EMERGING TECH CONFERENCE – Edge Intelligence

Volume 01, 2022, Page 1 – 10

Proceedings of Emerging Tech Conference:
Edge Intelligence 2022

Finding optimal ambulance parking spots in an urban area using artificial intelligence

Myrto Petta, Vasileios Stylianakis, Panagiotis Perivolaris*

University of Patras, Rio Patras, 26504, Greece

p.perivolaris@upatras.gr

Abstract

Over time, the protection and health services of each country face increasing challenges in sending mobile units to the scene of an emergency. In civil protection emergencies, immediate coverage of incidents in multiple locations is required, creating a complex problem as the number of mobile units is finite. However, there is also a need to improve the existing operational system due to the population growth in large urban centres and the small number of mobile units available in small urban centres. This paper therefore attempts to approach a solution to this problem structured around the placement of mobile health units at selected locations within a city using artificial intelligence tools, specifically clustering, to identify possible ambulance parking spots and the simulated annealing algorithm in order to highlight the optimal solution. The criterion of the optimization is the detection of ambulance parking locations in order to minimise the overall average response time.

1. Introduction

Emergency medical services, more commonly known as EMS (Emergency Medical Services), is a comprehensive system that provides emergency medical care. Its goal is focused on the timely care of victims of sudden and life- threatening injuries or emergencies in order to prevent unnecessary mortality or long-term morbidity. [1]

Artificial intelligence has the power to improve service delivery across the entire spectrum of healthcare. In the context of ambulance operations there are two main areas in which it can significantly assist: in decision making and in the planning of vehicle movement. This study focuses on the latter as most ambulance stations are not properly located and are currently placed in or next to hospitals, which does not take into account the distribution of actual emergency requests.

This is a highly complex problem of a random nature shaped by demand, travel times, service times and the (un)availability of ambulances and staff. The demand factor is a separate subject of study that obeys the same rules as police forecasting and fire engine demand. Using a high resolution predictive system, dynamic deployment systems can drastically reduce response time. Travel time is also dynamic and proportional to traffic conditions. The variation in travel time on the network is caused by random changes in travel demand, poor driving and accidents occurring on the network, affecting the capacity of the network. The randomness of call traffic,

travel times and service times result in random unavailability of ambulance. [2] The accumulation of these multi-nature characteristics makes ambulance demand a non- linear dynamic system.

The first models proposed were simple integer linear programming formulations, but over time more realistic features were gradually introduced and solution techniques evolved. Each model has different objectives, constraints on coverage, the total number of ambulances, the number of ambulances available at a location, the types of ambulances and the assumption on ambulance availability. The fundamental categories of ambulance location and relocation models are deterministic, stochastic and dynamic models.

Larson in 1974 developed the first location-distribution model, known as the Hypercube model, to minimize response time in urban EMS. [3] Daskin and Dean later considered three basic models to solve the problem, including set coverage, maximum coverage, and P-median models.

Two initial models for the static ambulance deployment problem are the well-known LSCM and MCLP. Several extensions of both models have been proposed in the literature related to the considered problem. These models do not recognize the fact that in some cases vehicles of different types can be dispatched to the scene of an incident. For this reason, a number of deterministic models have been proposed to address the issue of multiple coverage, as is the tandem equipment allocation model, also known as TEAM. Deterministic models are used at the planning stage and ignore stochastic considerations regarding ambulance availability.

Stochastic models reflect the fact that ambulances act as servers in a queuing system and cannot always respond to a call. The basis of these models was the Maximum Expected Covering Location Problem formulation or MEXCLP (Daskin, 1983). An extension of this is the probabilistic TIMEXCLP model, and of note are the MALP I and MALP II models proposed in 1989.

Finally, dynamic models can be used to periodically update ambulance locations throughout the day. Tests have shown that they can work in practice, provided there are fast heuristics and sufficient computational power. [4]

In recent years there have been many studies trying to develop different methods for solving ambulance station location problems. Lu and Sheu in 2013 proposed a robust model for finding optimal locations for emergency dispatch stations among a set of candidate locations. They assumed that the travel time between stations and demand points was uncertain and tried to minimize it. An algorithm based on the simulated annealing algorithm was also proposed to solve the model. Zhang, Dong on the other hand used genetic algorithm combined with Ant Colony optimization algorithm to solve a multi-objective optimization model for large-scale networks under different scenarios. In 2020 Carvalho and Captivo applied a heuristic pilot method algorithm to solve the dispatch and location problems of EMS systems aiming at maximum coverage, while in 2021 Mohri and Haghshenas developed a mathematical programming model to investigate the maximum coverage edge location problem to find the location of a set of ambulances to meet relief requests caused by rare and random traffic collisions. [5]

