Solving the probabilistic drone routing problem - searching for victims in the aftermath of disasters

Amadeu Almeida Coco, Christophe Duhamel., Andréa Cynthia Santos, Matheus Nohra Haddad

Received: 14 November 2022 | Revised: 15 January 2024 | Accepted: 26 January 2024

DOI: 10.1002/net.22214

SPECIAL ISSUE ARTICLE

WILEY

Solving the probabilistic drone routing problem: Searching for victims in the aftermath of disasters

Amadeu Almeida Coco[1] | **Christophe Duhamel**[1] | **Andréa Cynthia Santos**[2] | **Matheus Nohra Haddad**[1,3]

[1]LITIS, Université Le Havre Normandie, Le Havre, France
[2]LITIS, ISEL, Université Le Havre Normandie, Le Havre, France
[3]CRP-IEP, Universidade Federal de Viçosa, Rio Paranaíba, Brazil

Correspondence
Christophe Duhamel, LITIS, Université Le Havre Normandie, Le Havre, France.
Email: christophe.duhamel@univ-lehavre.fr

Funding information
Région Normandie

Abstract

Several major industrial disasters happen each year around the world. They usually involve limited accessibility, poor ground conditions, and toxic wastes. As a consequence, this reduces the efficiency of humanitarian operations. In such a context, flying drones may be a viable alternative: faster, no dependency on ground conditions, and larger areas scanned. They are also better suited for following the population and the crisis dynamic. For such a purpose, various issues have to be addressed such as defining and optimizing the drone's routes, their energy consumption, choosing the relay points for recharging equipment, among others. In this study, several additional features from existing works are considered: first, a probability of identifying individuals is defined. Thus, each node can be scanned several times in order to improve the observation. In addition, the nodes are prioritized according to a given heatmap. The probabilistic drone routing problem (PDRP) consists of finding a route, that is, a sequence of trips, for each drone such that the sum of the expected number of identified individuals on all routes is maximized. Constraints on energy consumption, collision avoidance and drone-base assignment are considered. We propose a heuristic and metaheuristics based on the adaptive large neighborhood search for the PDRP. The methods are tested on theoretical instances, as well as on a case study of the Beirut Port explosion on August 4, 2020, in order to analyze the performance of the proposed methods.

KEYWORDS

disaster logistics, drone routing problem, heuristics, search planning

1 | INTRODUCTION

Several major industrial disasters happen each year around the world. One example is the mining dam collapse near Brumadinho in Brazil, 2019: a tide of mud containing water and toxic mining residues rushed 80 km down a valley and resulted in about 270 casualties. Several critical issues were observed during the rescue operations. One of the most relevant was the complexity of performing the search for survivors and victims, as the accident prevented access by land vehicles to the affected area. Another limiting factor was the presence of toxic wastes, complicating human operations by requiring dedicated equipment. This kind of situation is typical of industrial or technological disasters as, for instance, in Fukushima, Japan, in 2011.

This study focuses on the use of a fleet of drones to perform the search and the identification of survivors and victims in an area impacted by an industrial disaster. The drones can be equipped with thermal or optical sensors, depending on the ground

and air conditions. Flying drones are considered for the search in case of major disasters since they are easy to deploy and cheaper to operate than classic means (helicopters, planes). They can move fast, they do not depend on ground and contamination conditions, and a single observation can scan large areas at once. Note that the methods proposed here could straightforwardly be adapted to the case of natural disasters since the main difference is the contamination by toxic wastes.

In terms of rescue operations, the search can be organized using a priority level assigned to each search region. This level is usually given by experts from civil security and fire brigade. Flying drones can follow the population and crisis dynamic which can change very fast. This can be done by a centralized or a decentralized control, with or without flying coordination between drones. Naturally, the coordination of a swarm of drones flights requires more energy consumption due to the communication requirements and the flight decision system. It is worth mentioning that civil drones have a limited autonomy. However, some research teams are already investigating civil drones with alternative energy sources like hydrogen. The models considered here address the drones' autonomy as an input parameter.

In the scientific literature, several studies are dedicated to the definition of drones' routes in case of disasters [4, 9, 10, 17, 19, 28, 31, 34, 39]. Most of these works address the problem as variants of the vehicle routing problem (VRP), where costs or distances optimization for the drone's routes are considered. Setting and organizing the relay points for recharging equipment, using load capacity, and other classical aspects are already addressed in VRP. For instance, studies [4, 9, 10, 17, 28, 39] have proposed approaches that aim to minimize costs in the post-disaster context. The main difference among these works lies in the circumstances under which cost minimization is applied. In [4, 9, 10] the damage is caused by an earthquake, an industrial disaster, and a flood, respectively. The authors in [17] introduce a multi-vehicle scheduling and routing problem to deliver relief packages after natural disasters. The study [28] develops a stochastic multi-vehicle approach to evaluate the state of route networks and to deliver relief packages. The goal of [39] is also to deliver relief packages, but with a focus on locations that can be reached only by drones. On the other hand, studies [19, 31, 34] deal with VRP variants with alternative objective functions. The authors of [19] map an affected area in the aftermath of an industrial accident to predict the distribution of hazardous substances across this region. Work [31] studies a routing problem with drones in order to minimize the maximum arrival time at any observation point. Study [34] presents a bi-objective multi-vehicle routing problem, where the goals are to maximize the amount of assessed (i) road segments and (ii) points of interest in a region damaged by a disaster.

