to reach requires traveling through the blocked path, the lowest-cost point is considered. The result is the set J, where each element represents the closest vertex of the graph G_t to the true location of each demand point with the estimated number of casualties.

Assuming $|J| < |I|$, it is more efficient to compute G_t^{-1} and run Dijkstra's algorithm $|J|$ times than to use G_t and run the algorithm $|I|$ times. The result is the cost of traveling from each vertex of G_t to p_j, which is stored in a Pandas dataframe that relates the *id* of each vertex and its distance to j. In addition, another dataframe is stored that relates each vertex to its predecessor. Finally, the dataframes define the cost matrix, where each row represents a base $i \in I$ and each column represents a demand point $j \in J$. The relationship between position (row or column number) and base identifier (*node* attribute) or demand point (*num* attribute) is stored to know which indices i and j correspond to each base and demand point, respectively. The implementation of Dijkstra's algorithm is based on the cuGraph library.

Demand estimation. Since each b_i has limited capacity and the total number of ambulances must be distributed to cover all the demand points, the value of $d_j, \forall p_j \in J$ is estimated with the ratio between the number of victims at that demand point and the total number of victims. After identifying the closest activated bases, the number of ambulances assigned to each group can be adjusted, provided that there is a tolerance in the total capacity of each base. To have a tolerance in the capacity of each base, half of the total capacity is taken as a value. Equation (11) describes the calculation of demand (d_j) for a given demand point (p_j).

$$d_j = \text{round}\left(\frac{n_{victims_j}}{\sum_{j \in J} n_{victims_j}} \times (k - |J|)\right) + 1 \tag{11}$$

where $n_{victims_j}$ is the number of victims in p_j and k is the number of ambulances that are available to meet the total demand. The number of ambulances in Mexico City varies, although most of the sources consulted agree that there are more than 290 ambulances. Considering that not all of them will be available at the time of a disaster, it is proposed to use the number of 250 ambulances.

Instance storage. When all the data used to describe the problem are available, the data are stored as Python objects. This process is performed for each of the 10 hypothetical scenarios proposed. Each scenario has its own set of demand points and random traffic data for one week in November at different times.

3.5. Algorithm Selection

In multi-objective optimization problems, one of the most widely used algorithms is the NSGA-II algorithm, both in computational engineering problems and in other areas [42–44]. For this reason, the NSGA-II Genetic Algorithm was used and in order to improve the performance of this algorithm; its parameters were modified by using the following procedure.

3.5.1. Design and Implementation

The implementation of the NSGA-II algorithm is carried out using the Python library *pymoo*. The algorithm works with solutions using a feasible first selection method, as described in the [45] library documentation. To solve the optimization problem, several objects should be defined.

Problem and solution representation. This is an object that describes the mathematical model of the problem.

Algorithm. It is the object through which the proposed algorithm is carried out. It receives as input the parameters of the algorithm and three additional parameters:

1. Remove duplicates: A *boolean* parameter that indicates whether or not solutions that are the same in the current population should be removed. There is no guarantee that there will be no repetition of solutions from previous generations that have been removed. In this paper, this value is set to *true*.
2. Stop criteria: This is an object that contains the criteria that determine when the algorithm stops, which remain fixed for all configurations with the values described in Table 2. These values were chosen after a tuning stage, defining test scenarios with a high maximum number of generations, $n\_max\_gen = 10,000,000$, and analyzing the behavior of both *xtol* and *ytol*. The smallest values found were higher than 0.0001 and the best point was reached in fewer than 500 generations; therefore, this limit was used for *n_max_gen* as well as 0.0001 for *xtol* and *ytol*. The number of evaluations of the objective function was determined considering the worst case, assuming 200 subjects in the population, and supposing the limit of 500 generations is reached, the result is 100, 000. The 100 value used in *period* and *n_skip* prevents evaluating the stop criteria in each generation, but different values can be assigned. Tolerance values are used to compare the Pareto fronts of the current generation P_n and a generation from a previous period $P_{n-period}$, each *n_skip* generations. This comparison depends on the space in which it is performed:
   - In the decision space *(xtol)*, two Pareto fronts P_n and $P_{n-period}$ are compared using Equation (12):

     $$xtol = \frac{1}{|X|} \sum_{b \in X} \left(\frac{\max_{i=1}^{|Y|} \left(\sum_{j=1}^{|b|} (b_j \neq Y_{ij}) \right)}{|b|} \right) \tag{12}$$

     where $P_n = [f_1(X), f_2(X)]$ and $P_{n-period} = [f_1(Y), f_2(Y)]$.
   - In the target space *(ftol)*, the Inverse Generation Distance (IGD) is calculated between the normalized fronts to be compared, that is: $ftol = IGD(\overline{P_n}, \overline{P_{n-period}})$.
3. Repair function: Depending on the allocation method, the different solution vectors b can change and be reduced to the same solution vector, because activated bases that do not correspond to any demand point are discarded. There are two ways to repair:
   - Normal repair: The assignment process is performed and the modified solution vector b that satisfies the required constraints is returned to the algorithm to remove duplicates in the current population.
   - Repair with mutation: The assignment process is performed and the modified solution vector b is checked; if this vector has already been obtained after the assignment process, regardless of the number of generations, the solution vector is mutated and the assignment process is performed iteratively until a valid modified vector that has not been evaluated before is obtained, that is, an unexplored solution.

   Since the assignment process is performed during repair and then before evaluation (for new solutions), it is important that the assignment is the same at both times, that is, the same b always produces the same a. In the case of the urgency-based and demand–urgency–based allocation methods, this is not a problem, since they are deterministic methods; but in the case of the classic allocation method, the first assignment must be recorded for future processing, since it has a stochastic component. For this purpose, a dictionary is used for storing the pairs (b,a), where b is encoded with the hash method *sha256*.

Table 2. Values used in the tuning phase for the values of the stop criteria.

Parameter	Name	Values	Selected Value
Tolerance in the decision space	xtol	$[0.01, 0.0075, 0.005, 0.0025, 0.0001, 0.000075, 0.00005]$	0.0001
Tolerance in the target space	ftol	$[0.01, 0.0075, 0.005, 0.0025, 0.0001, 0.000075, 0.00005]$	0.0001
Criteria evaluation period	period	100	100
Ignore generations for criteria evaluation	n_skip	100	100
Maximum number of generations	n_max_gen	≤10,000	500
Maximum number of evaluations of items	n_max_evals	≤10,000,000	100,000

3.5.2. Parameter Selection

Since each algorithm requires a specific set of parameters, the correspondence between the required parameters and the information described in the Geospatial Model, categoric or numeric variables, is determined. Categoric variables can be different methods of performing an internal procedure of the algorithm, such as the assignment process, crossover and mutation operators, and the metric used to measure performance, among others. Numeric variables are the hyperparameters of an algorithm that can take different integer or real values, such as the number of individuals, the percentage of crossover and mutation, and the number of offspring, among others. For numeric variables, it is important to define a finite set of options to limit the search space. Table 3 shows the parameters proposed to find a configuration that solves the problem in a reasonable time.

Table 3. Proposed values for algorithm parameters.

Parameter	Name	Key	Type	Values
Crossover operator	crossover	–cr	categoric	un—cross uniform hun—cross half uniform
Mutation operator	mutation	–mu	categoric	bf—bit flip mutation bfo—active bit flip mutation
Allocation method	allocation	–a	categoric	cl—classic ur—urgency-based urc—demand–urgency-based
Repair method	repair	–rep	categoric	rn—normal repair rm—repair with mutation
Population size	population	–pop	categoric [1]	(20, 50, 100, 200)
Percentage of offspring	offsprings	–dec	categoric [1]	(25, 50, 75)
Crossing percentage	prob_crossover	–prob_cr	categoric [1]	(0.5, 0.75, 0.9, 0.99)
Mutation percentage	prob_mutation	–prob_mu	categoric [1]	(0.5, 0.25, 0.1, 0.01)
Start probability	prob_initial	–prob_ini	categoric [1]	(0.25, 0.1, 0.01)

[1] Numeric attributes that, according to the documentation, should be converted to categoric attributes [46].

The biased bit flip mutation operator uses a probability of 0.5 for active bits (value 1) and zero for inactive bits (value 0). In addition, for each inverted bit, another bit is inverted that is different from all the bits that were originally inverted, so that the number of active bits remains intact after the mutation. Note that the probability of mutation refers to the probability of a chromosome mutating, and each bit within the chromosome has a fixed probability of mutation of $\frac{1}{|I|}$. Similarly, the probability of crossover represents the probability that two chromosomes will recombine, and each bit has a probability of inheriting children of 0.5.

Furthermore, the parameter α is kept fixed for all instances and the ideal value of 9000 m is proposed, that is, 9 min of response from the candidate base to the demand point, under the assumption that the ambulance travels at a constant speed of 60 km/h. The value of 9 min is a standard recommended by [47] and is used in the models described in [7,48]. However, this is an ideal case, so they propose the range [9–13] min. Our assumption is that there is at least one ambulance on standby in each location, which is dispatched the moment a victim is confirmed, and another unit is immediately requested from the corresponding base, even if no other victim is confirmed. Therefore, a response time in this range would only be necessary in the worst case.