We can therefore see that over the last 50 years, the literature on ambulance positioning systems has grown significantly, reflecting the rapid technological and computational advances. However, so far in Greece, the study of this topic is extremely limited and consequently ambulance stations are mainly placed inside or next to hospitals, resulting in response times that are often prohibitive. This paper is a first approach to the issue with the ultimate goal of improving the ambulance location system in the Greek territory and is structured as follows: the second chapter presents the theoretical background of the computational tools used in the paper and

the third chapter presents the model we composed to represent the problem and the solution procedure. In the fourth we present the results and in the fifth chapter we draw conclusions and make suggestions for future research.

2. Background

This paper is structured by two main computational tools that fall within the field of artificial intelligence: clustering and the Simulated Annealing algorithm.

2.1 Clustering

In general, the first step in solving an ambulance demand or location problem is that the area under consideration is divided into small zones. In related work these zones are either the result of GIS (Geographic Information Systems) or are divided in a static way as a grid. This paper proposes a dynamic way of dividing the area through clustering of ambulance call history data in order to highlight high-level causality relationships related either to the number of possible accidents or to the probability that something unfortunate will happen and an ambulance will be needed in a given cluster during a given time interval.

Clustering is the arrangement of a collection of data into clusters based on some measure of similarity. Elements belonging to the same cluster show greater similarity than elements belonging to different clusters. It is a type of unsupervised learning method where it is used as a process to find meaningful structure, explanatory underlying processes, genetic features and clustering inherent in a set of examples.

Cluster analysis is performed using the K-means algorithm (MacQueen 1967). Its purpose is simple: to cluster similar data points and discover underlying patterns. To achieve this it looks for a fixed number (k) of clusters in a dataset. The number k refers to the number of clusters needed in the dataset. The K-means algorithm identifies the number k of centroids and then allocates each data point to the nearest cluster while keeping the centroids as small as possible. [6]

To determine the optimal value of k, the Elbow method is often used. This method involves examining a graph showing that by adding centroids the range of data around those centroids decreases. Usually, inertia, i.e. the sum of the squared distances of the samples from the nearest cluster center is used. To determine the optimal number of clusters, we need to choose the value of k at the "elbow", i.e. the point after which inertia starts to decrease linearly.

The K-means algorithm consists of three main steps. In the first step, the initial centroids are selected, with the most basic method being to select samples from the dataset. After initialization, K-means consists of repeating the other two steps. The second step assigns each sample to the nearest centroid. The third step creates new centroids by taking the average of all samples assigned to each previous centroid. The difference between the old and new centroids is calculated and the algorithm repeats these last two steps until this value is less than a threshold. The algorithm converges to a point where the centroids cannot move any further. It stops creating and optimizing clusters when the centroids have stabilized and thus there is no change in their values because the clustering was successful or when the specified number of iterations has been reached.

In the context of this work, Scikit-learn (sklearn), a popular Python machine learning module based equally on Matplotlib, SciPy and NumPy, is exploited. Data clustering is achieved using the

sklearn.cluster module. The K-Means algorithm is implemented with methods of sklearn.cluster.KMeans.

2.2 Simulated Annealing

Evolutionary algorithms are algorithms that use mechanisms inspired by nature by solving problems through processes that mimic the behavior of living organisms. They try to achieve optimal situations by successive improvements, which occur in a variation (mutation, recombination).

The simulated annealing algorithm belongs to the evolutionary algorithms and is a quite popular method in terms of solving optimization problems. It was introduced by Kirkpatrick et al. inspired by the metal annealing process. [7] In general, the algorithm adopts an iterative motion according to the temperature variable parameter that mimics the metal annealing transaction.

The novelty of this algorithm is that it solves this problem by accepting also "bad solutions" or more precisely by incorporating two iterative loops which are the cooling process for the annealing process and the Metropolis criterion. The basic idea behind the criterion is to execute it randomly in order to perform additional search in the neighborhood of the candidate solution to avoid getting trapped in local outliers. [8]

More specifically, the algorithm starts by defining the basic parameters, which are the initial temperature, the cooling system and the number of iterations. Based on the initial temperature, the initial energy, i.e. the value of the objective function, is calculated. Then, by slightly varying the variables of the objective function, a new solution of the objective function, i.e. a new energy value, is calculated and compared with the previous one.