This study considers new features compared to the existing work in the literature. The problem addressed here is not a classical or a known VRP variant for the following reasons: the location of survivors or victims is not known in advance. Instead, the geographical data is available, along with indications of potential locations, referred here to as heatmap. These indications may be inaccurate. A practical example of such a situation occurred after the dam collapse near Brumadinho, Brazil, in 2019, where inhabitants were displaced from their homes, and were found hundreds of meters or kilometers away from the priority search area. In addition, unlike a VRP with multiple node visits, the number of drone visits at each node is not fixed in advance and the quality of the node observation improves over the number of scans. Another point differing from existing studies in the literature is the coordination of drone flights, which takes into account rules to avoid collisions and additional damages.

The probabilistic drone routing problem (PDRP) is defined as follows. Given a heatmap and a complete simple graph associated with it, where each node is given a probability of identifying survivors or victims individually, the problem consists of defining routes for a set of drones in a search area such that the number of expected identified individuals is maximized. Constraints of energy consumption, collision avoidance and drone-base assignment are considered.

The reminder of this manuscript is organized as follows. Related works are discussed in Section 2. The PRDP is formally defined in Section 3, together with the way data is gathered and processed. The proposed heuristics and metaheuristics are detailed in Section 4. In Section 5, numerical experiments and analysis using theoretical instances, and a case study based on the explosion in Beirut Port on August 4, 2020, are given. Then, concluding remarks and perspectives are discussed in Section 6.

2 | LITERATURE REVIEW

Drones' operational and logistic planning raises many optimization problems. Among others in the scientific literature, several interesting surveys are available, such as [29, 32, 36, 49]. For instance, in [29] the use of drones is reviewed in the context of parcel delivery. In [32], the survey focuses on routing problems in logistics systems, and the authors highlight a lack of studies dedicated to the application of drones for food and mail deliveries. The authors in [36] survey more than 300 articles using Operations Research methods to solve problems related to the use of drones in civil applications such as post-disaster management [3, 28, 38, 43], agriculture [1, 47], and environmental monitoring [45, 54]. Study [49] analyzes 135 publications involving drones to solve variants of the well-known VRP [48].

According to the application and goals, the problems involved in the use of drones are associated with different optimization models. A significant part of the proposed problems consists of extensions of the VRP, collectively referred here to as the drone routing problem (DRP). DRP finds applications in post-disaster logistics [4, 9, 10, 31], military operations [15, 42],

entertainment and media [20, 37], among others. Survey [29] reviews 63 publications addressing DRPs applied to delivery. It is worth noting that DRPs are often employed in last-mile delivery [2, 24, 46], and for transporting small goods, like medical supplies [17, 21, 39, 51] or meals [27]. Recently, survey [32] reviews 101 studies, including the previous surveys [29, 36, 49], and proposes a structured and comprehensive framework for classifying drone-based optimization problems. It is worth mentioning other optimization problems for using drones in post-disaster management which do not rely on VRP such as: area coverage [33], path planning [3], and the integration of scheduling and routing problems [12]. Some of the main contributions using these kinds of approaches are detailed in Section 2.1. Other closely related studies and differences of our study from works published in the literature are given in Section 2.2.

2.1 | Dedicated drone optimization problems in crisis management

The pioneering DRP work [31] studies a post-disaster management problem, where a set of observation points in an area impacted by a catastrophe must be visited regularly during a given time period. The goal of this DRP is to find a solution that minimizes the maximum arrival time at any observation point, considering that each point is visited at least once and the number of available drones is limited. A constructive heuristic and two methods based on the variable neighborhood search are developed to solve this DRP. Computational experiments performed on a large set of theoretical instances indicate that all algorithms compute feasible solutions in less than one second, even for instances with 442 nodes, 6 recharging bases, and 9 drones, demonstrating the methods' efficiency.

The authors in [4] address a DRP in order to assess the damage caused by an earthquake. Given a set of locations and a drone fleet, this problem seeks to minimize the total route length needed to visit all sites with the available drones. They propose a genetic algorithm and a simulated annealing over a case study in Sicily, Italy. Experimental results show that the latter approach produces better solutions than the former in a significantly shorter time. Furthermore, it is demonstrated that increasing the fleet size results in larger times to compute a feasible solution, and does not lead to better solutions since the cost almost stabilizes when six or more drones are used to cover the affected region. The authors claim that having a larger fleet does not necessarily result in more efficient operations.