3.5.3. Parameter Tuning

The parameter tuning technique irace was used through its official library in R. Since this technique is designed for single-objective algorithms, it must be adapted to a multi-objective problem by representing the Pareto front by a scalar value [46]. There are several ways to scalarize the Pareto front. Some require knowledge of the ideal Pareto front to quantify their similarity. Since no ideal front is known, these alternatives are not an option. On the other hand, in the implementation guide irace [46] suggests the use of the hypervolume indicator, which requires a reference point to scalarize a Pareto front that can be used as a comparison metric between different algorithms or configurations.

The reference point for this metric should be chosen to ensure that every solution on the Pareto front contributes to the hypervolume. For this reason, it must be greater than the maximum value of each objective. To guarantee this, the point $(1, 1)$ is defined as the reference point and both objectives are normalized from 0 to 1 by dividing the value of each by the maximum possible value, i.e.,

$$f_{O_{normalized}} = \frac{f_{O_i}}{f_{O_{i_{max}}}}, \forall i \in P, O \in [1, 2] \tag{13}$$

where P is the Pareto front obtained by the algorithm, and the maximum $f_{O_{max}}$ for each objective O is calculated using Equations (14) and (15):

$$f_{1\_max} = \alpha \times |J| \tag{14}$$

$$f_{2\_max} = \frac{|J|}{2} + 1 \tag{15}$$

Regarding the implementation of irace, the parameter files are defined together with the scenario and *target-runner.py*, which is a Python script that provides the interface between R and the program that receives the parameters and implements the algorithm. In the scenario, a budget of 5000 iterations is defined and the execution method is "*execute_threaded_timeout*" with a maximum of two attempts per configuration and a maximum time of 420 s (7 min), after which the operation of irace is aborted. To avoid interrupting the execution of the *target-runner.py* script, a maximum time of 6 min is defined for the internal operation of the algorithm. This is based on the assumption that a reasonable time to obtain a solution is 5 min plus the time it may take to load the data or generate the instance data. This means that the target-runner time limit will always be reached, because the optimizer itself stops the execution when the 6 min limit is exceeded and returns a value. If it finishes in time, it returns the value of the inverse of the hypervolume of the Pareto front obtained, and if it exceeds the time, it saves the configuration parameters and returns a value of 1 to indicate poor performance and discard it. Note that irace is a minimization algorithm by default, so the inverse of the hypervolume is used as the evaluation metric.

3.5.4. Performance Evaluation

This enables the evaluation of the result of different configurations in different cases of the problem. The result is a set of elite configurations. Their performance can be compared to determine how to implement them in future problems. For this purpose, the five elite configurations with the best performance are taken as a reference, along with the three instances that describe the lowest, median, and highest number of demand points. To observe convergence to the Pareto front of elite configurations, the algorithm is executed 20 times with different random seeds for each elite configuration, obtaining the minimum, maximum, average, and standard deviation values of the hypervolume indicator to compare with the information obtained in the parameter adjustment stage. According to the performance of the configurations and the requirements in case of an emergency, different paths are chosen to find the solutions to present, for example, to the decision manager:

- Run the algorithm with the best performance multiple times using different random seeds until the maximum time is reached.
- Run the algorithm with the different elite configurations obtained as many times as possible without exceeding the maximum time.

For this purpose, the time required for the execution of the algorithm, as a function of its parameters, must be recorded, weighting the importance of the solution quality and the speed with which it is generated. Additionally, the execution times of the last 20 iterations per configuration are recorded. To make comparisons between configurations, the median value (50th percentile) of the inverse of the hypervolume is used as a reference, which represents the maximum value that 50% of the executions will have.

3.6. Optimization Process

This stage performs the algorithm using the parameters with the best performance obtained in previous stages. For new cases, steps 1, 2, 3, 4, and 6 should be executed using the algorithm with the best combination of parameters. After the result is obtained, the decision-maker must choose one of the available options on the Pareto front to decide how the temporary bases are allocated. By performing multiple executions, the multiple Pareto fronts obtained are merged, and the non-dominated solutions separated, among which the final decision for subsequent implementation will be made. To avoid the fact that multiple executions do not represent a disadvantage in terms of sequential execution time, parallelization of the executions is proposed using the joblib library. In the process of determining final solutions, if there are activated bases that cover a single demand point, the actual assignment of ambulances will be made directly to the waiting area of that point, rather than using the closest temporary base determined by the algorithm, since it is not relevant to assign ambulances to a neighboring base.

3.7. Solution Visualization

Once the Pareto front with the candidate solutions is computed, it is possible to plot it to facilitate decision-making. For this purpose, the following elements will be displayed on a map of Mexico City:

- The locations of the attention centers, showing the number of resources required;
- The locations of the activated bases, displaying the number of ambulances;
- The routes from each enabled base to each assigned demand point. They are identified by using a different color for each base and their cost;
- Boundaries of townships;

- The attention regions defined in the earthquake emergency plan serve as a reference to assign a person directly responsible for the administration of each base, depending on where they are located;
- The location of the hospitals and, if available, the number of beds available;
- The location of collection centers and refuges.

Visualization is performed using the geopandas library, which contains functions that allow other functions of the folium library to be implemented indirectly, providing visualization parameters without having to develop an implementation from scratch. The product of using this library for visualization is an HTML file that can be included in a web page for visualization.

3.8. Implementation Plan

Once the decision-maker selects a solution, the implementation plan must be defined to coordinate response activities with different emergency services. Although this stage is not part of the objectives of this research, a proposal is presented to relate the solution to the current earthquake action plan. Initially, more information is obtained from the selected solution, as follows:

- **Voronoi diagram with activated bases:** The Voronoi diagram of the graph G_t can be obtained using activated bases $b_i = 1$ as control points. In the case of the external Voronoi diagram, the routes for each vertex of the graph are obtained as the destination and the closest activated base as the origin, which can be useful for addressing other incidents that occurred after the current solution was chosen. In the case of the internal Voronoi diagram, the route can be obtained from any vertex of the graph G_t to its closest activated base, allowing one to identify the shortest route that ambulances can take to the assigned base from any point they encounter.
- **Voronoi Diagram with hospitals:** Obtaining the internal Voronoi diagram of G_t using the vertices closest to each hospital as control points, we obtain the shortest path from any other vertex in G_t to the nearest hospital. This can be useful in determining which hospital to take patients to, depending on the location of the victims and whether it is an isolated victim transported in a private vehicle or a victim from a care center transported in an ambulance.
- **Attention flow:** Once the available ambulances have been assigned to activated bases as preparation points, a person in charge of each is designated and the commander in charge of the attention centers is notified that they will be served by that base. The sequence they must follow to distribute the workload is as follows:

  - At least one ambulance must remain in the waiting area near the attention center.
  - Once a victim is confirmed, he is assigned to the ambulance in the staging area and the appropriate staging area is notified to dispatch another unit.
  - The victim is transported to the nearest hospital and the ambulance returns to the same staging area from which it originated, taking the last spot in line. On the way back to the staging area, the operator of the ambulance has the choice of fuel loading.
  - If one of the bases runs out of ambulances and still has demand, ambulances from other bases with lower demand can be arbitrarily reassigned, omitting the formula for estimating demand and adapting the solution to the new emergency conditions.

- **Changes in the Emergency Situation:** Although a static model is considered at the time the optimization algorithm is performed, some actions are required to adjust the solution to the new conditions. First, when a significant number of headlights are introduced, the algorithm is repeated, reassigning the ambulances from the old solution to the closest bases from the new solution. If the changes are primarily traffic-related and there is an Internet connection, it is possible to perform the traffic query to generate the new weights of the edges with these data, and then compute the routes from the bases to the ambulances. However, since the bases are already activated, instead of performing the entire optimization process from scratch, it is sufficient to obtain the external Voronoi diagram with the bases as control points to find the new paths. In this case, the Voronoi algorithm would update the assignment vector a based on the previous solution vector b, but it cannot ensure that the constraints defined in the model are satisfied.

4. Results

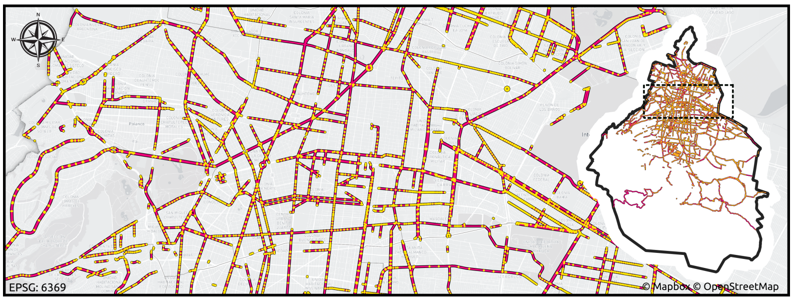
The computing equipment used to implement the methodology has an Intel Core i9-9900K processor and NVidia GeForce RTX 2080 graphics card; consequently, all execution time measurements of the procedures are relative to these characteristics.

4.1. Generated Data: Candidate Base Locations

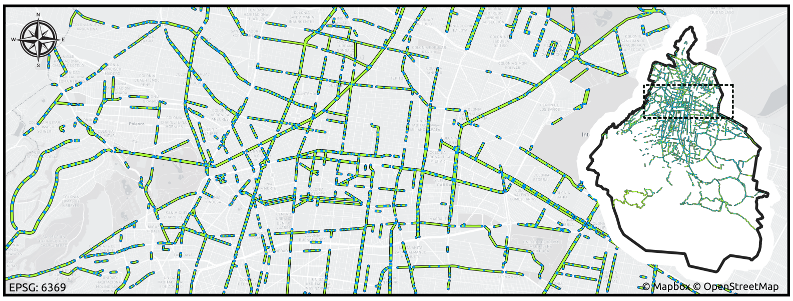
In order to solve the proposed problem, a data set must be generated that describes the candidate base locations that can be used as temporary ambulance bases. Table 4 describes the number of resulting elements after each step described in the previous section. Furthermore, the steps to obtain candidate locations in a specific area are sketched in Figure 6.