If the solution found leads to a lower energy value, it is accepted for the calculations of the next cycle. If not, then it is evaluated using the Metropolis criterion. In physical annealing, the distribution of the energies of the molecules obeys Boltzmann's Law:

$$P(E) = e^{\frac{\Delta E}{kT}} \quad (1)$$

In the above relation, ΔE is the change in energy, k is the Boltzmann constant and T is the temperature. By analogy, the acceptance probability is calculated with k=1. The acceptance probability is calculated continuously and dynamically in such a way that in the initial generations it is sufficiently large and towards the end of the algorithm it is sufficiently small. At the same time a random number δ is generated based on the uniform distribution. If:

$$\delta \leq P(E) \quad (2)$$

i.e. if the Metropolis criterion is satisfied, then the "bad solution" is accepted and the process continues. Otherwise, it is rejected, and a new solution is considered.

The above process continues until the maximum number of cycles is reached. Then the current (last) energy value is the optimal solution of the problem.

Each iteration starts by reducing the temperature according to

$$T_{i+1} = \alpha \cdot T_i \quad (3)$$

where α is the temperature reduction factor set at the beginning. The above process continues until the maximum number of cycles is reached. Then the current (last) energy value is the optimal solution of the problem.

3. Problem Structure and Solution Procedure

The problem under consideration concerns a virtual urban area over a 2-hour period. During this time the call center receives 10 calls, while the area has one hospital and initially two ambulances at its disposal to serve the cases. It is then examined how the program responds with the addition of another ambulance. The goal in both cases is to identify optimal ambulance parking spots to minimize the total average response time.

3.1 Version 1

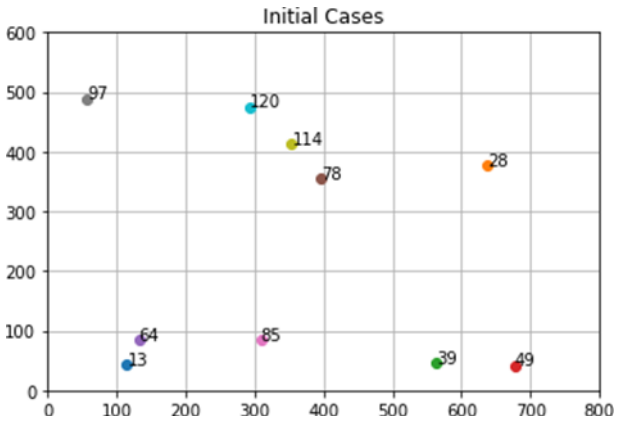

Figure 1: Ambulance call visualization for version 1

Then clustering, i.e. grouping the incidents to create the desired zones in a dynamic way, is done. Four clusters are created, the number of which was chosen arbitrarily. The centroid of each cluster is also a potential ambulance parking place.

Once the potential ambulance locations have been identified, it must be decided which ambulance will serve which incident so a decision function is needed. Each time a combination of two possible locations is tested. We assume that each ambulance serves the incidents that belong to its cluster and those that are closer to it than the other clusters that are not used.

Since the hospital has two ambulances at its disposal a decision must be made each time as to which ambulance will serve a call. We consider the facts:

- A: Ambulance A1 is closer to the incident
- B: Ambulance A1 is available
- C: Ambulance A2 is available

B, C are independent events. If an event is true it takes the value 1, otherwise it takes the value 0.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Table 1: State table

In column F, if the result is 0, ambulance A1 is chosen, and if the result is 1, ambulance A2 is chosen. Based on this

table, the decision function is:

$$F = \sim \mathrm{C} \,\&\, (\mathrm{A} \,\hat{}\, \mathrm{B}) | \,(\mathrm{A} \,\&\, \mathrm{B}) \tag{4}$$

The decision function indicates each time the ambulance that will take over the upcoming incident. Based on this, the distances L1 and L2 are calculated, i.e. the distance of the ambulance from the call location and the distance of the call from the hospital respectively. The function of the total response time for an incident is given by relation 5:

$$t = \frac{L1}{U} + \frac{L2}{U} \tag{5}$$

This is the total sum of the time it takes for the ambulance to get to the place of call and the time it takes to get to the hospital. For each combination of centroids, the average of the total response times of all incidents is finally compared. We then proceed to the objective function which is the core of the solution procedure. It combines all of the above for all ambulances and all incidents, calculates and returns the value of the average total response time which will essentially be the energy in the simulated annealing algorithm.

$$\frac{\sum_{i=0}^{N} t_i}{N} \tag{6}$$

At the beginning of the simulation each ambulance is at its parking position. We assume that the ambulance after delivering the patient to the hospital stays there for 3 minutes before returning to its position so that in case another call has arisen, it is considered available and if it is closer, it can take over. The 3-minute interval was chosen after testing.