Study [34] presents a multi-vehicle routing problem that aims at assessing an area impacted by a disaster using both drones and motorcycles. Given a graph with a node-set, an edge-set, and a weight assigned to each node and each edge, the proposed bi-objective DRP seeks to maximize both the sum of the weights of (i) the evaluated nodes and of (ii) the visited edges, such that each vehicle has a limited range, performs a single trip, and visits a node at most once and an edge at most twice (one time in each direction). An ϵ-constraint method, a constructive heuristic based on the Dijkstra algorithm and four local searches for improving the heuristic solution are proposed for solving this problem. The proposed methods are tested on a realistic instance of the Kartal district in Istanbul, Turkey. Results indicate that a multi-vehicle approach in a post-disaster context contributes to the assessment of a larger area in a smaller time.

A DRP in which relief packages must be delivered to locations that cannot be reached by ground vehicles after a natural disaster is studied by [39]. The authors introduce a mixed integer linear programming formulation that seeks a solution with the minimal traveling cost, such that each drone has a limited range and payload, and recharging stations can be installed along the route to extend the drone's operating distance. This mathematical model is tested on small theoretical scenarios, demonstrating its applicability and potential.

Another DRP variant, where a fleet of gliders has to soar over a set of locations impacted by a flood, is introduced in [10]. The goal of this DRP variant is to minimize the total operation time, ensuring that only one route is assigned to each glider, and that each location is visited at least once. The authors propose two mixed integer nonlinear formulations. The first aims at minimizing the operation time, and the second seeks the solution with the smallest makespan. The two mathematical models differ on their respective objective functions and on an additional constraint-set in the latter. Both formulations are transformed into mixed-integer second-order cone programming problems, and tested on realistic instances based on risk maps of several United Kingdom cities.

The objective of the DRP presented in [19] is to map an affected area in the aftermath of an industrial accident, in order to predict the distribution of hazardous substances across this region. The authors address this problem by developing an exact algorithm based on dynamic programming and an adaptive large neighborhood search metaheuristic (ALNS). Computational experiments are performed on benchmarks introduced in [6, 11]. They indicate that the dynamic programming algorithm solves instances with up to 40 observation points to optimality, and that the metaheuristic returns better solutions than the algorithms proposed for the team orienteering problem [6] which are adapted to this problem.

The work [28] addresses a multi-vehicle stochastic routing problem in which drones are used to check the state of route networks with stochastic damage levels, and ground vehicles are employed to deliver relief goods to the affected area. A mixed integer nonlinear formulation is presented, aiming at coordinating the drones with the relief vehicles, while also minimizing the overall mission cost. The authors also propose a sequential heuristic and a genetic algorithm. Both methods are analyzed

in a case study based on the Haiti road network after the 2010 earthquake. Computational experiments compare the developed methods with a deterministic approach and a non-assisted relief delivery operation. They show that the former improves the vehicle travel time when compared to the latter.

Recently, the authors in [9] dealt with a heterogeneous fleet DRP that aims to minimize the cost of inspecting a disaster-affected area, ensuring that (i) each drone has a range and a recharge time; (ii) each drone route is given by a set of trips; (iii) each node is visited only once; and (iv) drones can recharge at any recharging base. A mixed integer linear programming formulation, an ALNS and a modified backtracking adaptive threshold accepting (MBATA) [9] are proposed. Computational experiments performed on a set of realistic instances based on a county in Mississippi State, United States, indicate that the mathematical model solves to optimality all instances in the small and medium instance-sets. MBATA outperforms ALNS, although both algorithms return feasible solutions for all instances in less than 30 min.

Study [33] introduces an area coverage problem for post-earthquake damage assessment using a drone fleet. The goal of this problem is to minimize the total operation time, ensuring that each location is visited exactly once and that the flight time does not exceed the drone's range. The authors propose an approach based on covering paths [18], where the instances correspond to maps transformed into grid graphs in which each node is represented by a square. Two linear formulations are developed, and their performance is assessed using the CPLEX solver. Both mathematical models have similar objective function and constraints, differing only in flow variables. The first formulation uses 5-index variables that consider the side of the square on which a drone leaves a node, while the second uses 4-index variables that do not take this aspect into account. Computational experiments indicate that the 5-index mathematical model outperforms the 4-index one on a set of theoretical instances. Study [3] investigates the use of drones for path planning problems in post-disaster and humanitarian contexts. This is solved using a partially observable Markov decision process [5] in which the data collected in previous observations is considered during the process of choosing the next location to be visited by a single drone. It is worth mentioning that constraints on drone flight duration are not considered.

Integrated optimization problems such as scheduling and routing in disaster relief [12], and using drones [17] have also been studied. A multi-vehicle scheduling and routing problem is studied in [17]. Ground vehicles and drones are used to deliver relief packages to demand points after a disaster, considering that a subset of roads is unavailable due to the disaster, and the drones are subject to range and payload constraints. A decision-support system (DSS) based on [35] is developed. It contains a mixed-integer programming formulation, an agent-based simulation algorithm, a constructive heuristic, and a Tabu search metaheuristic. Computational experiments are performed on realistic instances based on two regions along the Danube in Austria. They indicate that the DSS developed is a useful tool for helping decision makers in choosing the most efficient way to send relief goods to an impacted region after a natural or industrial disaster.

