applied sciences

MDPI

Article

An Efficient Greedy Randomized Heuristic for the Maximum Coverage Facility Location Problem with Drones in Healthcare

Sumayah Al-Rabiaah *, Manar Hosny and Sarab AlMuhaideb

College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; mifawzi@ksu.edu.sa (M.H.); salmuhaideb@ksu.edu.sa (S.A.)
* Correspondence: salrabiaah@ksu.edu.sa

Abstract: Recently, drones, have been utilized in many real-life applications including healthcare services. For example, providing medical supplies, blood samples, and vaccines to people in remote areas or during emergencies. In this study, the maximum coverage facility location problem with drones (MCFLPD) was studied. The problem is the application of drones in the context of the facility location and routing. It involves selecting the locations of drone launching centers, which maximizes patient service coverage within certain drone range constraints. In this study, a heuristic named the maximum coverage greedy randomized heuristic (MCGRH) is developed. The idea of the algorithm is to first choose some facilities to open at random from among those that can handle the most weight of the patient demands. After that, patients are assigned to the closest opened facility with the capacity to serve them. Finally, drones are assigned to patients based on the least amount of battery consumed between the patient and the facility. Extensive testing of MCGRH indicated that it ranks efficiently alongside other methods in the literature that tried to solve the MCFLPD. It was able to achieve a high coverage of patients (more than 80% on average) within a very fast processing time (less than 1 s on average).

Keywords: maximum coverage; facility location problem; routing and scheduling; drones; combinatorial optimization; heuristics; metaheuristics; healthcare

check for updates

Citation: Al-Rabiaah, S.; Hosny, M.; AlMuhaideb, S. An Efficient Greedy Randomized Heuristic for the Maximum Coverage Facility Location Problem with Drones in Healthcare. *Appl. Sci.* **2022**, *12*, 1403. https://doi.org/10.3390/app12031403

Academic Editor: Samaneh Kouchaki

Received: 30 December 2021
Accepted: 25 January 2022
Published: 28 January 2022

1. Introduction

The provision of vital medical services to save patients in emergency and post-disaster situations is an essential outreach activity in healthcare. Recent years have witnessed an increase in natural disasters, including North American Wildfire Season in 2021 [1], the Haiti Earthquake and Tropical Storm Grace in 2021 [2], the Atlantic Hurricane Season in 2021 [3], the COVID-19 pandemic in 2020 [4], the bushfires in Australia in 2019 [5], Cyclone Idai in Africa in 2019 [6], and both an earthquake and tsunami in Indonesia in 2018 [7]. Natural disasters present obstacles to healthcare workers attempting to provide essential treatments to patients and victims, which may exacerbate the spread of epidemics and increase the number of fatalities. To rescue people in emergency and post-disaster situations, healthcare services must be quickly provided to injured individuals. Such rescues can be accomplished by assigning patients to the nearest medical provider, medical institution, or community center as well as by sending essential medical deliveries, such as defibrillators and blood supplies, to the injured person's location.

Most of the challenges related to accessing a patient's location are based on geographical characteristics such as dispersed islands, poor transportation infrastructure, and limited means of transport. These challenges tend to be particularly severe in both rural and developing countries. Moreover, when natural disasters occur, surrounding transportation infrastructure is often disrupted. For example, roads may become blocked because of natural disasters, or bridges may be broken [8].

To overcome these challenges and to provide necessary healthcare for patients in emergency and post-disaster situations, it is crucial to implement efficient methods to serve them. Ideally, transport means that it is not affected by damage to transportation infrastructure and that it is not significantly more expensive than ground transportation. Therefore, it is noteworthy that this can be achieved using unmanned aerial vehicles (UAVs), also known as drones.

Drones have proven successful in many applications in rugged environments, including agriculture, aerial photography, and data collection [9,10]. Drones have also been used in several healthcare applications (Figure 1). For example, drones have been used to spray chemicals in China to prevent the spread of COVID-19 [11]; to deliver medications during the Haitian earthquake of 2010 [12]; and to transfer blood samples [13], vaccines [14], and stool [15] to laboratories or wherever they are needed by patients.

Figure 1. Example of a drone used in healthcare [16].

Importantly, healthcare applications demand specific considerations related to patients' individual situations to provide them with an ideal service. For example, people facing emergency situations may require devices such as defibrillators to quickly revive an injured person, or blood delivery may be needed within the shortest possible time while ensuring the validity of the supply [17]. In such cases, optimization of healthcare applications, given these considerations, is highly important [18,19].

With the above considerations in mind, this study examines several problems faced by healthcare workers in emergency situations, namely: (i) the selection of drone launching centers (facilities) to maximize the ability of drones to serve patients (in this case, the drone centers must be selected from existing community centers); (ii) the distribution of available drones to drone launching centers; and (iii) the assignment of patients to drone launching centers and the available drones in those centers, considering the center's capacity and the range constraints of the drones. This problem has been referred to in the literature as the maximum-coverage facility location problem with drones (MCFLPD) [20], which is a variant of the well-known Facility Location Problem (FLP). The FLP is a typical NP-hard combinatorial optimization problem [21,22]. Usually, NP-hard problems are solved using heuristic oSr metaheuristic algorithms, rather than exact methods. Such algorithms can provide optimal or near-optimal solutions at relatively low computational costs.

The contribution of this paper is a novel heuristic for solving the MCFLPD, which is termed the maximum coverage greedy randomized heuristic (MCGRH). The MCGRH can solve the MCFLPD problem and significantly achieve high patient coverage with a low computational time compared to the approaches attempted in [20], which are a Gurobi mixed-integer programming solver [23] and two heuristics. The idea of the MCGRH is to first randomly select some facilities to be opened from among the facilities that can serve the highest weight of the required patient demands. Following this, each patient was assigned to the nearest open facility with the capacity to serve them, after which drones were assigned to patients according to the lowest battery consumed between the patient and its facility.

The remainder of this paper is organized as follows: Section 2 reviews the work related to the MCFLPD, Section 3 describes the problem formulation, Section 4 presents the proposed method, Section 5 explores the results and discussion, and Section 6 concludes with an examination of directions for future research.

2. Literature Review

In this section, we explore and discuss state-of-the-art methods for solving the FLP for drone launching centers. Few studies have been conducted on FLP and its variants by using drones [24]. In fact, FLP has important applications in the field of drone usage, particularly in post-disaster situations or when seeking to enter locations that are difficult to access. Many issues must be considered when selecting the locations of drone launching centers, including the drone battery life, drone flying range, and other variables [25,26]. In the field of healthcare, maintaining a short distance between the drone launching center and the patient is particularly important to enable a rapid medical response [27]. In this section, some efforts of prior researchers who have attempted to solve the FLP, as well as its variants, for the selection of drone-launching centers are reviewed.

MCFLPD is an NP-hard problem that was introduced in 2019 by Chauhan et al. [20] In the MCFLPD, the locations of the drone launching centers were selected from a network of prespecified capacitated facilities. The objective is to maximize the ability of drones to meet patient demands. In [20], three approaches were used to solve the problem: a mixed-integer programming solver, a novel greedy heuristic, and a three-stage heuristic (3SH). The two heuristics (greedy and 3SH) were proposed by the authors. The greedy heuristic first builds a weight matrix, where the rows represent instances of the demand point and the columns represent the facilities. Each element is filled by the percentage difference between the demand weight of a point and the battery consumed between the point and the facility. Subsequently, facilities are opened and specific demand points are assigned to them using the weight matrix. Then, the number of drones needed by each facility ND_j is defined according to the ratio of the total battery consumed between the facility and the demands assigned to it and the battery capacity of the drone. After this, opened facilities $\hat{J}$ are sorted according to ND_j. A drone from the available drones is sequentially assigned to each open facility. Finally, the demand assignment to the drone is performed according to the battery consumed between the demand and the facility.