Table 4. Reducing transport network.

Step	Result	Size
Transport network	$\{V_t, E_t\}$	128,825
Selection by characteristics	$\{V_f, E_f\}$	15,560
Selection by capacity	$\{V_f, E_f\}$	8,184
Elimination of redundancy	V_s	5,784
Proximity grouping	V	1,846

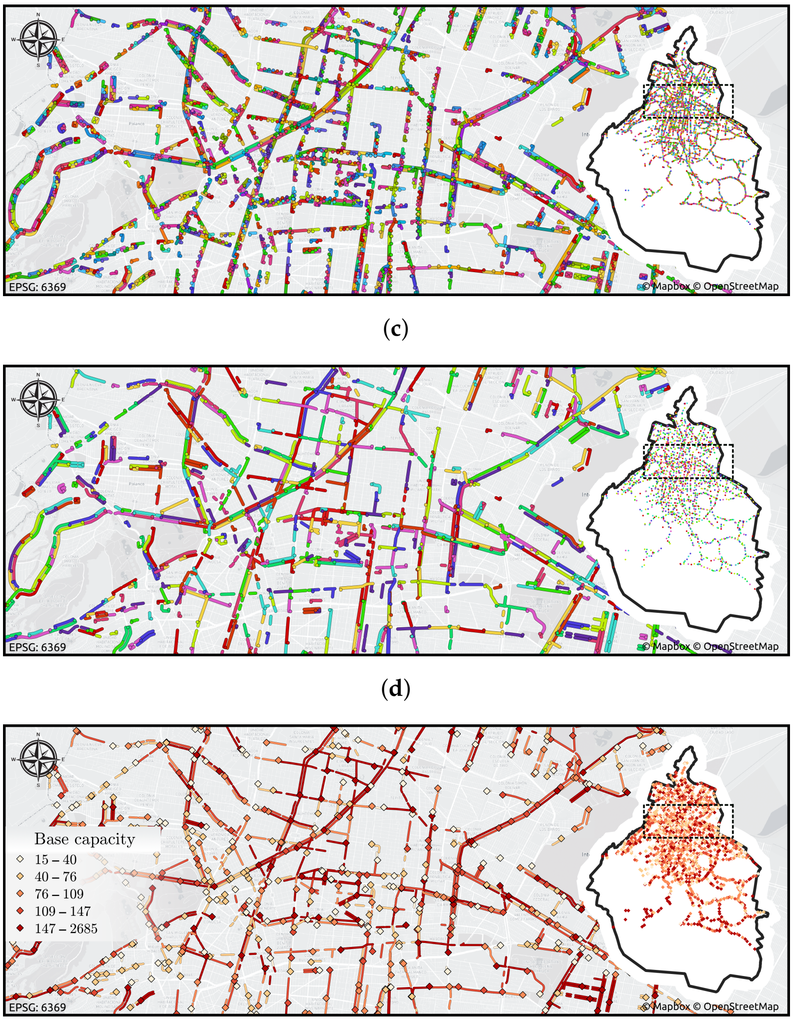


(**a**)



(**b**)

Figure 6. *Cont.*



(**c**)

(**d**)

(**e**)

Figure 6. Process for obtaining candidate bases. (**a**) Selection by features, vertices, and resulting edges. (**b**) Selection by capacity, vertices, and resulting edges. (**c**) Redundancy elimination, representative vertices, and their associated edges. (**d**) Proximity grouping, representative vertices, and their associated edge groups. (**e**) Candidate bases colored by capacity.

To perform coverage analysis, we applied the weights corresponding to the average traffic at each edge according to the defined hours, considering the traffic speeds of 20, 30, 40, 40, 50, and 60 km/h. Applying the Voronoi diagram algorithm, the data in Table 5 were obtained. This table presents the percentage of graph vertices that are less than 9 min and the response time at which 95% of the vertices are located (95th percentile). This table reveals that the difference between coverage at different times does not vary substantially. In particular, at 7 and 8 p.m. and 3 p.m., the percentage of vertices covered by the nearest base is smaller for the speed 20 km/h. Note that higher speeds increase the percentage of vertices that can be reached from a candidate base in less than 9 min. Starting at 50 km/h, all vertices satisfy this property, which is essential to justify the maximum cost constraint used in the problem model.

Table 5. Response time to V_t vertices from the closest candidate base, % = percentage of vertices covered in less than 9 min, p95 = maximum response time for 95% of vertices (95th percentile), and $\mathcal{X}$ = number of candidate bases that are discarded by level traffic > 2.

Daytime	$\mathcal{X}$	20 km/h		30 km/h		40 km/h		50 km/h		60 km/h	
		%	p95	%	p95	%	p95	%	p95	%	p95
7:00 a.m.	47	0.96	8.15	0.99	5.44	0.99	4.08	1.0	3.26	1.0	2.72
8:00 a.m.	69	0.96	8.25	0.99	5.5	0.99	4.13	1.0	3.3	1.0	2.75
9:00 a.m.	66	0.96	8.17	0.99	5.45	1.0	4.09	1.0	3.27	1.0	2.72
11:00 a.m.	52	0.96	8.09	0.99	5.39	1.0	4.04	1.0	3.24	1.0	2.7
1:00 p.m.	74	0.96	8.39	0.99	5.59	1.0	4.19	1.0	3.36	1.0	2.8
3:00 p.m.	122	0.95	8.75	0.99	5.84	0.99	4.38	1.0	3.5	1.0	2.92
5:00 p.m.	57	0.96	8.47	0.99	5.65	0.99	4.23	1.0	3.39	1.0	2.82
7:00 p.m.	139	0.95	9.0	0.98	6.0	0.99	4.5	1.0	3.6	1.0	3.0
8:00 p.m.	55	0.95	8.71	0.99	5.81	0.99	4.36	1.0	3.49	1.0	2.9
9:00 p.m.	21	0.96	8.45	0.99	5.64	0.99	4.23	1.0	3.38	1.0	2.82

4.2. Generated Data: Demand Points

The Kernel Density Estimation (KDE) used to generate the hypothetical scenarios is shown in Figure 7, where the best bandwidth value found (1365) was used. The projection of the KDE on the x (longitude) and y (latitude) axes is also observed with the projection of the original data used to generate the KDE (black lines). In total, 10 hypothetical problem scenarios were generated, each associated with a random date and time from which traffic data can be obtained to define a problem instance. Figure 8 shows the scenarios with the lowest and highest number of demand points. To generate an instance of the problem, each hypothetical scenario is associated with a time and date from which traffic data can be obtained. The characteristics of these instances are described in Table 6.

Table 6. Characteristics of the instances created to describe demand points, θ: instance, $\mathcal{X}$: number of candidate bases that are discarded, n_v: number of victims, d: demand.

θ	Date/Hour	$\mathcal{X}$	$\|I\|$	$\|J\|$	Total n_v	Min n_v	Max n_v	Min d	Max d
0	27 October–09:00 a.m.	319	1,527	43	334	2	45	2	26
1	27 October–03:00 p.m.	482	1364	45	463	2	45	2	19
2	28 October–11:00 a.m.	265	1581	44	381	2	45	3	17
3	29 October–07:00 p.m.	227	1619	38	515	2	45	2	67
4	30 October–1:00 p.m.	240	1606	19	193	2	45	5	44
5	30 October–05:00 p.m.	235	1611	53	570	2	45	2	16
6	31 October–07:00 a.m.	306	1540	23	192	2	45	5	29
7	1 November–08:00 a.m.	267	1579	37	275	2	45	3	16
8	1 November–08:00 p.m.	280	1566	35	376	2	45	3	28
9	2 November–09:00 p.m.	139	1707	26	302	2	45	2	34

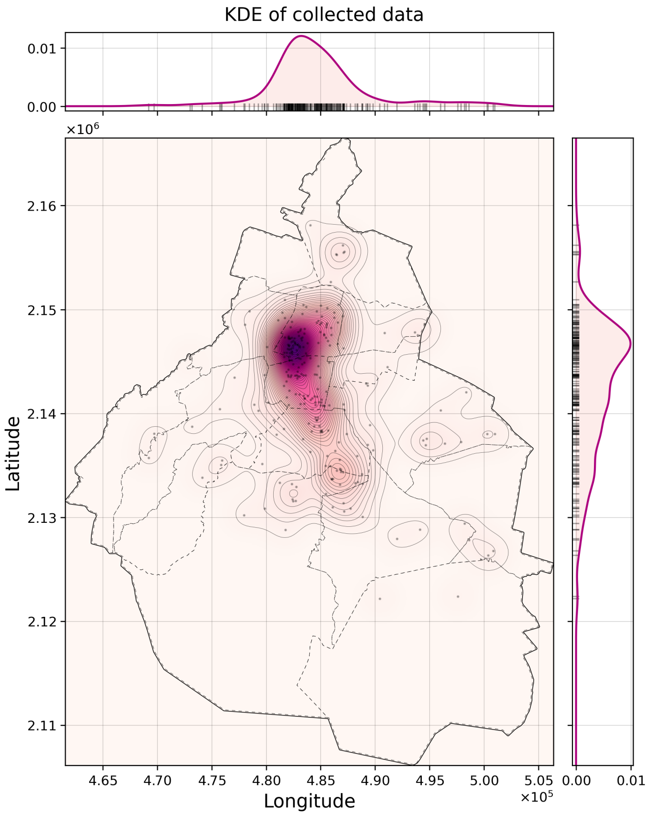


Figure 7. Kernel Density Estimation of buildings damaged by the 19 September 2017 earthquake.