To manage each incident we first check the condition of the ambulances by distinguishing cases. If an ambulance is on the hospital site at the given time, it is sent to the scene of the incident. Otherwise, the decision function is called.

Once the ambulance to take over the given incident is determined, the distances L1 (from the parking place to the incident), L2 (from the incident to the hospital) and S (total distance) as well as the total response time are calculated. These results are compiled in a table which is used to calculate the average of the total response time of all incidents at the end.

The last part of the program includes the simulated annealing algorithm, according to which all possible combinations of possible points are tested and the one that gives the lowest average total response time is selected. Therefore, this algorithm guarantees that the ambulance parking spots will be optimal in the defined frames. In each iteration, two centroids are randomly selected to feed the objective function. Because the SA algorithm works with small parameter changes for each iteration, one centroid is permuted at a time. The solution procedure starts with the centroids (463, 381.667) and (185.333, 70.333), and the basic parameters are set as follows:

- Maximum number of iterations = 1000
- Initial temperature = 10000.0
- Cooling system a = 0.98

The objective function is then called and its value, i.e. the total response time (relation 5) of the above combination is stored in a variable, which expresses the annealing energy.

In the first iteration the parameters are changed according to relation 3 and the new energy is calculated in the same way. To determine whether the solution is acceptable we examine the ΔE.

If ΔE < 0, then the solution is acceptable and we continue with a new temperature, and if ΔE > 0, we examine whether the metropolis criterion is satisfied.

If it is satisfied, then we accept the "bad solution" and continue the procedure. Otherwise, we reject it and proceed to a new solution. At the end of the iteration loop, the termination criteria related to the number of iterations are examined. If they are not met, the temperature and the counter are updated appropriately.

3.2 Version 2

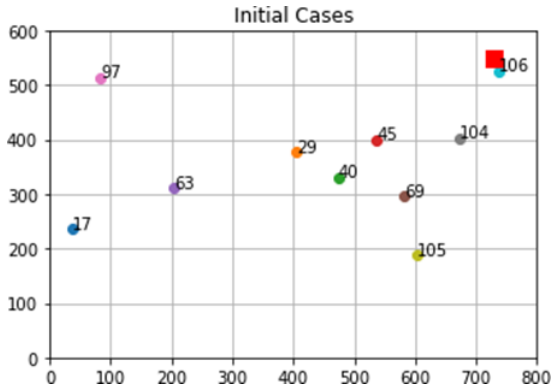

Figure 2: Ambulance call visualization for version 2

Considering that version 1 is a recorded history of ambulance calls in an urban area, the new sub-regions for scenario 2 are created based on the above data in order to highlight high causality relationships in terms of demand. To create them, clustering is again performed but with the difference that the Elbow Method is tested for the number of candidate parking spaces.

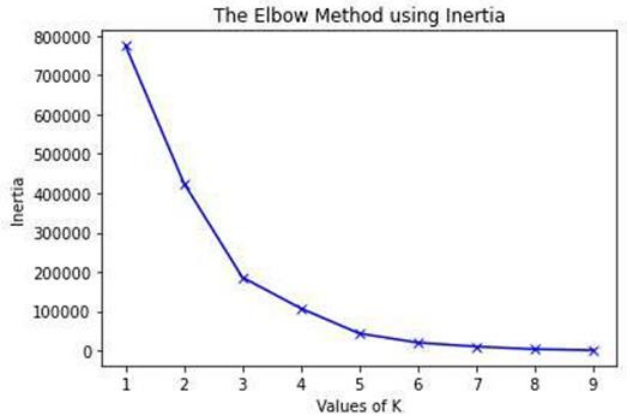

Figure 3: Elbow Method using Inertia

Since the hospital now has three ambulances at its disposal, the objective function must be modified to reflect the new conditions. For each combination of possible locations, each time the call centre receives a call, the distance of the incident from the considered parking places and from the hospital is calculated. These distances are ranked from shortest to longest and indicate the priority of ambulances in order to serve the incident. At the same time, the availability of the ambulances is examined, i.e. whether they are at their parking place, whether they have arrived at the hospital and when they are scheduled to leave the hospital and return to the centre. Having identified these two key parameters, the priority list is then examined.