In contrast, the 3SH proposed in [20] performs the following three steps: (i) solve a capacitated p-median facility location problem by trying to maximize the total weight, where the weight is determined by the percentage difference between the demand weight of a point and the battery consumed between the point and the facility; (ii) consider the assignment of the drones to facilities and the demand points to drones as a maximum-profit knapsack problem; and (iii) use a local exchange heuristic to improve the solution by randomly replacing a facility that has a low weight with another one. Extensive testing of the approaches proposed in [20], as well as a comparison, indicated that 3SH had the best performance in terms of both time and solution quality.

Lynskey et al. [28] studied a problem derived from the FLP, in which the objective was to select drone ports to minimize the average distance to the locations of patient demand to be served by each port. A k-means algorithm was applied to cluster the demand areas. Next, several traveling salesman algorithms—2-opt, a genetic algorithm, exhaustive search, and ant colony optimization (ACO)—were applied to identify the preferred locations of drone ports to ensure that the demand in each cluster could be met by drones, as well as to ensure that the drone trips would involve the shortest routes. The ACO algorithm outperformed other algorithms investigated in this study.

In another study, Shavarani et al. [29] studied an FLP to identify an optimal solution for finding the locations of launching drones and recharging stations. Their objective was to minimize the total cost of the system. A genetic algorithm and a hybrid genetic algorithm were applied and compared, and the latter was found to provide the best solution. The last noteworthy study is Kim et al. [30], wherein the authors introduced a stochastic framework for regions affected by disasters. The framework accounts for uncertain trip distances for drones and aims to solve the FLP by identifying the optimal number of drone launch facility locations and their capacities. The authors used Benders decomposition and linear programming rounding to develop a heuristic algorithm that provides high-quality and efficient solutions.

The literature shows that limited research has examined the use of drones in the healthcare field, despite their importance in saving human lives and reducing the burden on both healthcare facilities and individual patients. The paucity of research in this area has become even more noticeable with the emergence of the recent COVID-19 pandemic, where drone applications have tremendously alleviated the delivery of services to remote areas while being cheaper, safer, and faster than traditional modes of transport.

Another important finding from this literature review is that exact methods are evidently unsuitable for solving large datasets of FLP and its variants. As an NP-hard problem, exact methods require very long processing times to generate solutions. Therefore, for real-life problems with large datasets, the existing literature typically utilizes heuristics and metaheuristics to identify near-optimal solutions (in terms of quality) with an acceptable processing time.

3. Problem Formulation

As previously mentioned, the maximum coverage facility location problem with drones (MCFLPD) [20] is a variant of the Facility Location Problem (FLP). Specifically, the FLP serves the demands of a set of customers using a set of facilities that can be allocated or chosen from existing locations. In contrast, in the MCFLPD, there are additional features to the FLP, namely drone-to-facility and demand-to-drone distribution, in addition to the consideration of the drone range constraints. These features require hard computations, which makes the MCFLPD more complex than the classical FLP.

The aim of this study was to implement a solution to improve the MCFLPD results described by Chauhan et al. [20] Specifically, the MCFLPD is concerned with identifying the optimal or near-optimal number of locations of drone launching centers such that the coverage is maximized for serving patients. Drone centers must be chosen from the existing community centers. Figure 2 shows an example of the locations of the community centers and patients to be served.

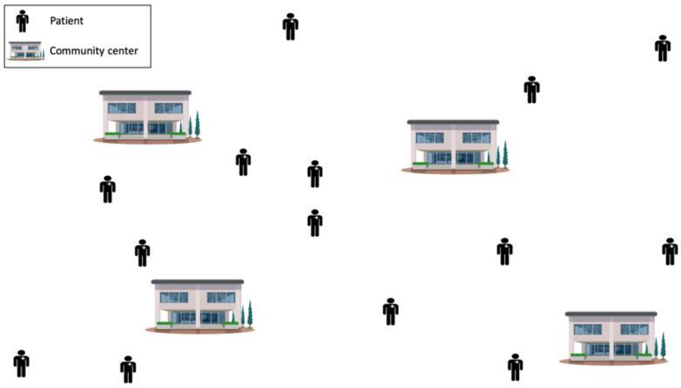

Figure 2. An example of locations of community centers and patients.

Formally, within a certain planning period, there is a set of locations for patient I, where each has a demand weight w_i, along with a set of potential facility locations (i.e., drone launching centers) J and a set of drones K, each with a full battery charge. The MCFLPD formulation, as described by Chauhan et al. [20], is shown in Table 1.

Table 1. Formulation of MCFLPD.

Sets	I: Set of locations of patients $(i \in I)$ J : Set of all possible facility locations $(j \in J)$ K : Set of drones $(k \in K)$

Table 1. *Cont.*

Parameters	η: Efficiency of the power transfer θ_s : Lift-to-drag ratio B: Drone battery capacity b_{ij} : Battery consumed on one trip between patient $i \in I$ and facility $j \in J$ d_{ij} : Travel distance between patient $i \in I$ and facility $j \in J$ m_b : Drone battery mass m_t : Drone mass tare, without battery and load p: Maximum number of facilities to be opened U: Capacity of each opened facility w_i : Weight of demand at patient location $i \in I$
Decision Variables	$x_{ijk} = \begin{cases} 1 & \textit{if patient } i \textit{ is served by the } k^{th} \textit{ drone of facility } j \in J \\ 0 & \textit{otherwise} \end{cases}$ $y_j = \begin{cases} 1 & \textit{if facility is opened at } j \in J \\ 0 & \textit{otherwise} \end{cases}$ $c_{ij} = \begin{cases} 1 & \textit{if patient } i \textit{ is a potential patient at facility } j \in J \textit{ where } b_{ij} < B \\ 0 & \textit{otherwise} \end{cases}$ $a_{ij} = \begin{cases} 1 & \textit{if patient } i \textit{ is assigned to facility } j \in J \\ 0 & \textit{otherwise} \end{cases}$
Objective Function	To maximize the ability of the fleet of drones to cover the total patient demand $\sum\limits_{i \in I} \sum\limits_{j \in J} \sum\limits_{k \in K} w_i x_{ijk}$
Constraints	1. Each patient location is covered at most once 2. The number of chosen facilities must be less than or equal to the maximum number of existing facilities 3. The drone battery range must cover the trip distance 4. The patient demand served by each selected facility must be less than or equal to the facility's capacity 5. Drones must be assigned only to the selected facilities
Assumptions	• The cost of transportation is not considered • If the demand of a patient is higher than the carrying capacity of the drone, multiple trips will be conducted to meet the patient's needs; in this case, the number of trips will be based on drone capacity • Each drone can complete several one-to-one trips (where a trip lasts from drone launching center to a patient's location and back) until the battery range B is met • Recharging of drone batteries is not considered (i.e., assume that the drone batteries are fully charged prior to the start of a trip) • The effects of charging cycles and weather on drone battery capacity are not considered • The effect of obstacles such as high buildings or mountains on drone battery capacity is not considered • The total power consumed b_{ij} in a delivery from facility j to patient i is given by Equation (1) [31]. • The capacity of each opened facility is calculated by Equation (2) [32].