2.2 | Main model contributions

Unlike previous studies, the authors in [50] focus on data collection. They investigate the influence of the sensor data quality collected by drones, the energy limitations, the environmental hazards and the level of information exchange of a drone fleet during a search and rescue operation. They also assess the performance of three classes of search algorithms for a problem that aims at minimizing the time required to find a single missing target. Our work differs from [50] in three main aspects. The first difference is the context, since [50] deals with search and rescue operations in general, while our study focuses on the post-disaster context. The second one is the number of targets, as our problem seeks to locate multiple victims, whereas only one is sought in [50]. The last difference is the objective function, which in [50] is to minimize the operation time. Our work, on the other hand, aims at maximizing the expected number of target identifications.

As with our work, other studies, including the ones reviewed in Section 2.1 and summarized in Table 1, have addressed issues related to post-disaster relief and management. Table 1 is organized as follows. The first and second columns list the authors and the year of each publication, respectively. The third column identifies the problem addressed in each article, while the fourth column indicates the objective function of the problem. The methods used to solve each problem are mentioned in the fifth column. The last column shows whether the proposed strategies were tested on theoretical instances, on a realistic case study, or on a combination of both. One can observe that various researches have also considered limitations on the drone flight due to battery autonomy, a homogeneous drone fleet, recharging bases, and have presented computational experiments on theoretical and realistic instances. Yet, we provide additional and significantly different features in our approach. First, no work aims at maximizing the identification of missing targets, most of the previous studies minimize the operation costs, deliver relief packages or assess the damage on a disaster-affected area. Furthermore, no other DRP-related study allows multiple visits at an observation point and address anti-collision constraints. Therefore, it can be noted that our work addresses aspects that, as far as we know, have not been previously dealt with in the literature.

TABLE 1 An overview of Section 2, highlighting the main features of the PDRP and illustrating the differences between the PDRP and the related problems in the literature.

Authors	Year	Problem	Criteria	Method(s)	Case study
Waharte and Trigoni [50]	2010	Data collection	Operation time (min)	Heuristic	Theoretical
Mersheeva and Friedrich [31]	2012	Drone routing	Maximum arrival time at any observation point (min)	Heuristic, metaheuristic	Theoretical
Fikar et al. [17]	2016	Multi-vehicle routing and scheduling	Average time to deliver packages (min)	MILP model, simulation heuristic, metaheuristic	Danube river, Austria
Nedjati et al. [33]	2016	Drone area coverage	Total operation time (min)	MILP model	Theoretical
Cannioto et al. [4]	2017	Drone routing	Travel distance (min)	Metaheuristic	Sicily, Italy
Oruc and Kara [34]	2018	Drone routing	Assessed road segments and points of interest (max)	MILP model, exact, heuristic	Kartal, Turkey
Rabta et al. [39]	2018	Drone routing	Travel cost (min)	MILP model	Theoretical
Bravo et al. [3]	2019	Drone path	Search an entire affected area to find victims	Partially observable Markov decision process	Xanxeré, Brazil, camps, South Sudan, Fukushima, Japan
Coutinho et al. [10]	2019	Drone routing	Total operation time (min)	MINLP model	Various cities, United Kingdom
Glock and Meyer [19]	2020	Drone routing	Predict distribution of hazardous substances	Exact, metaheuristic	Theoretical
Macias et al. [28]	2020	Stochastic drone routing	Mission cost (min)	MINLP model, metaheuristic	Port-Au-Prince, Haiti
Chowdhury et al. [9]	2021	Drone routing	Inspection cost (min)	MILP model, metaheuristic	Mississippi, USA
Coco et al.	2023	Probabilistic drone routing	Expected number of target identifications (max)	Heuristic metaheuristic	Theoretical Beirut, Lebanon

3 | PROBLEM DEFINITION

The PDRP is formally defined as follows. The map of a region impacted by a disaster is divided into cells of a regular grid such that one cell can be fully scanned in one observation from a drone. Then, the grid is transformed into a complete graph $G = (N, E)$. Each cell corresponds to an observation node $c \in C$. Each base where the drones are operated from is a node $b \in B$. Thus $N = C \cup B$. Each arc from the arc set E corresponds to the direct drone flight from the origin node to the destination node. For each arc $(i, j) \in E$, e_{ij} is the energy consumption to travel from i to j and t_{ij} is the travel time from i to j. Both also include the time and energy consumption for a single observation when $j \in C$ since at least one observation must be done at j. An identical flying speed is assumed for all the drones. Consecutive observations can be done by staying at the node, that is, using the arc (i, i).