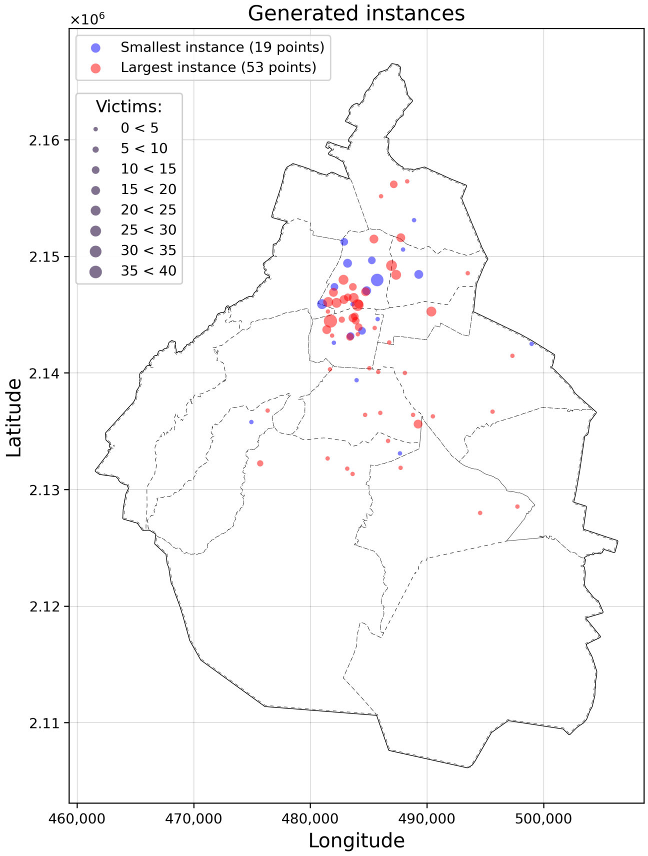


Figure 8. Instances with the lowest and highest number of demand points described in Figure 7.

4.3. Data Obtained: City Traffic

Table 7 describes the parameters used to capture traffic information for most CDMX streets by applying the procedure described in the methodology section. Furthermore, Figure 9 shows the tiles required to map the city; additionally, on the right a zoomed-in region on the Cuauhtémoc district is displayed. Finally, to obtain the traffic for each edge of the graph, Figure 10 shows the different stages required.

Table 7. List of parameters used in TomTom API.

Parameter	Value
Zoom level	16
Total number of tiles (to cover the bounding box)	79 × 106 = 8374
Tiles occupied (intersecting the polygon of the CDMX)	4789
Meters per tile	611.4962
Meters per pixel	1.1943

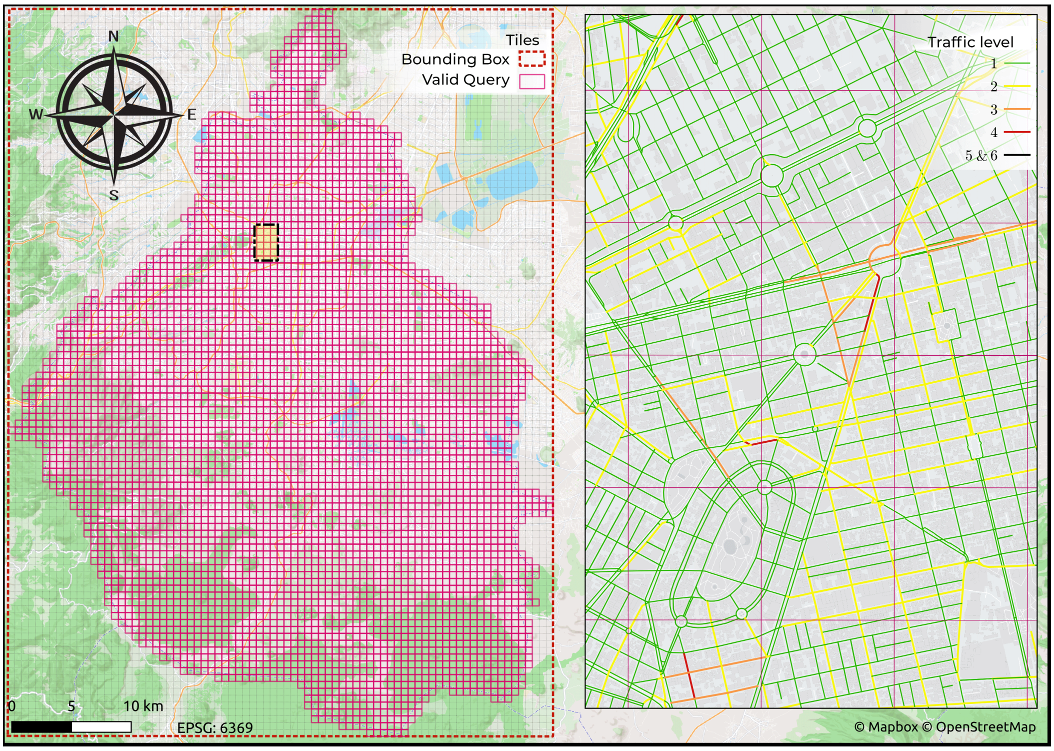
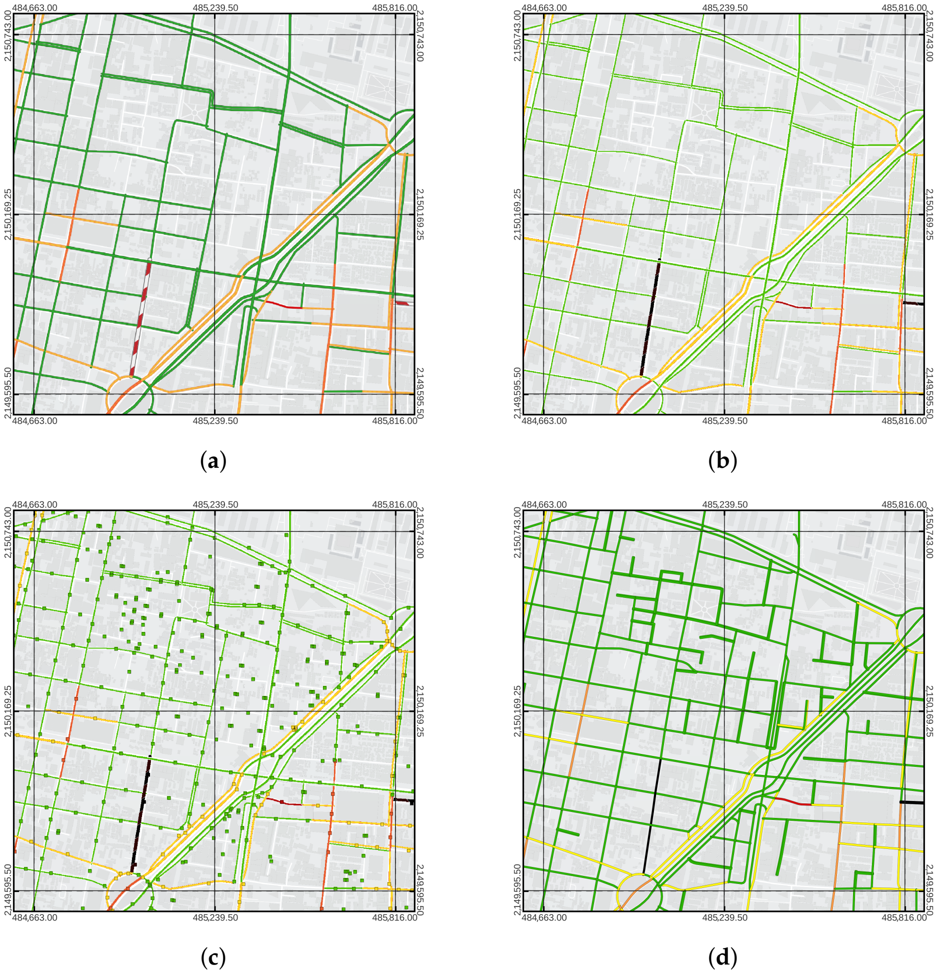


Figure 9. Tiles used to determine the amount of traffic in Mexico City. Example of data obtained on 14 November 2023 at 13:45 h.



(**a**) (**b**)

(**c**) (**d**)

Figure 10. Stages for obtaining traffic data using the TomTom API and the OSMNX network. (**a**) RGBA raster obtained through the TomTom API. (**b**) Conversion to single-band raster. (**c**) Sampling windows of the single-band raster. (**d**) Road network graph with traffic as attribute.

4.4. Parameter Tuning

Once the instance generation process is finished, execution of irace is carried out with the implementation of the NSGA-II algorithm [43,49,50]. The execution of the algorithm is characterized by the features listed in Table 8.

Table 8. Values used for parameter tuning.