Equation (1) [31] calculates b_{ij}, where a drone travels to patient i with a load w_i and drops w_i then returns empty to the facility j.

$$b_{ij} = \frac{m_t + m_b + w_i}{\theta_s\ \eta}\, d_{ij} + \frac{m_t + m_b}{\theta_s\ \eta}\, d_{ij} \qquad \forall\, i \in I,\ \ j \in J \tag{1}$$

Equation (2) [32] calculates U, which is equal to the total demands weight of all patients divided by 80 percent (according to the reference [32], the average utilization of the facilities is equal to 80 percent of the total available capacity) of the number of facilities to be opened.

$$U = \frac{\sum_{i \in I} w_i}{0.8\ \ p} \tag{2}$$

4. Solution Method

This paper proposes a novel heuristic known as MCGRH, the purpose of which is to solve the MCFLPD problem. This guarantees that all constraints in Table 1 are satisfied. At the outset, the idea of the algorithm is to select a set of facilities S to be opened, with size p, from the set of potential facilities J that maximizes $\sum_{i \in I} \sum_{j \in S} y_j \, c_{ij} w_i$ (i.e., maximizing the total demand of patients that it is possible to serve). In turn, the algorithm assigns each potential patient to the nearest open facility $j \in S$ with the available capacity U_j. Any available drones are then assigned to the opened facilities according to the highest unserved demand weights $\sum_{i \in I} w_i \, a_{ij}$, where $a_{ij} = 1$ and $j \in S$. Finally, patients are assigned to drones (in each opened facility) according to the lowest battery consumed b_{ij} followed by the highest demand weight w_i of all unserved patients of a specific facility j, such that for each drone k, the total battery consumed by the drone in serving the demand is less than the drone's total battery capacity (that is, $\sum_{i \in I} b_{ij} \le B_k$, where $j \in S$.)

The proposed method proceeds in three steps: (i) select the number of facilities to be opened from the potential facilities, (ii) assign patients to the opened facilities, and (iii) assign drones to the opened facilities and assign patients to those drones. Algorithms 1–3 present the details of these steps. An explanation of each algorithm is provided below.

4.1. Select a Number of Facilities to Be Opened from the Potential Facilities

The purpose of this algorithm is to select from potential facilities J a number of facilities p to be opened that maximizes the total demand of patients that can be served. Let I be the list of all potential patients and J be the list of all potential facilities with their potential patients such that patient i can be covered by facility j if a drone can reach i in one trip using its battery charge. In other words, $b_{ij} \le B$, where b_{ij} is calculated using Equation (1). In addition, suppose that p is the maximum number of facilities to be opened. Let S be an empty list of the initial solution of size p to maintain open facilities. Let U_f be the capacity of each facility calculated according to Equation (2) (i.e., all facilities have identical capacities).

After implementing the above steps, the algorithm repeats the following steps p times, until J is exhausted or until all patients in I have been served:

First, calculate $\sum_{i \in I} w_i \, c_{ij} \, \forall \, j \in J$. That is, we calculated the total demand weights of the potential patients of each facility $j \in J$. In turn, sort J is sorted in descending order according to these values. Then, select a subset HF consisting of n facilities from J with the highest $\sum_{i \in I} w_i \, c_{ij}$, and select a facility f—to be opened from HF at random. In this case, the idea is to utilize a greedy randomized approach to ensure that a facility is selected from among the top facilities that can cover large patient demand. This maximizes the coverage and, at the same time, eliminates selection bias and variegates the results using random selection, which cannot be achieved using only the greedy criterion. Next, remove f is removed from J so that it will not appear again in subsequent iterations when selecting a new facility to be opened.

Second, the potential patients i of f, where $i \in I$ and $c_{if} = 1$, in descending order according to the demand weight w_i. Following this, iterate over these patients and mark them as covered until U_f is filled, or until the list of f's potential patients is exhausted (thus making these patients unavailable for other facilities). Hence, if there is more than one facility in J with the same potential patient A, marking A as covered by f will change $\sum_{i \in I} w_i \, c_{ij} \, \forall \, j \in J$ (i.e., the total demand weights of the potential patients of each facility $j \in J$). It is noteworthy that this affects their appearance in HF in the next iteration. The last part of the second algorithm adds f to S.

Finally, list S now contains the facilities to be opened that can serve the greatest possible demands. Algorithm 1 presents the details of the above steps.

Algorithm 1: Select a number of facilities to be opened from the potential facilities.

```
$I \leftarrow$ List of all potential patients
$J \leftarrow$ List of all potential facilities with their potential patients
$p \leftarrow$ Maximum number of facilities to be opened
$S \leftarrow \{\ \}$
Calculate $U_f = \frac{\sum_{i \in I} w_i}{0.8\ p}$ \\ capacity of each facility $f$ in $S$
Repeat
        Calculate $\sum_{i\ \in\ I}\ w_i\ c_{ij} \forall\ j\ \in J$ \\ the total demand weights of the potential patients of each facility $j\ \in J$
        $Sort(J)$ \\ in descending order of the total demand weights of the potential patients
        $HF \leftarrow J[1 \ldots n]$
        $f = Random(HF)$
        $Remove(f, J)$
        $Sort(potiential\ patients\ of\ f)$\\ in descending order of patients demand weight $w_i$
        Repeat
                $MarkAsCoverd(patient_i, f)$
        Until filling $U_f$ OR the list of $f$ 's potential patients is exhausted
        $S \leftarrow S \cup\ f$
Until Size $p$ is reached OR $I$ is exhausted OR $J$ is exhausted
Output: $S$
```

4.2. Assign Patients to the Opened Facilities

The purpose of this algorithm is to assign patients to the nearest open facility in S whose capacity is available, and drone constraints are satisfied. Using S from Algorithm 1 as an input list of opened facilities with size p, along with the list of all potential patients I, several operations should be performed. First, for each patient i in I, traverse the patients one by one until the list of patients ends, such that the current patient i is assigned to the first open facility f in the list S based on two conditions: (i) f is the nearest opened facility to i, where $b_{if} \leq B$; and (ii) there is available capacity in f to add i (i.e., $w_i\ \leq U_f$). The second operation recalculates U_f (i.e., the available capacity) and $\sum_{i\ \in\ I}\ w_i\ a_{if}$ (total demand served by f). Algorithm 2 presents the details of the above steps.

Algorithm 2: Assign patients to the opened facilities.

```
Input: $S$ \\ the list of opened facilities from Algorithm 1
$I \leftarrow$ List of all potential patients
For each patient $i$ in $I$ Do
        Pick best $f$ for $i$ from $S$ \\ the nearest opened facility where $b_{if} \leq B$ with available capacity $U_f$
        $Assign(i, f)$
        Recalculate available capacity $(U_f)$ and total required demands weights from $f$ $(\sum_{i\ \in\ I}\ w_i\ a_{if})$
Output: $S$
```

4.3. Assign Drones to the Opened Facilities and Assign Patients to Those Drones

The purpose of this algorithm is to assign the available drones to the opened facilities in S according to the required demand weights of their assigned patients. This is achieved using the concept of the maximum-profit knapsack problem, as proposed in [20], along with additional conditions. The idea is to prioritize facilities that have large required demand weights by assigning more drones to them and, at the same time, assigning more patients of these facilities to drones to maximize coverage.