The fleet D of drones is homogeneous. Each drone $d \in D$ is assigned to a pre-defined base $b_d \in B$. We consider the drone range R as its total amount of energy. It does not allow routes longer than the total duration H (the time horizon is H). A drone route is a sequence of several trips starting and ending at its base, where it performs its recharge. A fixed duration T_R is set for the recharge operation, and T_O for a drone observation. The energy consumption for one observation is E_O.

For each observation node $c \in C$, let Q_c be the number of targets to identify and let p_c be the probability of any target identification in a single observation. The random variable X_c measures the number of targets identified in one drone observation. Assuming independence for the identification of each target, X_c follows a binomial distribution $\mathcal{B}(Q_c, p_c)$ whose expected value is $\mathbb{E}[X_c] = Q_c p_c$. Since the drones are allowed performing several observations, let X_c^k be the random variable for measuring the number of new targets identified at observation k and let S_c^k be the total number of targets identified up to observation k. Thus, X_c^k corresponds to targets unidentified so far, as they were not identified in the previous $k-1$ observations.

Since the independence of consecutive observations might be a strong assumption in many contexts, we associate a weight factor $\alpha_c \in (0, 1)$ to the probability p_c such that the probability of individual identification in observation k becomes $p_c^k = (\alpha_c)^{k-1} p_c$. Small values for α_c correspond to a strong correlation, while high values correspond to a low correlation.

X_c^k follows a binomial distribution $\mathcal{B}(Q_c - S_c^k, p_c^k)$ and, under the assumptions and definitions, the expected value of X_c^k and S_c^k can be computed using the recursive formulas

$$\mathbb{E}[S_c^0] = 0,$$
$$\mathbb{E}[X_c^k] = (Q_c - \mathbb{E}[S_c^{k-1}])p_c^k,$$
$$\mathbb{E}[S_c^k] = \mathbb{E}[S_c^{k-1}] + \mathbb{E}[X_c^k].$$

Given a set R of drone routes containing the sequence of nodes and the number of consecutive observations at each node, one can compute the total number n_c of observations at node $c \in C$. Its expected number of targets identified is $\mathbb{E}[S_c^{n_c}]$.

The PDRP consists in finding a route for each drone maximizing $\sum_{c \in C} \mathbb{E}[S_c^{n_c}]$, the sum of the expected values for the number of targets identified on all routes. The energy consumption of each trip r is defined as the sum of the energy consumption for each edge $(i,j) \in r$ (travel and observation). It cannot exceed R. Moreover, every drone must return to its base no later than time H. Finally, at least one trip is assigned to each drone, and a node can belong to several routes, as it can be visited several times to improve the expected number of targets identified at this node. Moreover, in order to avoid drone collision, a node cannot be visited by two drones at the same time.

The collision constraint is defined as follows: let a drone $d \in D$ performing a trip r. It leaves its base b_d at time s_r and returns to it at time e_r. Then any drone $d' \neq d$ is prohibited from visiting any node $c \in r$ in the time interval (s_r, e_r).

As opposed to the VRP, the route length is not explicitly minimized in the PDRP. However, given a set of visited nodes in a route and a time limit, the less time spent on the drone moves, the more time used in observations. Thus, the larger the number of targets identified.

Figure 1 depicts the process of transforming the map of a region impacted by a disaster into a complete graph and a heatmap. After receiving the data from the catastrophe, the map of the affected area (Figure 1A) is covered by discs as illustrated in Figure 1B. These discs correspond to the area scanned by a drone in one observation, given its operational flight height h and its camera angle θ. Thus, their radius is $r = h \tan \theta/2$ which leads to the length l of the inscribed square $l = r\sqrt{2}$. Each disk is then converted into a $l \times l$ m^2 cell (Figure 1C). Given the way the discs are set on the map, the squares cover the area. It is worth mentioning that r may be set to a smaller value than the possible one, in order to prevent possible distortions and error propagation [52] at the border, which can lead to incorrect assumptions about a cell. The resulting grid graph is then transformed into an instance (Figure 1D) by computing the flight time between the center of each pair of neighboring cells and choosing the bases $b \in B$. Finally, the graph is transformed into a complete graph (Figure 1E), by computing the flight time between each pair of nodes $(i,j) \in E$, respecting the grid graph constraints (for instance some areas unsuitable for flying over). The heatmap is built on top of the grid graph by estimating the number of targets to identify at each node $c \in C$ (Figure 1F).

Figure 2 illustrates an example containing a complete graph with 9 nodes and 36 edges. For the sake of clarity, only edges that belong to at least one route are shown. Let us consider a unique base $B = \{0\}$ and a single drone $|D| = 1$. The time horizon is set to $H = 70$, the drone range is $R = 30$, the drone recharge time is $T_R = 10$ and the observation time is $T_0 = 1$. For sake of simplicity, $e_{ij} = t_{ij}\ \forall (i,j) \in E$. One can note that the drone makes at least two trips during the time horizon (when fully used). The flight duration is shown for each arc and the number of consecutive drone observations is shown for each visited node.