Parameter	Value
Number of iterations	10
Performed experiments	4982
Remaining budget	18
Number of elites	5
Number of settings used	679
Total user time on CPU	880,930.4 (10.19 days)

The five elite configurations with the highest average performance are listed in performance order in Table 9. There are parameter values that are repeated in most cases, including the uniform crossover operator (4/5), the emergency assignment method (5/5), the repair with mutation (5/5), the population of 200 with 50% offspring (5/5), the crossover percentage of 0.5 (4/5), and the mutation percentage of 0.5 (4/5). The experiments carried out are shown in Figure 11, which were obtained using the Python code provided in [51]. This figure shows that in the first iterations, parameters are sampled that produce regular configurations. These configurations do not converge near the minimum value found (marked with ×); however, as iterations progress, the elite configurations start to appear in the graph (♢,⋆), and most of the time they are found at values closest to the known minimum. In addition, instances that were aborted because they exceeded the execution times of 6 min (they return the value 1) are shown at the top, and these instances have been registered in a nonfeasible configuration file. In total, 164 iterations were interrupted.

Table 9. Elite configurations.

top	id	-cr	-mu	-a	-rep	-pop	-dec	-prob_cr	-prob_mu	-prob_ini
1	625	un	bf	ur	rm	200	50	0.5	0.25	0.1
2	499	un	bf	ur	rm	200	50	0.75	0.5	0.25
3	432	hun	bfo	ur	rm	200	50	0.5	0.5	0.25
4	655	un	bf	ur	rm	200	50	0.5	0.5	0.1
5	617	un	bfo	ur	rm	200	50	0.5	0.5	0.1

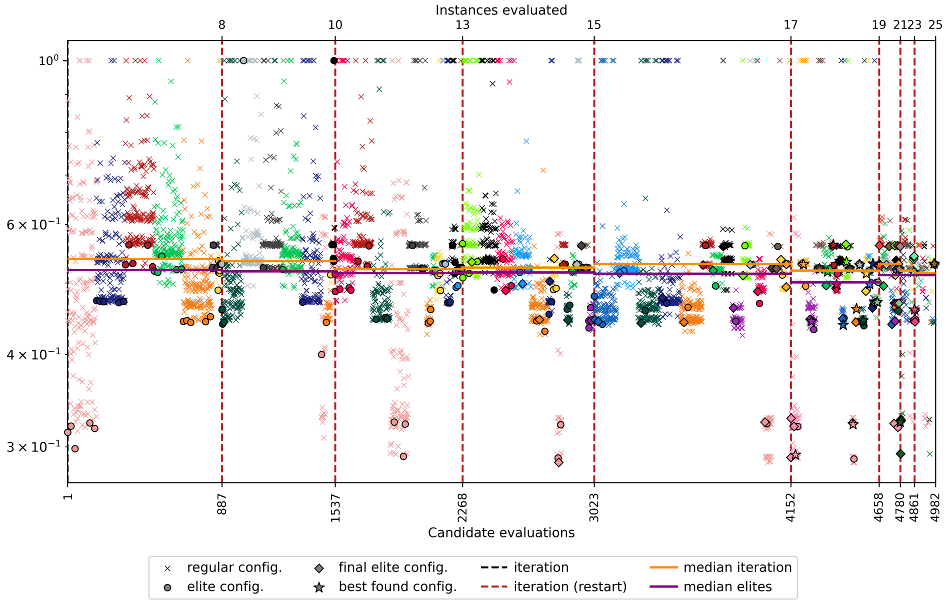


Figure 11. Experiments performed during parameter tuning. Each color represents a specific instance.

4.5. Solution Visualization

The result of the optimization process for each instance is a preview visualization of all Pareto front solutions. Considering the 2017 earthquake post-disaster scenario solutions described in Figure 12, and the relevant data for each solution of this scenario in Table 10, Figure 13 contains the preview of three solutions: 1, 8 and 14. Figure 14 shows the HTML file generated when processing solution number 8 (the knee point of the Pareto front in the Figure 12). It is possible to interact with this generated map to visualize more information about the elements found.

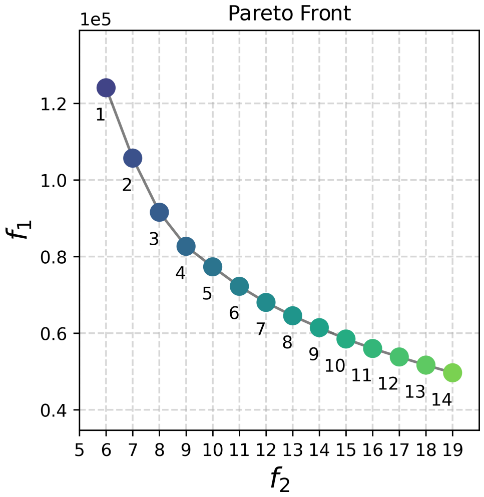


Figure 12. Pareto front of 2017 earthquake post-disaster scenario.

Table 10. Statistics of the Pareto front solutions of the 2017 scenario.

Solution	f_1	f_2	r_{max}	r_{prom}	*	b_{min}	b_{max}
1	12,4065.2	6	7904.6	3264.9	1	2	16
2	10,5696.9	7	7800.3	2781.5	1	2	13
3	91,577.8	8	6107.8	2409.9	1	2	8
4	82,685.1	9	6258.1	2175.9	1	2	7
5	77,329.4	10	6258.1	2035.0	2	2	7
6	72,245.0	11	5013.1	1901.2	4	3	7
7	68,039.6	12	4907.8	1790.5	5	3	7
8	64,555.3	13	4907.8	1698.8	6	2	7
9	61,456.4	14	4907.8	1617.3	7	2	7
10	58,493.7	15	4907.8	1539.3	7	2	7
11	56,013.6	16	4907.8	1474.0	8	2	7
12	53,807.9	17	4907.8	1416.0	8	2	5
13	51,680.9	18	4907.8	1360.0	9	2	5
14	49,675.7	19	4907.8	1307.3	10	2	5

* Number of activated bases that are assigned to only one demand point.

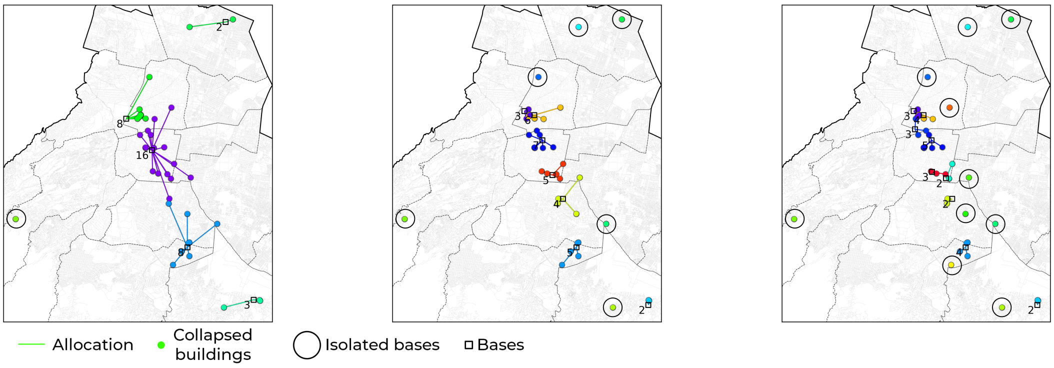
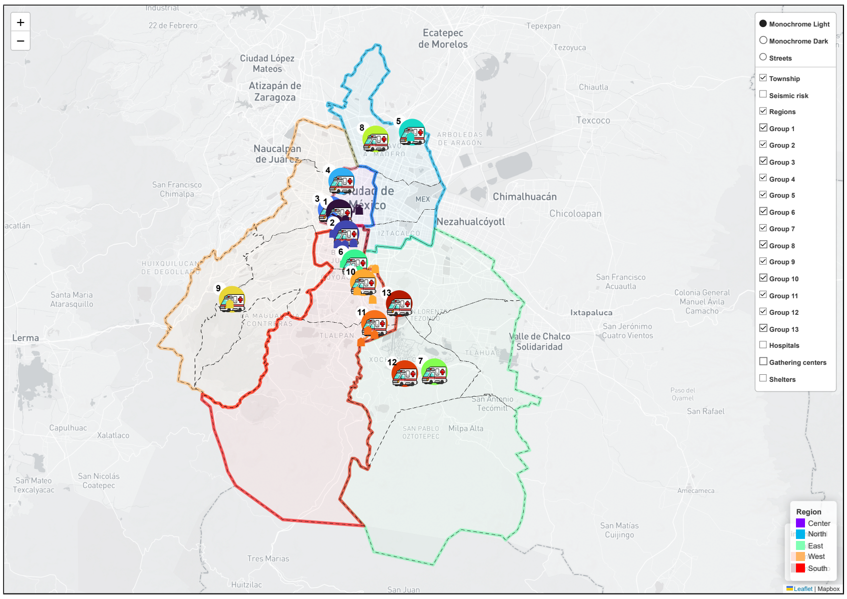


Figure 13. Preview of the solutions 1, 8, and 14 for the 2017 scenario.