Given S, which is the input list of the opened facilities from Algorithm 2, for each facility f in S, in order to give priority to patients who consume lower batteries, sort the assigned patients i of f, where $i\ \in\ I$ and $a_{if}\ = 1$, in ascending order according to b_{if}, followed by w_i in descending order. In other words, if two patients are equal in b_{if}, the patient with the higher w_i will occur first in the list.

Assuming K is the number of available drones and $Total\_demand\_served$ is initialized at a value of 0, repeat the following steps K times or until all patients in S are served:

First, select a drone k from the set of available drones K. To maximize coverage, find MF facility of S that has the maximum demand weights of the patients assigned to this facility that are not yet assigned to any drone (i.e., maximum facility MF from S according to maximum $\sum_{i \in I} w_i \, a_{if}$, where $x_{ifk} = 0 \; \forall \; f \in S$ and $\forall \; k \in K$). In turn, assign k is assigned to MF and the demand served by drone k $(Demand\_served_k)$ to 0.

Second, traverse the patients assigned to MF who are not assigned to any drone, and then check whether the value of $b_{i \; MF}$ of the current patient i is lower than the remaining battery capacity B_k of drone k. If this is the case, assign i to k, subtract its consumed battery $b_{i \; MF}$ from B_k, and add w_i to the drone's $Demand\_served_k$; otherwise, if the current patient i cannot be served by the remaining B_K and to maximize the coverage, try to replace patient i with patient z already assigned to k, where $w_z < w_i$, without violating the drone constraints. This is done by sorting the patients already assigned to k according to ascending demand weights w, traversing them one by one, and checking whether there is a patient z for whom $w_z < w_i$ and $b_{z \; MF} +$ the remaining $B_k \; \geq \; b_{i \; MF}$. Then, release the assigned patient z and assign the current patient i to k. Also, and update $Demand\_served_k$ and B_k according to the changes.

Finally, after completing the assignments to drone k, we update $Total\_demand\_served$. After finishing all drones or serving all patients in S, calculate and return $Coverage$, which is equal to the percentage of $Total\_demand\_served$. Algorithm 3 presents the details of the above steps.

Algorithm 3: Assign drones to the opened facilities and assign patients to those drones.

```
Input: $S$ \\ from Algorithm 2
$K \leftarrow$ number of the available drones
$Total\_demand\_served \leftarrow 0$
$Sort\ (patients\ in\ each\ facility\ f\ in\ S)$ \\ by $b_{if}$ in ascending order, followed by $w_i$ in descending order
Repeat
  $Select$ (new $k \in K$)
  $B \leftarrow$ battery capacity of $k$
  $MF \leftarrow \max(S)$ \\ according to max $\sum_{i \in I} w_i \, a_{if}$ where $x_{ifk} = 0 \; \forall \; f \in S$ and $\forall \; k \in K$
  $Assign\_drone\ (k, MF)$
  $Demand\_served_k \leftarrow 0$
  For each patient $i$ assigned to $MF$ Do \\ $i$ is not assigned to any drone ($x_{ifk} = 0 \; \forall \; f \in S$ and $\forall \; k \in K$)
    $b_{i\;MF} \leftarrow$ battery consumed between current patient $i$ and current facility $MF$
    IF $b_{i\;MF} \leq B$ Then
      $Assign\_patient\ (i, k)$
      $B \leftarrow B - b_{i\;MF}$
      $Demand\_served_{\;k} \leftarrow Demand\_served_{\;k} + w_i$
    Else
      $Sort(patients\ assigned\ to\ k)$ \\ according to the ascending demands weights $w$
      For each patient $z$ assigned to the drone $k$ Do
        IF $w_z < w_i$ AND $b_{z\;MF} + B \; \geq \; b_{i\;MF}$ Then
          $Release\ (z, k)$
            $Assign\_patient\ (i, k)$
          $B \leftarrow B + b_{z\;MF} - b_{i\;MF}$
          $Demand\_served_k \leftarrow Demand\_served_k + w_i - w_z$
          $break$
```

Algorithm 3: *cont.*

```
$Total\_demand\_served \leftarrow Total\_demand\_served + Demand\_served_k$
**Until** $K$ times **OR** until all patients in $S$ are served
$Coverage \leftarrow$ The percentage of $Total_{demand_{served}\text{ of patients covered}}$
**Output:** $Coverage$
```

5. Results and Discussion

This section describes the dataset used to test the proposed method, the MCGRH, and the parameter settings. A numerical analysis of the proposed approach is also presented.

The MCGRH was implemented using Python software. Experiments were conducted using a computer with an Intel Core i7 processor running at 3.1 GHz using 16 GB 2133 MHz LPDDR3 of RAM running Macintosh HD. SPSS version 28.0.0.0 was used for statistical analysis [33].

5.1. Dataset and Parameter Settings

We used the Portland metropolitan area as a case study to apply MCGRH [20]. There were 122 patients in the study area. The patients are represented by the centroid of the ZIP code tabulated areas (ZCTAs). There are 104 potential facility locations, which are community centers in the Portland metropolitan area. Figure 3 shows the patient and potential facility locations for the case study.

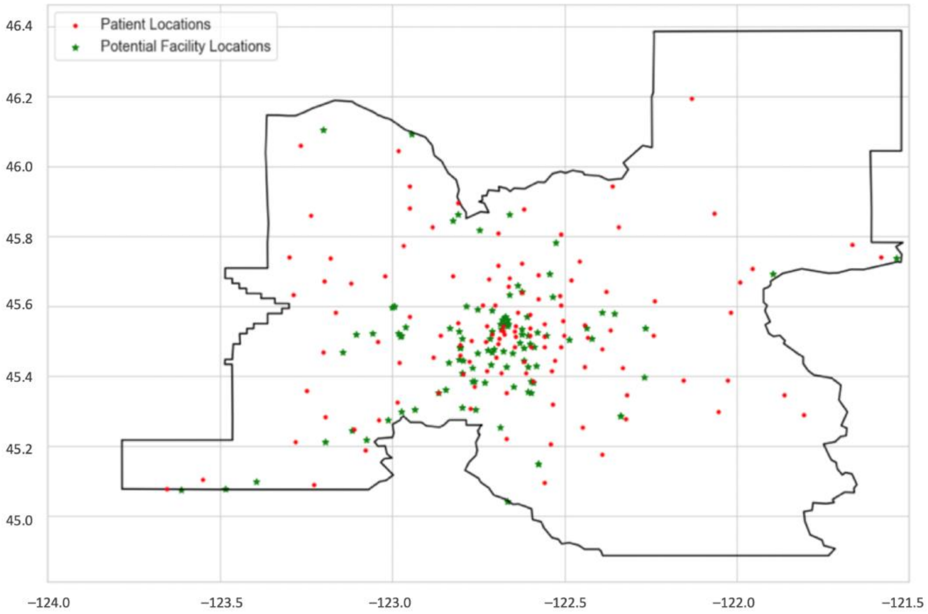

Figure 3. Patient locations and potential facility locations in the Portland Metropolitan Region.

The total payloads requested by patients were 366.5 kg. It is worth mentioning that there were four patients (ZCTA points 97028, 97049, 97064, and 98616) with a total demand of 15.75 kg (their respective demands were 4.75, 2.25, 4, and 4.75) who could not be reached by any facility. This is because of the battery capacity required for one trip between each of them and any of the potential facilities, which is greater than the battery capacity of the drone (777 W h). This means that the optimal coverage of patients that could be reached was 350.75 kg (95.7%). U is the capacity that each facility can offer, which is calculated based on the total demand weight of patients w_i and the number of facilities to be opened p [20], as shown in Equation (1).