The first trip is displayed on Figure 2A. The drone starts from the base and goes to node 6 where it performs five scans. It travels to node 7 for three observations, before flying to node 4 to do three further scans. Then the drone returns to the base to recharge. The first trip consumes $2.021 + 5.000 + 9.517 + 3.000 + 2.532 + 2.000 + 5.820 = 29.890$ time units. After recharging, the drone starts its second trip, shown in Figure 2B. It flies to node 1 for one single observation. It then goes to node 2 for six scans, and then to node 7 again, taking only one scan. As both the time horizon and its battery are ending, it returns to the base. The second trip takes $7.636 + 1.000 + 3.226 + 6.000 + 2.729 + 1.000 + 8.191 = 29.782$ time units. It is worth noting that node 7 is visited during both trips. Unlike most VRPs and DRPs, a drone is allowed visiting a node more than once if necessary. The total operation time is $29.890 + 10.000 + 29.782 = 69.672$ time units.

4 | METHODS FOR SOLVING THE PDRP

In this section, the algorithms developed to solve PDRP are presented. A greedy constructive heuristic (GCH) is presented in Section 4.1. In the sequel, two methods based on ALNS metaheuristic are described in Section 4.2.

4.1 | Greedy constructive heuristic

The GCH is a parallel insertion heuristic based on the well-known nearest neighbor heuristic for VRP [48]. In GCH, all drone routes are initially available for scan or node insertions, and, at each iteration, a single operation is added for only one drone. Each iteration of GCH works as follows. First, the drone with the shortest route, in terms of time, is selected. In the case of a tie, the drone with the lowest index is chosen. Subsequently, the following operations are assessed: (i) adding a scan to the current node i; (ii) moving the drone to another node j and performing one observation on it; and (iii) returning the drone to the recharging station. Moves (i) and (ii) consider the time and the energy to go back to the recharge station afterwards. Thus, the last move (iii) is always feasible and it is made when the drone cannot move to another node or scan the current one again, due to range or time horizon constraints. Collision is also checked for moves of type (i) and (ii). It is worth mentioning that GCH