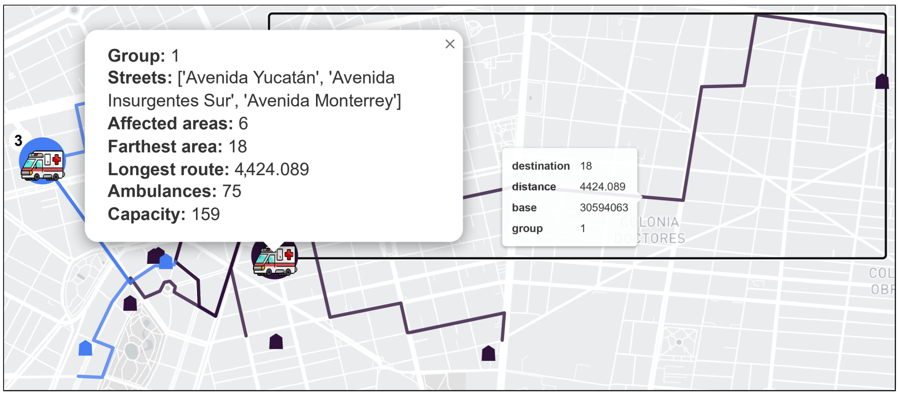
● Monochrome Light
○ Monochrome Dark
○ Streets
☑ Township
□ Seismic risk
☑ Regions
☑ Group 1
☑ Group 2
☑ Group 3
☑ Group 4
☑ Group 5
☑ Group 6
☑ Group 7
☑ Group 8
☑ Group 9
☑ Group 10
☑ Group 11
☑ Group 12
☑ Group 13
□ Hospitals
□ Gathering centers
□ Shelters

Region
🟪 Center
🟦 North
🟩 East
🟧 West
🟥 South

Figure 14. Interactive visualization of the solution, displaying the available layers and the default map view.

4.6. Implementation Plan

An example of the Attention Flow described in Section 3.8 is presented. Clicking on the marker of an activated base will display the group number, the locations where ambulances should be assigned, the number of locations to be serviced, the most distant focus of attention, the number of ambulances to be deployed, and the total capacity of the activated base. Similarly, clicking on a route produces geometry information and a message indicating the group to which the route belongs, the focus of attention associated with the base, and the base's ID in the transport network. These elements are shown in Figure 15.

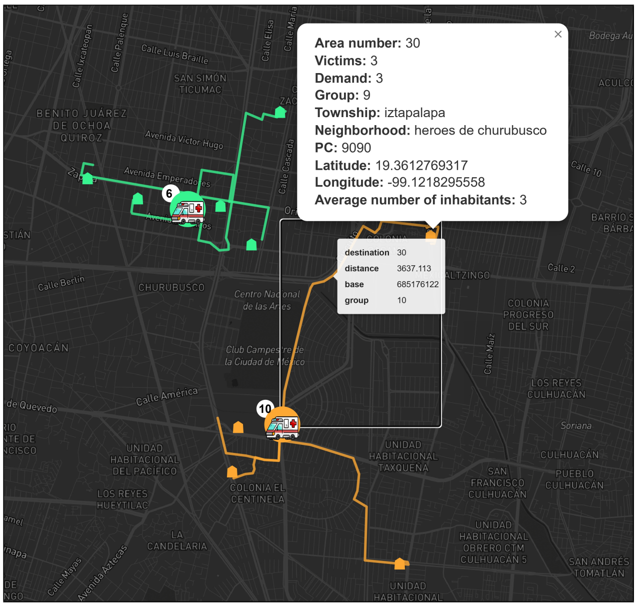


Group: 1
Streets: ['Avenida Yucatán', 'Avenida Insurgentes Sur', 'Avenida Monterrey']
Affected areas: 6
Farthest area: 18
Longest route: 4,424.089
Ambulances: 75
Capacity: 159

destination	18
distance	4424.089
base	30594063
group	1

Figure 15. Elements of an activated base and a route. Only two specific groups are shown.

An *X* symbol of the color assigned to the group is displayed next to the demand points. Clicking on one of these points displays the number of the demand point, the number of victims, the estimated demand, the group to which it belongs, the address, the coordinates, and the average number of residents in the block, as shown in the Figure 16. Finally, other points of interest can be shown, such as the publication of hospitals, gathering centers, and temporary shelters that can be installed, related to geographic information obtained from official data.



Area number: 30
Victims: 3
Demand: 3
Group: 9
Township: iztapalapa
Neighborhood: heroes de churubusco
PC: 9090
Latitude: 19.3612769317
Longitude: -99.1218295558
Average number of inhabitants: 3

destination	30
distance	3637.113
base	685176122
group	10

Figure 16. Elements of a demand point. Possible regions of attention are displayed.

4.7. Runtime Results

The creation of an instance from the location of the demand points and the traffic information requires an execution time that needs to be analyzed; therefore, 15 executions of the complete procedure were carried out, obtaining the data described in Table 11.

Table 11. Runtime (in seconds) of the three main stages of data integration, θ: instance, avg: mean, std: standard deviation.

			Reading		Processing		Dijkstra		
θ	$\lvert I \rvert$	$\lvert J \rvert$	Avg	Std	Avg	Std	Avg	Std	Total
0	1527	43	23.935	1.039	32.511	1.689	30.361	1.461	86.806
1	1364	45	24.452	1.208	32.973	1.449	30.775	1.199	88.2
2	1581	44	24.225	0.988	32.070	1.204	26.961	0.965	83.256
3	1619	38	23.358	0.235	31.373	0.569	22.408	0.137	77.139
4	1606	19	23.268	0.235	31.126	0.288	10.737	0.040	65.131
5	1611	53	22.983	0.178	30.961	0.361	30.305	0.165	84.249
6	1540	23	23.120	0.185	30.616	0.430	14.456	0.075	68.192
7	1579	37	22.946	0.200	30.621	0.434	22.071	0.110	75.638
8	1566	35	22.880	0.134	30.754	0.454	21.503	0.072	75.137
9	1707	26	22.869	0.182	30.295	0.438	14.182	0.053	67.345

Figure 17 depicts the comparison between the computation of the routes by Dijkstra's algorithm with the implementation of NetworkX and the implementation of cuGraph. In this figure, the processing time is proportional to the number of demand points, and the performance with NetworkX is almost three times slower compared to cuGraph for all instances. Therefore, the final implementation uses the cuGraph library.

The rest of the execution time to create the instances corresponds to the reading of data and the execution of procedures without taking into account the path calculation. Figure 18 shows the percentage of time required for each stage in the total runtime, where the reading is between 20 and 25 s and the processing takes about 30 and 33 s, which does not represent a significant difference between the different instances, as in the case of routes.

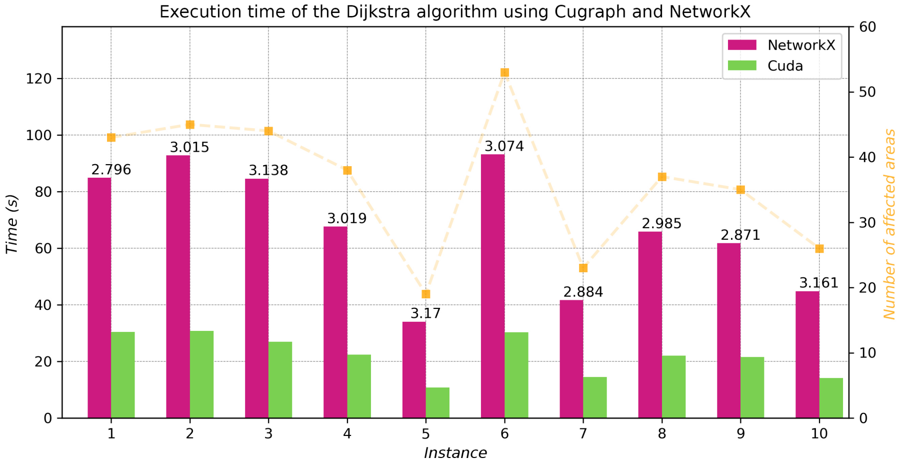


Figure 17. Comparison of the execution time of Dijkstra's algorithm with NetworkX and Cugraph, ratio $t_{NetworkX}/t_{CuGraph}$.

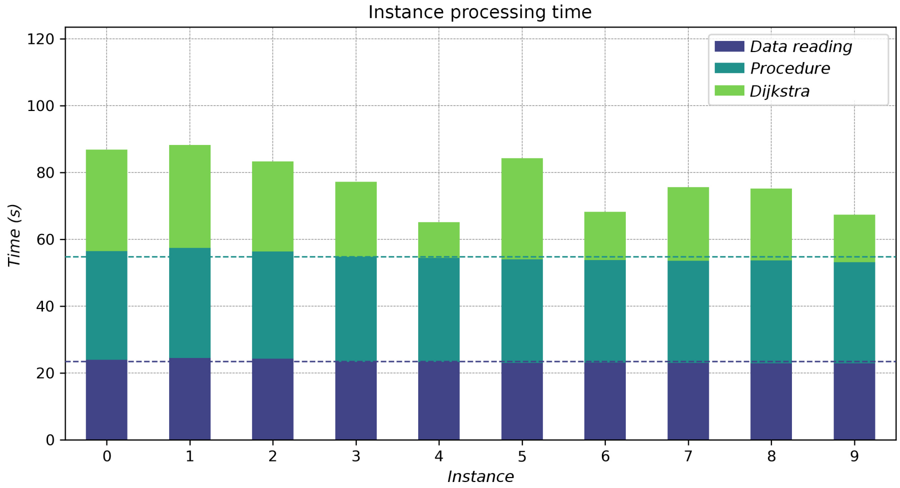


Figure 18. Ratio of runtime to generate the instances between data reading, computation of the routes, and the remaining procedure.