As assumed in [20], the number of facilities to be opened ranges from 5 to 30. We computed the traveling distance between a patient and a potential facility in miles using the Euclidean distance based on the latitude and longitude of their locations. According to the average latitude of the dataset, 1-degree latitude (y-axis distance) = 111.13976776 km and 1-degree longitude (x-axis distance) = 78.000735479 km. It was assumed that there were no obstacles. The parameters we used for drones in this paper are based on reference [20] and are shown in Table 2.

Table 2. The parameters of the drone.

Power Transfer Efficiency (η)	0.66
Lift-to-Drag Ratio (θ_s)	3.5
Tare Weight	10.1 kg
Maximum Payload	5 kg
Battery Capacity	777 W h

5.2. Numerical Analysis

In this study, we developed a new heuristic for solving an MCFLPD, referred to as the MCGRH. We compared the results of MCGRH with those of the Gurobi mixed-integer programming solver (MIP), greedy heuristic, and three-stage heuristic (3SH) methods used in [20] to achieve the best maximum coverage within the minimum time.

In terms of coverage paired with time performance, the method that achieved the best results in [20] was 3SH. As described in Section 2, 3SH comprises three stages. In the first stage, a weight $\frac{w_i}{b_{ij}}\, a_{ij}$, which is the weight of assigning each potential demand point $i \in I$ to each facility $j \in J$, where $b_{ij} < B$. At this stage, the algorithm attempts to maximize the total weights assigned to each facility. The second stage deals with the problem of assigning drones to facilities and demand points as a maximum profit knapsack problem. In the third stage, the solution is improved by performing a local exchange heuristic, which removes the facility with the lowest demand and replaces it with the available ones at random.

In contrast, the MCGRH overpasses 3SH, where it attempts to work on elements that maximize coverage with fewer processes. Thus, in MCGRH, the p facilities to be opened are selected from the facilities with the highest demand weights for their potential patients $\sum_{i\,\in\, I}\, w_i\, c_{ij}\; \forall\; j\;\in J$, which is achieved using a randomized-greedy approach. In addition, the assignment of potential patients to the opened facilities is performed in a greedy manner by selecting the nearest patient with available capacity U. This is intended to reduce the amount of battery consumed by the drone to serve patient i, enabling as many patients as possible to be served using the drone. In other words, because the two factors that affect the consumed battery b_{ij} are the demand weight w_i and the distance d_{ij} between patient i and facility j, and because w_i is fixed and d_{ij} is variable, selecting the facility with the shortest distance from i gives more importance to the demand weight than the distance when calculating the amount of consumed battery. This helps achieve the objective of maximizing the total demand served.

Hence, the MCGRH addresses the problem of assigning drones to facilities as a maximum-profit knapsack problem. In addition, the proposed method addresses the problem of assigning patients to drones by prioritizing patients with the lowest consumed battery b_{ij}, followed by the highest demand w_i. These features account for the fact that MCGRH outperformed 3SH, as shown in detail in due course.

Table 3 shows the results of all the considered methods on 22 different instances reported in [20], as well as the MCGRH. The instances were grouped by the number of opened facilities p followed by the number of available drones K. The table shows the performance of the four methods as measured by time (in seconds) and coverage (i.e., the percentage of the total patient demand that is satisfied). The Gurobi MIP is an exact and deterministic solution method that runs until a solution is found or a limit of 7200 s is reached. 3SH and MCGRH are not deterministic, so the table shows their statistical results (average, minimum, and maximum) over 30 different runs of each instance. In addition, in the last row of the table, the average time and coverage of the 22 instances are indicated (since we are concerned with the minimum time and maximum coverage, we considered them when computing the overall average results of the 3SH and MCGRH in the last row of Table 3). MCGRH achieved a coverage of 95.1%, whereas the optimal coverage was 95.7% (as explained in Section 5.1) in less than two minutes, which outperformed the other methods.

Table 3. Comparison of Gurobi, Greedy heuristic, 3SH and MCGRH.

p [1]	$\lvert K \rvert$ [2]	**Gurobi**		**Greedy**		**3SH**						**MCGRH**					
		Time (s)	**Coverage (%)**	**Time (s)**	**Coverage (%)**	**Time (s)**			**Coverage (%)**			**Time (s)**			**Coverage (%)**		
						Avg	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**
5	20	7200	**56.4**	**0.1**	45.2	14.7	14.4	15	54.5	53.6	55.1	0.67	0.55	1.11	51.53	45.4	56.1
5	25	7200	61.9	**0.1**	50.3	15.9	15.6	16.2	59.5	58.6	60.2	0.61	0.56	0.78	57.89	54.5	**62.6**
5	30	7200	66.3	**0.1**	55.3	16.7	16.4	17.1	63.7	62.9	64.5	0.67	0.6	0.81	63.38	59.3	**66.7**
5	35	7200	70.2	**0.1**	58.9	18.3	17.6	22.8	67	66.5	67.9	0.69	0.59	1.3	68.09	64.9	**71.8**
5	40	7200	72.7	**0.1**	62.5	18.8	18.2	19.1	69.9	69.1	70.5	0.67	0.59	1.09	71.02	67.8	**74.1**
10	20	7200	**64.4**	**0.2**	48.2	16.6	16.1	16.9	61.4	59.8	62.6	0.70	0.67	0.84	57.36	54	61.6
10	30	7200	**75**	**0.2**	59.8	18.3	17.9	19	71.5	70.1	72.9	0.80	0.7	1.34	69.98	65.3	73.2
10	40	7200	**83.8**	**0.2**	67.1	20.1	19.2	20.8	78.4	76.3	80.4	0.85	0.69	1.35	79.15	76.3	82.1
15	30	7200	**79.7**	**0.2**	59.2	21.9	21.4	22.7	75.2	73.2	77.2	1.05	0.82	1.72	72.40	68.9	76.6
15	45	7200	**90.2**	**0.2**	73.1	24.3	23.4	25.5	83.9	80.3	86.6	0.92	0.82	1.55	86.43	83.6	88.7
15	60	7200	92.6	**0.3**	73.1	24.8	23.6	25.7	85	81	88.7	1.07	0.83	1.59	90.43	86.8	**94.5**
20	20	7200	**71.2**	**0.3**	52.8	25.1	24.6	25.6	65.8	63.2	67.5	1.04	0.93	1.19	59.32	56.1	63.9
20	40	7200	**90.4**	**0.3**	70.7	28.7	28.1	29.3	84.2	82.7	85.3	1.03	0.95	1.26	83.77	80.3	86.9
20	60	320	93.8	**0.3**	72.2	30.4	28.6	31.8	87.2	83.6	90.1	1.06	0.96	1.32	92.33	87.6	**95.1**
20	80	36	93.8	**0.4**	72.2	30.9	29.6	32.8	87.5	83.6	91.3	1.03	0.95	1.21	91.36	86.8	**95.1**
25	25	7200	**79.6**	**0.3**	53.6	33	32.1	35	71.5	69.2	73.3	1.10	1.05	1.36	67.70	64.3	71.4
25	50	337	**93.8**	**0.4**	71.4	36.6	35.7	38.3	88.9	85.2	92.2	1.21	1.11	1.36	90.94	87.6	93.7
25	75	27	93.8	**0.5**	71.4	37.4	36.2	39	88.2	84.2	91	1.23	1.08	1.6	92.21	87.9	**94.7**
25	100	43	93.8	**0.5**	71.4	38.1	36.9	39.5	89.5	86.1	92.2	1.24	1.07	1.84	92.16	87.9	**94.7**
30	30	7200	**85.3**	**0.3**	60.6	40.3	39	41.7	76.8	74.4	80.2	1.31	1.16	1.78	72.24	68.4	77.3
30	60	23	93.8	**0.5**	74.8	44.3	43.3	45.5	90.9	88.7	92.8	1.47	1.22	2.1	93.21	89.9	**95.1**
30	90	31	93.8	**0.6**	74.7	45.1	44.2	46.4	90.7	88.7	93	1.22	1.14	1.52	92.87	89.4	**95.1**
Average		4946.2	**81.65**	**0.28**	63.57		26.46				78.89		0.87				80.5

[1] p = number of the opened facilities. [2] $|K|$ = number of the available drones.