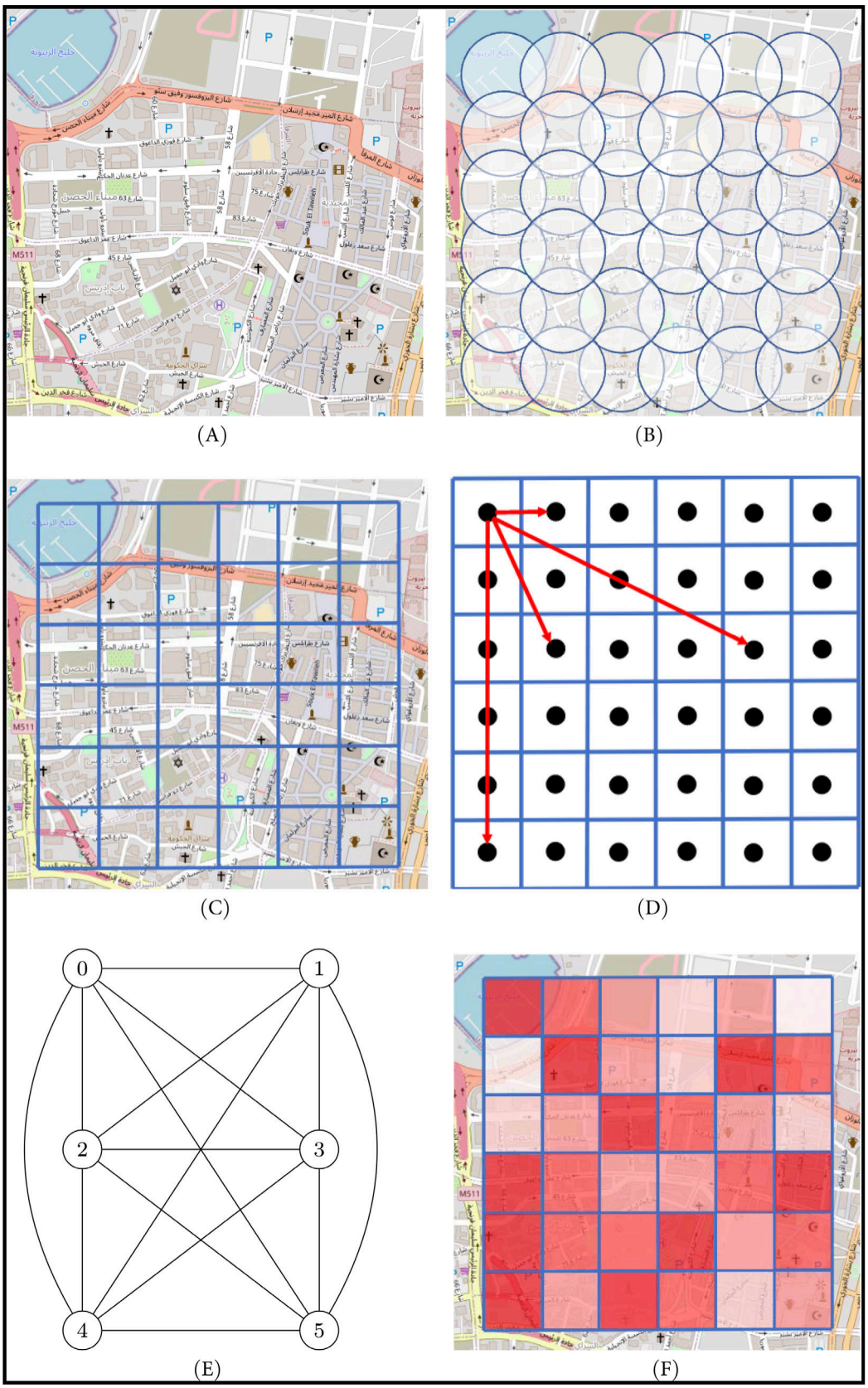

FIGURE 1 (A) Map of a disaster area; (B) map covered by circles, that is, areas captured by the camera of a drone; (C) grid graph obtained after dividing the zone of interest; (D) an instance derived from the original map; (E) a complete graph obtained from the instance; (F) instance heatmap.

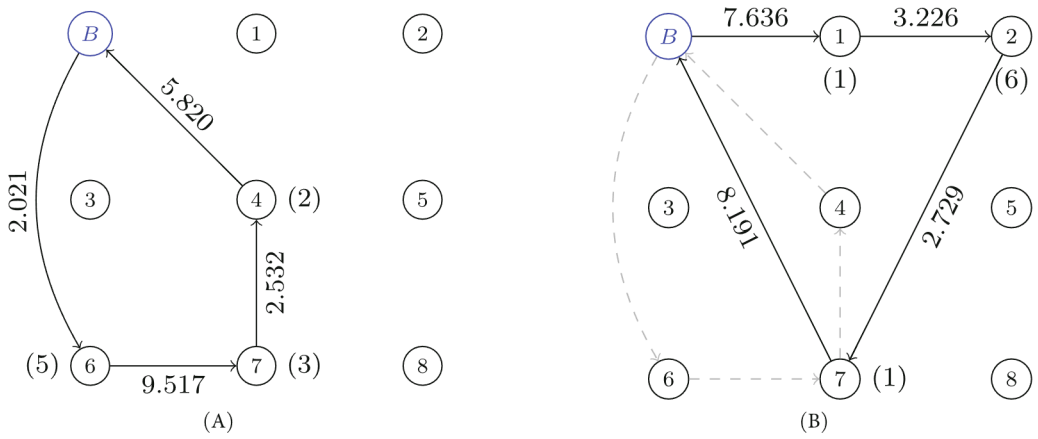

FIGURE 2 (A) Drone first trip; (B) drone second trip.

move selection is given by picking the node j with the highest ratio $\tau_{ij} = \frac{\mathbb{E}[X_j^{o_j+1}]}{t_{ij}}$, where o_j is the number of scans already done at node j, when going from node i to node j. Thus τ_{ij} is a ratio between the expected number of new identifications and the time required. Furthermore, note that GCH is allowed adding new observations at the current node i until a move to another node becomes more attractive, as t_{ii} includes the scanning time. After all operations have been assessed, the best one is selected and included at the end of the drone route. The algorithm ends when the time horizon is reached for all drones.

The pseudocode of GCH is shown in Algorithm 1. This method uses G, B, C, R, H, defined in Section 3, as input. It returns a solution s containing the routes r_d for each drone $d \in D$. First, it is worth noting that each drone is initially located at its predefined base $b_d \in B$. In the loop in lines 1–4, r_d and the time h_d in which the last flight ends are set to b_d and 0 respectively for each drone $d \in D$. Then, the number o_c of scans done at each node $c \in C$ is set to 0 in the loop in lines 5–7. In line 8, the current time h is set to 0. Thereafter, the loop in lines 9–24 is repeated until H is reached. In line 10, the drone $d \in D$ with the smallest h_d is selected and assigned to the variable d'. The set Γ of all nodes that can be reached from node $last(d') \in N$ where d' is currently located, without breaking the range, the collision and the time horizon constraints, is built (line 11). If $\Gamma \neq \emptyset$ (line 12), the node $j \in \Gamma$ with the highest τ_{ij} is chosen (line 13) and inserted at the end of r_d (line 14). Variables o_j and h_d are updated in lines 15 and 16, respectively. Otherwise, $\Gamma = \emptyset$ and the drone has to go back to its base for recharging. If the drone is not already at its base, b_d is inserted at the end of r_d (line 18) and the time between i and the recharge base is added to h_d (line 19) in addition to the recharge time. Otherwise, the drone is at its base and cannot visit other nodes anymore. It is then removed from the set of active drones in line 21. At the end of each iteration, the value of h is updated (line 23). In the loop in lines 26 and 27, each drone route r_d is inserted in s. Finally, s is returned in line 29.