5. Conclusions and Future Work

Relevant data can be obtained from official data sources in the study area to solve the location-allocation problem of ambulances in Mass Casualty Incidents with multiple demand points, integrating historical data of disaster incidents; the location of complementary resources, such as hospitals, temporary shelters, and collection centers; and the geographical delimitation of regions of care designated by the authorities of emergency services.

The proposed methodology to determine candidate emergency response areas (candidate bases) allows us to find a subset of 1846 representative vertices on the graph that model the road transport network, from which at least 99% of the remaining vertices are expected to reach within 9 min at 40 km/h, considering different daytime values. In contrast to the alternative of using the complete road transport network in the problem model, the resulting search space contains only 1.43% of the original graph. Although the identification of candidate bases is performed statically, it is possible to discard more bases considering complementary event data, such as traffic data.

By obtaining the set of roads that can be used as temporary ambulance bases with their respective capacities, the location of the demand points with their respective estimated number of victims, and the graph that models the road transport network, it is possible to analyze the location-allocation problem as a multi-objective optimization problem. This model requires the identification of the candidate bases to be activated (location) and the demand point served by each one (allocation), subject to eight constraints that ensure the congruence of the solution and limit the solutions to viable regions, considering the maximum response time or the maximum number of candidate bases to be activated.

By using the Kernel Density Estimation of the historical data of buildings damaged by the 19 September 2017 earthquake, it is possible to generate hypothetical scenarios of the problem. For each scenario (hypothetical or real), a specific instance of the same problem was generated. To generate such an instance, the read time of the data and the procedure have approximately the same duration, but the time needed to compute the cost matrix is proportional to the number of demand points. The Dijkstra algorithm was used to obtain the cost matrix by inverting the graph and running it twice for each demand point, using as the origin the end points of the edge that is closest to the demand points. Implementing this algorithm using graphs from the cuGraph library accelerated this calculation by approximately 300% compared to the implementation of the NetworkX library. The NSGA-II algorithm, implemented with the proposed parameters, allows us to find solutions that adapt to the two objectives considered and to the multiple constraints established. The elite configurations found in the parameter tuning stage for the NSGA-II algorithm have a better evaluation with the hypervolume indicator and have a tendency to converge on Pareto fronts that are close to the best known optimal zone compared to other regular configurations. The obtained parameters of the elite configurations suggest that it is preferable to have large population sizes (200 individuals), to use the urgency-based allocation method, to perform mutation repair, and to have a small initialization probability (0.1–0.25). Regarding time-interrupted configurations, it can be observed that increasing the percentage of descendants significantly increases execution time, so it is preferable to keep it in the middle (0.5) to avoid bottlenecks. However, the best option to compare these configurations would be to implement the time-based termination criterion instead of aborting when the execution time is exceeded. The urgency-based allocation method was considered one of the best configurations compared to the classic and the demand–urgency-based methods, suggesting that the relationship between base capacity and urgency at the demand point in this problem is not significant enough to benefit from a method that considers demand first and urgency afterward.

The possible parameter values included populations of small size (20, 50) and a percentage of descendants. On the other hand, the mutation repair method outperformed the normal repair method, indicating that the procedure used to discard already explored solutions tends to improve the performance of the algorithm. A more rigorous analysis is needed to verify the relationship between these parameters and the algorithm performance, but these results provide a baseline for designing the required experiments.

The evaluation of the elite configurations on three representative instances with 20 random seeds suggests that although these configurations stand out from other regular configurations evaluated in parameter tuning, better techniques and configurations still need to be evaluated to obtain more accurate results with less variability, exploring solutions on the Pareto front with a smaller number of activated bases. However, the quality of the solutions is improved by merging the Pareto fronts of multiple runs of the same configuration, with the disadvantage of requiring more time. Furthermore, by parallelizing the multiple runs to generate the unified Pareto front, the result was obtained in half the time of a sequential execution.

The different solutions of the Pareto front suggest different ways of assigning the ambulances to the focus locations, and the pre-visualizations allow for the intuitive identification of which points remain isolated and which bases should be activated. The final visualization, generated in HTML format, can be used for further implementation in a context that allows the emergency medical services involved in the care plan to visualize the information interactively, to show or hide elements of interest depending on the circumstances.

The comparison of the results with other algorithms is important in future work. However, in this paper, the main objective focuses on the mathematical modeling of the problem and the integration of real data that could be obtained during a large-scale emergency scenario, from which the quality of solutions can be described. Thus, this work represents a useful starting point for future studies.

Author Contributions: Conceptualization, M.S.-P. and G.G.; methodology, M.M.-P. and M.S.-P.; software, V.K.L.-S. and G.G.; validation, G.G. and M.S.-P.; formal analysis, M.M.-P. and V.K.L.-S.; investigation, M.M.-P. and V.K.L.-S.; resources, V.K.L.-S. and M.S.-P.; data curation, G.G. and M.M.-P.; writing—original draft preparation, V.K.L.-S. and M.M.-P.; writing—review and editing, G.G. and M.M.-P.; visualization, M.M.-P. and V.K.L.-S.; supervision, G.G. and M.M.-P.; project administration, M.S.-P. and G.G.; funding acquisition, M.S.-P. and G.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially sponsored by the Instituto Politécnico Nacional under grants 20240426, 20240632, and by the Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) under grants 960525 and 1183927.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in NextCloud at https://sadami.ddns.net/nextcloud/s/fLogjaH6CqNxHp3?path=%2F (accessed on 18 April 2024).

Acknowledgments: We are thankful to the reviewers for their time and invaluable and constructive feedback, which helped improve the quality of the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. The International Federation of Red Cross and Red Crescent Societies. What Is a Disaster? Available online: https://www.ifrc.org/what-disaster (accessed on 4 November 2022).
2. World Health Organization. Emergency Cycle. Available online: https://www.who.int/europe/emergencies/emergency-cycle (accessed on 4 November 2022).
3. American college of surgeons committee on trauma and Campos, Víctor and Roig, Félix Garcia. In *PHTLS Soporte Vital de Trauma Preshospitalario*; Jones & Bartlett Learning: Burlington, MA, USA, 2020.
4. International Resources Group and "Oficina de los Estados Unidos de Asistencia para Desastres en el Extranjero para Latino América y el Caribe". Curso Básico Sistema de Comando de Incidentes. 2021. Available online: https://www.gob.mx/cms/uploads/attachment/file/228836/Curso_Basico_SCI_material_de_referencia.pdf (accessed on 14 September 2022).
5. Halper, R.; Raghavan, S.; Sahin, M. Local search heuristics for the mobile facility location problem. *Comput. Oper. Res.* **2015**, *62*, 210–223. [CrossRef]
6. Barojas-Payán, E.; Sánchez-Partida, D.; Gibaja-Romero, D.E.; Martínez-Flores, J.L.; Cabrera-Rios, M. Optimization Model to Locate Pre-positioned Warehouses and Establish Humanitarian Aid Inventory Levels. In *Disaster Risk Reduction in Mexico: Methodologies, Case Studies, and Prospective Views*; Springer International Publishing: Cham, Switzerland, 2021; pp. 169–192. [CrossRef]
7. Zaffar, M.A.; Rajagopalan, H.K.; Saydam, C.; Mayorga, M.; Sharer, E. Coverage, survivability or response time: A comparative study of performance statistics used in ambulance location models via simulation–optimization. *Oper. Res. Health Care* **2016**, *11*, 1–12. [CrossRef]
8. Mohri, S.S.; Haghshenas, H. An ambulance location problem for covering inherently rare and random road crashes. *Comput. Ind. Eng.* **2021**, *151*, 106937. [CrossRef]
9. Toregas, C.; Swain, R.; ReVelle, C.; Bergman, L. The Location of Emergency Service Facilities. *Oper. Res.* **1971**, *19*, 1363–1373. [CrossRef]
10. Kaveh, M.; Mesgari, M.S. Improved biogeography-based optimization using migration process adjustment: An approach for location-allocation of ambulances. *Comput. Ind. Eng.* **2019**, *135*, 800–813. [CrossRef]
11. Hashtarkhani, S.; Matthews, S.A.; Yin, P.; Mohammadi, A.; Mohammad Ebrahimi, S.; Tara, M.; Kiani, B. Where to place emergency ambulance vehicles: Use of a capacitated maximum covering location model with real call data. *Geospat. Health* **2023**, *18*, 1198. [CrossRef]
12. Barojas-Payán, E.; Sánchez-Partida, D.; Caballero-Morales, S.O.; Martínez-Flores, J.L. A Hybrid Capacitated Multi-facility Location Model for Pre-positioning Warehouses and Inventories in Regions at Risk in Mexico. In *Disaster Risk Reduction in Mexico: Methodologies, Case Studies, and Prospective Views*; Springer International Publishing: Cham, Switzerland, 2021; pp. 193–221. [CrossRef]
13. Zheng, Y.J.; Chen, S.Y.; Ling, H.F. Evolutionary optimization for disaster relief operations: A survey. *Appl. Soft Comput.* **2015**, *27*, 553–566. [CrossRef]

14. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
15. Nolz, P.C.; Doerner, K.F.; Gutjahr, W.J.; Hartl, R.F. A Bi-objective Metaheuristic for Disaster Relief Operation Planning. In *Advances in Multi-Objective Nature Inspired Computing*; Coello Coello, C.A., Dhaenens, C., Jourdan, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 167–187. [CrossRef]
16. Chang, F.S.; Wu, J.S.; Lee, C.N.; Shen, H.C. Greedy-search-based multi-objective genetic algorithm for emergency logistics scheduling. *Expert Syst. Appl.* **2014**, *41*, 2947–2956. [CrossRef]
17. Schjølberg, M.E.; Bekkevold, N.P.; Sánchez-Díaz, X.; Mengshoel, O.J. Comparing Metaheuristic Optimization Algorithms for Ambulance Allocation: An Experimental Simulation Study. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Lisbon, Portugal, 15–19 July 2023; pp. 1454–1463. [CrossRef]
18. Saeidian, B.; Mesgari, M.S.; Ghodousi, M. Evaluation and comparison of Genetic Algorithm and Bees Algorithm for location–allocation of earthquake relief centers. *Int. J. Disaster Risk Reduct.* **2016**, *15*, 94–107. [CrossRef]
19. Geng, J.; Hou, H.; Geng, S. Optimization of Warehouse Location and Supplies Allocation for Emergency Rescue under Joint Government–Enterprise Cooperation Considering Disaster Victims' Distress Perception. *Sustainability* **2021**, *13*, 10560. [CrossRef]
20. Alinaghian, M.; Hejazi, S.R.; Bajoul, N.; Sadeghi Velni, K. A novel robust model for location-allocation of healthcare facilities considering pre-disaster and post-disaster characteristics. *Sci. Iran.* **2023**, *30*, 619–641. [CrossRef]
21. Oksuz, M.K.; Satoglu, S.I. Integrated optimization of facility location, casualty allocation and medical staff planning for post-disaster emergency response. *J. Humanit. Logist. Supply Chain. Manag.* 2023, *ahead-of-print*. [CrossRef]
22. Qi, W.; Ovur, S.E.; Li, Z.; Marzullo, A.; Song, R. Multi-Sensor Guided Hand Gesture Recognition for a Teleoperated Robot Using a Recurrent Neural Network. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6039–6045. [CrossRef]
23. Zhao, J.; Lv, Y.; Zeng, Q.; Wan, L. Online Policy Learning-Based Output-Feedback Optimal Control of Continuous-Time Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 652–656. [CrossRef]
24. Medina-Perez, M.; Legaria-Santiago, V.K.; Guzmán, G.; Saldana-Perez, M. Search Space Reduction in Road Networks for the Ambulance Location and Allocation Optimization Problems: A Real Case Study. In *Telematics and Computing*; Mata-Rivera, M.F., Zagal-Flores, R., Barria-Huidobro, C., Eds.; Springer: Cham, Switzerland, 2023; pp. 157–175.
25. Karpova, Y.; Villa, F.; Vallada, E.; Ángel Vecina, M. Heuristic algorithms based on the isochron analysis for dynamic relocation of medical emergency vehicles. *Expert Syst. Appl.* **2023**, *212*, 118773. [CrossRef]
26. Caglayan, N.; Satoglu, S.I. Multi-Objective Two-Stage Stochastic Programming Model for a Proposed Casualty Transportation System in Large-Scale Disasters: A Case Study. *Mathematics* **2021**, *9*, 316. [CrossRef]
27. Wang, J.; Wang, Y.; Yu, M. A multi-period ambulance location and allocation problem in the disaster. *J. Comb. Optim.* **2020**, *43*, 909–932. [CrossRef]
28. Nadar, R.A.; Jha, J.; Thakkar, J.J. Adaptive variable neighbourhood search approach for time-dependent joint location and dispatching problem in a multi-tier ambulance system. *Comput. Oper. Res.* **2023**, *159*, 106355. [CrossRef]
29. Daskin, M.S.; Maass, K.L. The p-Median Problem. In *Location Science*; Springer International Publishing: Cham, Switzerland, 2015; pp. 21–45. [CrossRef]
30. Ghoseiri, K.; Ghannadpour, S.F. Solving capacitated p-median problem using genetic algorithm. In Proceedings of the 2007 IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 2–4 December 2007. [CrossRef]
31. Instituto Nacional de Estadística y Geografía (INEGI). Límite de Alcaldías (áreas Geoestadísticas Municipales). 2023. Available online: https://datos.cdmx.gob.mx/dataset/alcaldias (accessed on 1 August 2023).
32. Secretaría de Gestión Integral de Riesgos y Protección Civil (SGIRPC). Capas Relacionadas al Sismo del 19/09/2017: Inmuebles Colapsados, Zona de Mayor Afectación e Infraestructura Dañada, Zonas de Riesgo Sísmico. Available online: https://www.atlas.cdmx.gob.mx/datosabiertos2.html (accessed on 1 August 2023).
33. Órgano de Difusión del Gobierno de la Ciudad de México. Gaceta Oficial de la Ciudad de México, No. 685 Bis. Protocolo del Plan de Emergencia Sísmica. 2021. Available online: https://data.consejeria.cdmx.gob.mx/portal_old/uploads/gacetas/18652b057da05daee4cf7a0784093fff.pdf (accessed on 26 March 2023).
34. Instituto de Planeación Democrática y Prospectiva. Hospitales Públicos y Privados en Operación de la ZMVM. 2023. Available online: https://datos.cdmx.gob.mx/dataset/hospitales-publicos-y-privados-en-operacion-de-la-zmvm (accessed on 1 August 2023).
35. Secretaría de Gestión Integral de Riesgos y Protección Civil. 22 Conjuntos de Datos de Secretaría de Gestión Integral de Riesgos y Protección Civil. 2023. Available online: https://datos.cdmx.gob.mx/organization/secretaria-de-gestion-integral-de-riesgos-y-proteccion-civil (accessed on 1 August 2023).
36. Secretaría de Gestión Integral de Riesgos y Protección Civil. Refugios Temporales. 2023. Available online: https://datos.cdmx.gob.mx/dataset/refugios (accessed on 1 August 2023).
37. Instituto Nacional de Estadística y Geografía (INEGI). Polígonos de Manzanas de la Ciudad de México. 2023. Available online: https://datos.cdmx.gob.mx/dataset/poligonos-de-manzanas-de-la-ciudad-de-mexico (accessed on 1 August 2023).
38. Secretaría de Desarrollo Urbano y Vivienda (SEDUVI). Descarga de Datos SEDUVI. 2023. Available online: https://sig.cdmx.gob.mx/datos/descarga#d_datos_seduvi (accessed on 1 September 2023).
39. Secretaría de Administración y Finanzas (SAF). Descarga de Datos del Catastro. 2023. Available online: https://sig.cdmx.gob.mx/datos/descarga#d_datos_cat (accessed on 1 September 2023).

40. Secretaria de Gestión Integral de Riesgos y Protección Civil de la Ciudad de México and Urzúa, Myriam. *Protocolo del Plan de Emergencia Sísmica*; Gaceta oficial de la Ciudad de México: Mexico City, Mexico, 2021.
41. TomTom Developers. Zoom Levels and Tile Grid. 2023. Available online: https://developer.tomtom.com/map-display-api/documentation/zoom-levels-and-tile-grid (accessed on 9 September 2023).
42. Liang, J.J.; Yue, C.T.; Qu, B.Y. Multimodal multi-objective optimization: A preliminary study. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 2454–2461. [CrossRef]
43. Ma, H.; Zhang, Y.; Sun, S.; Liu, T.; Shan, Y. A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artif. Intell. Rev.* **2023**, *56*, 15217–15270. [CrossRef]
44. Ahmadi, M.H.; Ahmadi, M.A. Multi objective optimization of performance of three-heat-source irreversible refrigerators based algorithm NSGAII. *Renew. Sustain. Energy Rev.* **2016**, *60*, 784–794. [CrossRef]
45. pymoo: Multi-Objective Optimization in Python. Available online: https://www.pymoo.org (accessed on 2 September 2023).
46. López-Ibáñez, M.; Dubois-Lacoste, J.; Pérez Cáceres, L.; Birattari, M.; Stützle, T. The Irace Package: User Guide. 2022. Available online: https://cran.r-project.org/web/packages/irace/vignettes/irace-package.pdf (accessed on 21 October 2023).
47. Fitch, J. Response times: Myths, measurement & management. *JEMS J. Emerg. Med Serv.* **2005**, *30*, 46–56. [CrossRef]
48. Wu, C.; Hwang, K.P. Using a Discrete-event Simulation to Balance Ambulance Availability and Demand in Static Deployment Systems. *Acad. Emerg. Med.* **2009**, *16*, 1359–1366. [CrossRef] [PubMed]
49. Tian, Z.; Zhang, Z.; Ning, C.; Peng, T.; Guo, Y.; Cao, Z. Multi-objective optimization of cable force of arch bridge constructed by cable-stayed cantilever cast-in-situ method based on improved NSGA-II. *Structures* **2024**, *59*, 105782. [CrossRef]
50. Bora, T.C.; Mariani, V.C.; dos Santos Coelho, L. Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm. *Appl. Therm. Eng.* **2019**, *146*, 688–700. [CrossRef]
51. de Souza, M.; Ritt, M.; López-Ibáñez, M.; Cáceres, L.P. ACVIZ: A Tool for the Visual Analysis of the Configuration of Algorithms with irace. *Oper. Res. Perspect.* **2021**, *8*, 100–186. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.