5.2.1. Coverage

Coverage is the percentage of the total accommodated patient demand. In [20], the authors reported that the greedy algorithm had the best time performance, albeit with very weak coverage compared to the other methods. Gurobi was the best of the three methods in terms of coverage but was associated with the worst time performance, which is attributable to the fact that it requires an unacceptably long running time to achieve a feasible solution. However, 3SH achieved good coverage (96.6% of Gurobi's solutions on average), whereas when compared with Gurobi's time performance, 3SH took only approximately 27 s on average.

Figure 4 shows a comparison of the maximum coverage of the MCGRH and the other three methods of [20], grouped by the number of opened facilities. The x-axis represents the number of opened facilities and available drones, while the y-axis represents the percentage coverage. The average maximum coverage obtained using each method was as follows: Gurobi (81.65%), greedy (63.57%), 3SH (78.89%), and MCGRH (80.5%). Evidently, MCGRH is the closest to Gurobi in terms of the average maximum coverage (approximately 98.6% of Gurobi's solutions on average), and it exceeded Gurobi in 11 instances (see Table 3 and Figure 5). Because the objective is to maximize coverage, a comparison between the Gurobi and 3SH methods is considered.

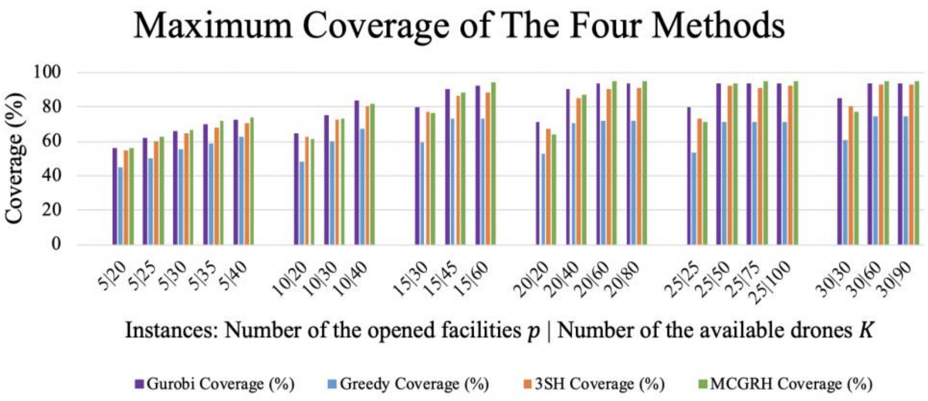

Figure 4. Comparison between the maximum coverage of the four methods.

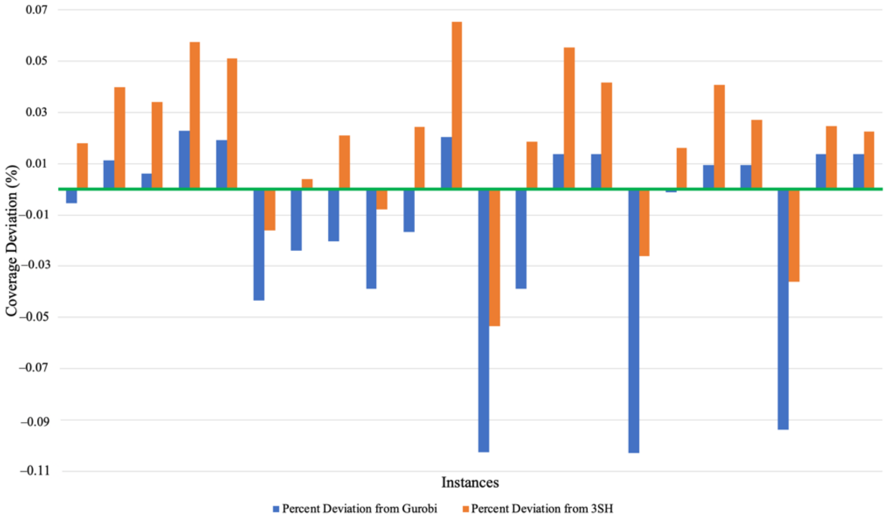

Figure 5. Percent deviation of MCGRH from Gurobi and 3SH.

Figure 5 shows the percentage deviation of coverage between Gurobi, 3SH, and MCGRH for all instances, where a positive deviation indicates that the MCGRH's result is better for that instance. In half of the instances (11 of 22), MCGRH exceeded Gurobi's results, whereas in 17 of 22 instances, MCGRH outperformed 3SH.

We used the Wilcoxon signed-rank test to ensure that the analysis relied on a solid statistical basis. We applied the test using SPSS to compare the coverage obtained by MCGRH and Gurobi, on the one hand, and MCGRH and 3SH, on the other. The null hypotheses and results for both tests are presented below.

1. Comparison of MCGRH and Gurobi.

- Null hypothesis: The population distributions of the Gurobi algorithm and MCGRH were identical with respect to coverage.
- Results: The results were comparable for both algorithms (11 negative and 11 positive ranks). In addition, the p-value = 0.262, where 0.262 > 0.05; thus, the null hypothesis was accepted.

2. Comparison of MCGRH and 3SH

- Null hypothesis: The population distributions of the 3SH and MCGRH are identical with respect to coverage.
- Results: There were 5 negative and 17 positive ranks. Thus, most of the pairs were positive ranks, which means that MCGRH had a larger coverage than 3SH. In addition, the p-value = 0.008, where 0.008 < 0.05; therefore, the null hypothesis was rejected. In other words, the results of the MCGRH were statistically significantly better than those of the 3SH.

For the sake of a visual comparison with the results in [20], Figures 6 and 7 show illustrations of solutions for some instances obtained using our MCGRH.

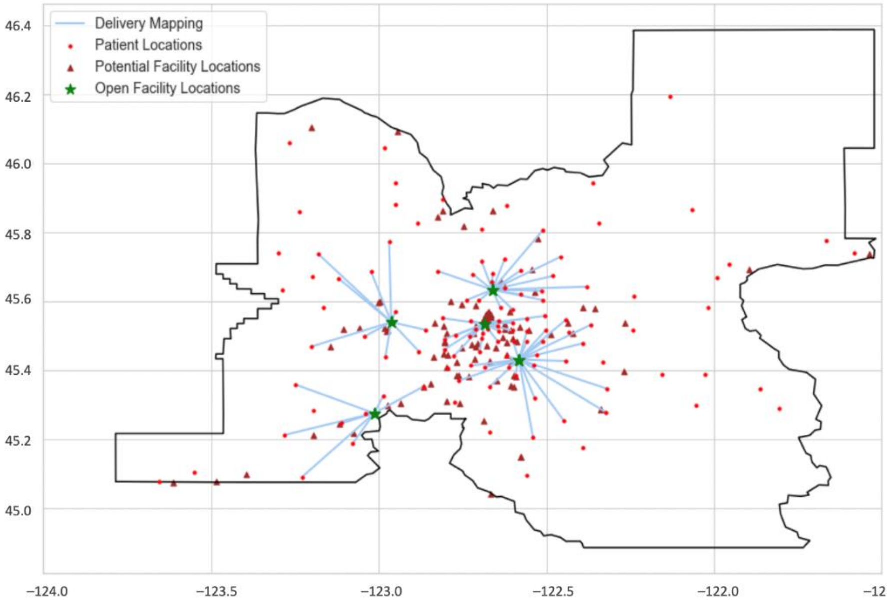

Figure 6. MCGRH solution for [*p* = 5 and K = 35] covers 71.8% of the total patients demands in kg.