If the closest location to the incident is the hospital, then it is checked if there is an ambulance there at that time, similar to version 1. If there is, then it takes over the incident.

Otherwise, the decision function per two ambulances is called. More specifically, after the function is called for the first two ambulances, for example for ambulances 1 and 2, and indicates ambulance 2, then it is called again for ambulances 2 and 3. Whichever one is indicated the second time is the one that will take over the incident. The reason for not creating a new decision function for three ambulances is to see whether the program is flexible to possible changes. In case all ambulances are unavailable the case is assigned to the ambulance that is closest according to the priority list.

4. Results

4.1 Version 1

As mentioned in the previous chapter we first create the call record and then clustering is done using the K-Means algorithm with K=4. The four clusters with their respective centroids are shown in the following figure. The labels of each cluster are depicted in a different color.

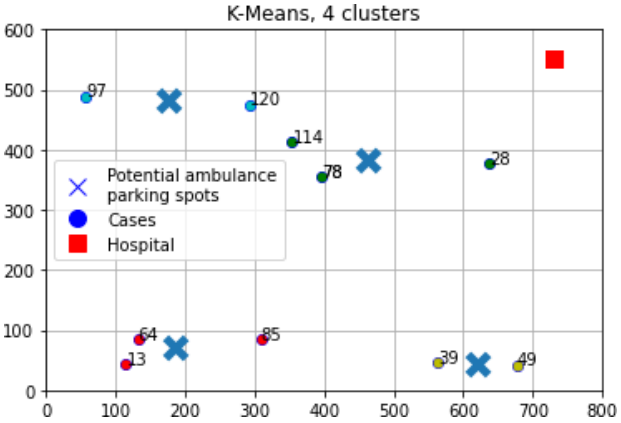

Figure 4: Clustering for K=4

Once the centroids have been determined, the program continues with the simulated annealing algorithm which results in the final parking spots (463, 381.667) and (185.333, 70.333) with a score of 14.797 and therefore an average total response time of 27.744 minutes.

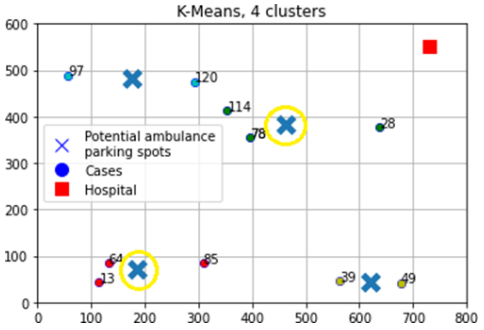

Figure 5: Optimal combination of parking spots for version 1

The distances L1, L2 and S correspond to the graph so to adapt them to real dimensions they have to be modified based on the 32:1 scale. Also, since the speed is set in kmh (specifically 50 kmh after a telephone call to the emergency services) the final time must be converted from hours to minutes. In the table the corresponding values are entered without the necessary conversions,

therefore the resulting response time is first calculated as a score without units of measurement and then converted to the actual response time.

4.2 Version 2

Using the version 1 call record the six possible ambulance parking spots for this case are:

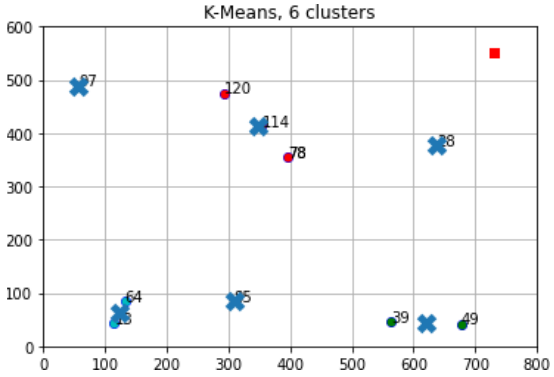

Figure 6: Clustering for K=6

So when combined with the candidate parking spaces the problem is summarized in the graph.