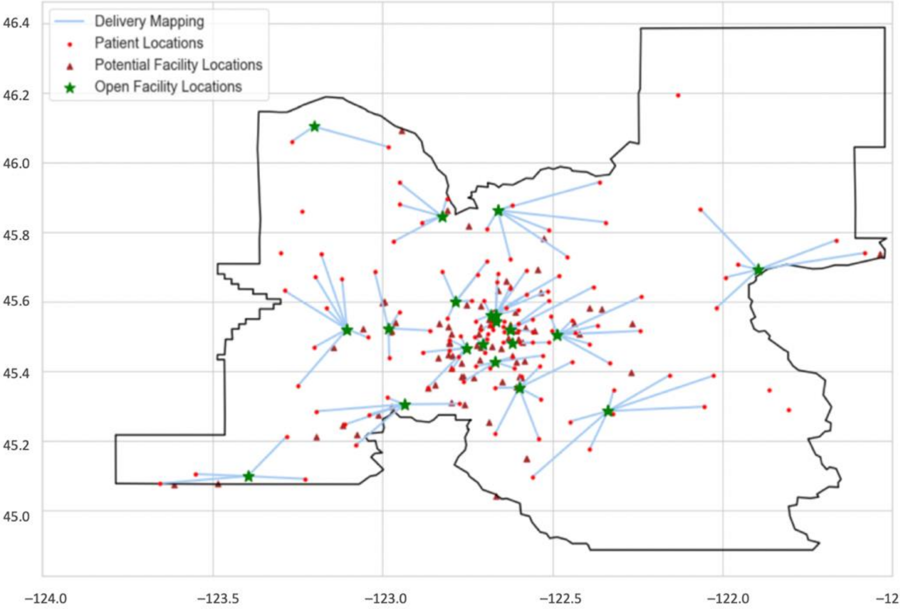

Figure 7. MCGRH solution [*p* = 20 and K = 60] covers 95.1% of the total patients demands in kg.

Finally, we calculated the average energy consumed per percent coverage for every 30 runs for each of the 22 instances using the following formula:

$$Energy/Coverage = \frac{Average\ Battery\ Used \times Number\ of\ Drones\ Used}{Coverage} \quad (3)$$

Compared to Gurobi, the greedy heuristic, and 3SH, the average energy consumed per percent of coverage of MCGRH was higher by approximately 20%, 31%, and 32%, respectively. However, as previously mentioned, MCGRH achieved the best coverage in the least time; therefore, it is expected that this comes at the expense of a slight increase in the energy consumed (i.e., because more patients are being served). In fact, given that the same resources are used for each method, and because the objective is to maximize the patients covered, it is more important to maximize coverage and help more patients in a very fast time than to reduce the consumption of the drones' batteries. This is especially true given that the difference in energy/coverage is not large when compared with the other methods.

5.2.2. Time Performance

Because the objective is to maximize coverage, we compared the Gurobi and 3SH methods because they achieved greater coverage than the greedy method. As shown in Table 3, the average time performances of the Gurobi was 4946.23 s, 3SH 26.46 s, and MCGRH 0.87 s. Although the experiments were conducted on different computers, it can still be observed that the MCGRH is substantially faster than the other methods. Furthermore, given that the hardware specifications of the computer used to test the MCGRH were lower than those of the machine used to test the Gurobi and 3SH methods, this again attests to the remarkable time performance of the MCGRH. In fact, the MCGRH runs extremely rapidly (less than 1 s on average). Again, we performed the Wilcoxon signed-rank test using SPSS to compare the time performance of MCGRH and each of the two other methods. The results are as follows:

1. Comparison of MCGRH and Gurobi.

- Null hypothesis: The population distributions of the Gurobi and MCGRH are identical with respect to time performance.
- Results: The ranks of all pairs were negative (22 negative and 0 positive), indicating that MCGRH outperformed Gurobi in terms of time performance. In addition, p-value = 0.001, where 0.001 < 0.05, indicating that the null hypothesis is rejected.

2. Comparison of MCGRH and 3SH.

- Null hypothesis: The population distributions of the 3SH and MCGRH are identical with respect to time performance.
- Results: The ranks of all pairs were negative (22 negative ranks and 0 positive ranks), indicating that MCGRH outperformed 3SH in terms of time performance. In addition, p-value = 0.001, where 0.001 < 0.05; therefore, the null hypothesis was rejected.

6. Conclusions

The delivery of medical supplies and the provision of aid to patients using drones can significantly contribute to improving healthcare services. This study aims to provide high-quality solutions within a reasonable time for the maximum coverage facility location problem with drones (MCFLPD), which is more complex than the traditional facility location problem. State-of-the-art methods in the literature have been proposed to solve this problem, including a Gurobi mixed-integer programming solver, which gives acceptable-quality solutions in terms of coverage but has an unacceptably long running time to find a feasible solution; a greedy heuristic, which is extremely fast but achieves low coverage compared to the other methods; and a three-stage heuristic (3SH) algorithm, which achieves around 96.6% of Gurobi's coverage within approximately 27 s.

In this paper, we propose a new heuristic called the maximum coverage greedy randomized heuristic (MCGRH) to solve MCFLPD. MCGRH is distinguished by the following: (i) the selection of facilities to be opened at random from those that can serve the largest number of patients with the highest demand; (ii) the assignment of patients to the nearest open facilities that are available to serve them; and (iii) the assignment of drones to the opened facilities that have the highest demands, as well as the assignment of patients of those facilities to those with the lowest battery consumed. We compared MCGRH with the state-of-the-art methods mentioned above, considering both time performance and solution quality. We used the Wilcoxon signed-rank test for statistical comparison, with the results indicating that the MCGRH excelled in solution quality (i.e., coverage), together with time performance. In particular, MCGRH produced a coverage of more than 80% (approximately 98.6% of Gurobi's average coverage) in less than 1 s of average processing time.

In future work, we intend to improve the MCGRH to reduce the total energy consumed while maintaining high coverage within a short processing time, besides considering obstacle avoidance. Moreover, it is possible to explore new variants of the problem that may reflect more realistic healthcare applications, such as making battery stations available to recharge a drone's battery during its trip.

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

Funding: This work was supported by the Deanship of Scientific Research, King Saud University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data presented in this study are available in ScienceDirect at https://doi.org/10.1016/j.trc.2018.12.001 [20] (accessed on 29 December 2021).

Acknowledgments: The authors would like to thank the Deanship of Scientific Research at King Saud University for funding and supporting this research through the initiative of DSR Graduate Students Research Support (GSR).

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

References

1. 2021 North American Wildfire Season-Center for Disaster Philanthropy. Available online: https://disasterphilanthropy.org/disaster/2021-north-american-wildfire-season/ (accessed on 29 December 2021).
2. 2021 Haiti Earthquake and Tropical Storm Grace-Center for Disaster Philanthropy. Available online: https://disasterphilanthropy.org/disaster/2021-haiti-earthquake-and-tropical-storm-grace/ (accessed on 29 December 2021).
3. 2021 Atlantic Hurricane Season Fast Facts-CNN. Available online: https://edition.cnn.com/2021/04/13/weather/2021-atlantic-hurricane-season-fast-facts/index.html (accessed on 29 December 2021).
4. Coronavirus Disease 2019. Available online: https://www.who.int/emergencies/diseases/novel-coronavirus-2019 (accessed on 23 March 2020).
5. Australia Fires: A Visual Guide to the Bushfire Crisis-BBC News. Available online: https://www.bbc.com/news/world-australia-50951043 (accessed on 23 March 2020).
6. Radu, S. 5 of the Deadliest Natural Disasters in 2019 | Best Countries | US News. Available online: https://www.usnews.com/news/best-countries/slideshows/5-of-the-deadliest-natural-disasters-in-2019 (accessed on 5 September 2020).
7. 2018 Indonesia Quakes and Tsunamis: Facts, FAQs, and How to Help | World Vision. Available online: https://www.worldvision.org/disaster-relief-news-stories/2018-indonesia-earthquake-facts (accessed on 23 March 2020).
8. Fukubayashi, Y.; Kimura, M. Improvement of rural access roads in developing countries with initiative for self-reliance of communities. *Soils Found.* **2014**, *54*, 23–35. [CrossRef]
9. Otto, A.; Agatz, N.; Campbell, J.; Golden, B. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks* **2018**, *72*, 411–458. [CrossRef]
10. Boysen, N.; Fedtke, S.; Schwerdfeger, S. Last-mile delivery concepts: A survey from an operational research perspective. *OR Spectr.* **2020**, *43*, 1–58. [CrossRef]
11. Liu, Y. China Adapts Surveying, Mapping, Delivery Drones to Enforce World's Biggest Quarantine and Contain COVID-19 Outbreak | The Star Online. Available online: https://www.thestar.com.my/tech/tech-news/2020/03/05/china-adapts-surveying-mapping-delivery-drones-to-enforce-worlds-biggest-quarantine-and-contain-covid-19-outbreak (accessed on 23 March 2020).
12. Raptopoulos, A. No Roads? There's a Drone for That. Available online: https://amandacoxdesign.wordpress.com/2013/12/02/tedtalk-andreas-raptopoulosno-roads-theres-a-drone-for-that/ (accessed on 20 April 2019).
13. DHL | Press Release | English. Available online: https://www.dhl.com/en/press/releases/releases_2016/all/parcel_ecommerce/successful_trial_integration_dhl_parcelcopter_logistics_chain.html (accessed on 25 March 2020).
14. Tilley, A. UPS Experiments with Drone Delivery in Partnership with Zipline. Available online: https://www.forbes.com/sites/aarontilley/2016/05/09/ups-experiments-with-drone-delivery-in-partnership-with-zipline/?sh=609b416b45a4 (accessed on 29 December 2021).

15. Dragolea, N. 9 Drones That Will Revolutionise Healthcare-Doctorpreneurs. Available online: http://www.doctorpreneurs.com/9-drones-that-will-revolutionise-healthcare/ (accessed on 25 March 2020).
16. Drone First Aid Kit Stock Illustration 192927128. Available online: https://www.shutterstock.com/image-illustration/drone-first-aid-kit-192927128 (accessed on 29 December 2021).
17. Khan, S.I.; Qadir, Z.; Munawar, H.S.; Nayak, S.R.; Budati, A.K.; Verma, K.D.; Prakash, D. UAVs path planning architecture for effective medical emergency response in future networks. *Phys. Commun.* **2021**, *47*, 101337. [CrossRef]
18. Chung, S.H.; Sah, B.; Lee, J. Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Comput. Oper. Res.* **2020**, *123*, 105004. [CrossRef]
19. Escribano Macias, J.; Angeloudis, P.; Ochieng, W. Optimal hub selection for rapid medical deliveries using unmanned aerial vehicles. *Transp. Res. Part C Emerg. Technol.* **2020**, *110*, 56–80. [CrossRef]
20. Chauhan, D.; Unnikrishnan, A.; Figliozzi, M. Maximum coverage capacitated facility location problem with range constrained drones. *Transp. Res. Part C Emerg. Technol.* **2019**, *99*, 1–18. [CrossRef]
21. Adeleke, O.J.; Olukanni, D.O. Facility Location Problems: Models, Techniques, and Applications in Waste Management. *Recycling* **2020**, *5*, 10. [CrossRef]
22. El-Ghazali, T. *Metaheuristics: From Design to Implementation*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2009; ISBN 9780470278581.
23. Mixed-Integer Programming (MIP)-A Primer on the Basics-Gurobi. Available online: https://www.gurobi.com/resource/mip-basics/ (accessed on 29 December 2021).
24. Ilkhanizadeh, S.; Golabi, M.; Hesami, S.; Rjoub, H. The Potential Use of Drones for Tourism in Crises: A Facility Location Analysis Perspective. *J. Risk Financ. Manag.* **2020**, *13*, 246. [CrossRef]
25. Udroiu, R.; Deaconu, A.M.; Nanau, C.Ş. Data Delivery in a Disaster or Quarantined Area Divided into Triangles Using DTN-Based Algorithms for Unmanned Aerial Vehicles. *Sensors* **2021**, *21*, 3572. [CrossRef] [PubMed]
26. Deaconu, A.M.; Udroiu, R.; Nanau, C.Ş. Algorithms for Delivery of Data by Drones in an Isolated Area Divided into Squares. *Sensors* **2021**, *21*, 5472. [CrossRef] [PubMed]
27. Otero Arenzana, A.; Escribano Macias, J.J.; Angeloudis, P. Design of Hospital Delivery Networks Using Unmanned Aerial Vehicles. *Transp. Res. Rec.* **2020**, *2674*, 405–418. [CrossRef]
28. Lynskey, J.; Thar, K.; Oo, T.; Hong, C. Facility Location Problem Approach for Distributed Drones. *Symmetry* **2019**, *11*, 118. [CrossRef]
29. Shavarani, S.M.; Nejad, M.G.; Rismanchian, F.; Izbirak, G. Application of hierarchical facility location problem for optimization of a drone delivery system: A case study of Amazon prime air in the city of San Francisco. *Int. J. Adv. Manuf. Technol.* **2018**, *95*, 3141–3153. [CrossRef]
30. Kim, D.; Lee, K.; Moon, I. Stochastic facility location model for drones considering uncertain flight distance. *Ann. Oper. Res.* **2019**, *283*, 1283–1302. [CrossRef]
31. Figliozzi, M.A. Lifecycle modeling and assessment of unmanned aerial vehicles (Drones) CO2e emissions. *Transp. Res. Part D Transp. Environ.* **2017**, *57*, 251–261. [CrossRef]
32. Pirkul, H.; Schilling, D. The capacitated maximal covering location problem with backup service. *Ann. Oper. Res.* **1989**, *18*, 141–154. [CrossRef]
33. SPSS Statistics-Saudi Arabia | IBM. Available online: https://www.ibm.com/sa-en/products/spss-statistics (accessed on 29 December 2021).