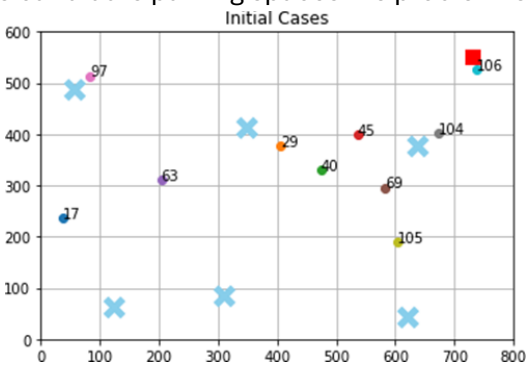

Figure 7: Visualization of calls and possible ambulance parking spots for version 2

Calling the optimization algorithm, the final parking spots for three ambulances are (57, 489), (638, 378) and (620, 43.5) with a score of 10,881 and an average total response time of 20,401 minutes.

While for the given problem with two ambulances, (175, 482) and (620, 43.5) are obtained with scores of 13,310 and an average total response time of 24,956 minutes.

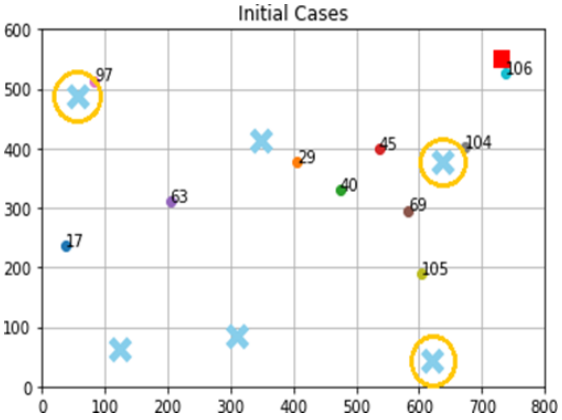

Figure 8: Optimal combination of parking spots for version 2

5. Conclusions

The problem of locating ambulances in an urban area is extremely complex because of the randomness that governs its nature. This paper attempts a first level analysis and solution finding through a simulation in a virtual urban area in Spyder environment (Python 3.8). All the elements of the simulation environment were selected in such a way that they correspond to reality. Version 1, which refers to the case of one hospital and two ambulances, is a real scenario: Argostoli, which is the capital of the prefecture of Kefalonia and Ithaca, has one hospital and two ambulances to meet the needs of its inhabitants.

As far as the results of the proposed solution are concerned, they seem to be quite satisfactory in both cases. The average total ambulance response time ranges from 20 to 30 minutes in a large urban area with limited resources. The results of the second edition highlight both the effectiveness of the program and the difference in the total average response time with the addition of an ambulance. The simulated annealing algorithm worked well, producing a solution in a short time, and was relatively easy to implement.

As the program is an initial approach to the given problem it can evolve in many different ways. A better approach would use a Geographical Information System (GIS). Also, the demand factor can be incorporated in another way: taking into account population density, while travel time can be made more realistic by taking into account traffic conditions at any given time. Regarding the better management of waiting ambulances, the use of queuing theory could be explored. Finally, the study of adding another hospital would be of great interest.

In Microsoft Word, pictures can be inserted into the document by going to Insert->Picture->From File... on the menu and selecting the desired file. To simplify working with the image, it is recommended that you insert the picture into a text box. In order to make it into a figure and add a caption, select the image by clicking on it and then go to Insert->Caption... (or Insert->Reference->Caption... in earlier versions of Microsoft Word). From here, you can select the position of the caption (this should be set to below the image) and edit the text within it. Make sure that “Figure” is selected in the “Label” drop-down list and click “OK” to generate it. Captions are numbered automatically in sequential order. Figure 1 is an example of a captioned image.

References

[1] Al-Shaqsi S. Models of international emergency medical service (EMS) systems. Oman Med J. 2010
[2] Papanikolaou N. , Optimal location of mobile health units in urban areas, Patras
[3] R.C. Larson, A hypercube queuing model for facility location and redistricting in urban emergency services, Computers & Operations Research, 1 (1) (1974), pp. 67-95
[4] Brotcorne L., Laporte G., Semet F., Ambulance location and relocation models, European Journal of Operational Research 147 (2003) 451–463
[5] Seyyed-Nader Shetab-Boushehri, Parisa Rajabi, Reza Mahmoudi, Modeling location–allocation of emergency medical service stations and ambulance routing problems considering the variability of events and recurrent traffic congestion: A real case study, 2022
[6] Dr. Michael J. Garbade, Understanding K-means Clustering in Machine Learning, 2018
[7] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, Optimization by Simulated Annealing, Science, 13 May 1983, Volume 220, Number 4598
[8] Yavuz Eren, Ibrahim Beklan Kucukdemiral, Ilker Ustoglu, in Optimization in Renewable Energy Systems, 2017