4.2 | Adaptive large neighborhood search

The ALNS is a metaheuristic introduced in [41] for the pickup and delivery problem with time windows. ALNS is widely used in the VRP literature. Several problems, such as the pollution-routing problem [13], the electric fleet size and mix VRP with time windows and recharging stations [23], the two-echelon VRP [22] and the service technician routing and scheduling problem [25], have been solved using ALNS. It has also been applied to address DRPs in various studies, such as [8, 26, 30, 44, 53]. Recently, [9, 19] developed ALNS-based algorithms for dealing with DRPs with applications in post-disaster management.

An ALNS is usually divided into three phases. First, an initial solution s_0 is computed by a constructive heuristic and set as both the current solution s and the incumbent solution s^*. Then, a sequence of destruction/repair operations is applied in order to improve s. During this phase, s^* is updated whenever the cost of s is better than that of s^*. After a predefined criterion is met, s is set as s^* and a shaking procedure partially modifies s, generating a new current solution. The main goal of this third phase is to avoid being trapped in local optima. Next, the sequence of destroy/repair iterations and shaking is repeated until a stopping criterion is met, as shown in Figure 3. At the end of ALNS, the incumbent solution s^* is returned. The vector *score* keeps the number of successes of each operator in improving the current solution. It provides the information for the adaptive mechanism in ALNS.

The ALNS for PDRP first uses GCH to get an initial solution. The destruction/repair step uses six removal and five insertion operators, respectively. For the former, four operators remove a node from each trip in a route, according to the following criteria: (i) a random node, (ii) the node with the highest sum of the travel times from its predecessor and to its successor, (iii) the most scanned, and (iv) the one with the smallest expected number of identified targets. The two other operators remove, respectively, a random number of scans and half the observations from each node in a route, while keeping at least one scan for each node. For the latter, three operators insert nodes into a route using the criteria: (i) the highest expected number of

Algorithm 1. GCH pseudocode

Input: G, B, C, R, H, D
Output: s
foreach $d \in D$ **do**
  $r_d \leftarrow \{b_d\}$
  $h_d \leftarrow 0$
end
foreach $c \in C$ **do**
  $o_c \leftarrow 0$
end
$h \leftarrow 0$
while $h < H$ **do**
  $d' \leftarrow \arg\min_{\{d \in D\}} h_d$
  $\Gamma \leftarrow ReachableNodes(d', last(d'), h_{d'}, C, R, H)$
  if $\Gamma \neq \emptyset$ **then**
    $j \leftarrow \arg\max_{\{j \in \Gamma\}} \tau_{ij}$
    $r_d \leftarrow r_d \cup \{j\}$
    $o_j \leftarrow o_j + 1$
    $h_d \leftarrow h_d + t_{ij}$
  else if $last(d') \neq b_d$ **then**
    $r_d \leftarrow r_d \cup \{b_d\}$
    $h_d \leftarrow h_d + t_{ib_d} + T_R$
  else
    $D \leftarrow D \setminus \{d\}$
  end
  $h \leftarrow \min_{\{d \in D\}} h_d$
end
$s \leftarrow \emptyset$
foreach $d \in D$ **do**
  $s \leftarrow s \cup r_d$
end
return s

targets to be identified in the next scan, (ii) the largest ratio τ_{ij} as defined in Section 4.1—node j being inserted after node i- and (iii) the highest expected number of targets not yet identified. Nodes are always inserted after the last observation of the previous node. The other two insertion operators add scans to some nodes already visited. In this case, new observations are added, respectively, to nodes i with the maximum ratio τ_{ii} and to the ones with the highest expected number of targets not identified.

At each iteration, a removal and an insertion operator are randomly chosen according to an adaptive criterion that considers which operators led to solution improvements in previous iterations. In addition, two shaking procedures are defined. The first one randomly replaces some trips with new ones generated using GCH. The second one rebuilds a route such that all scans for a node i are done sequentially, that is, the drone never returns to i after leaving it. These shakings are performed in sequence. Only feasible solutions after the repair, in terms of range, time horizon, collision and assignment to a base, are taken into account. Thus, the destruction/repair step aims at performing local modifications in trips, while shakings have a more global impact on the set of trips.

The anti-collision system works as follows: all nodes, except the bases, are initially set as available. When a drone $d \in D$ is selected to visit a node $n \in N$, n becomes assigned to d. Consequently, no other drone is allowed to access n during the current trip of d. Once d returns to its base, all nodes that have been visited during the trip are unassigned and become available again. This locking/unlocking mechanism prevents ensures collision-free operations.

Algorithm 2 presents the pseudocode of ALNS. It uses G, B, C, R, H as input and it returns the incumbent solution s^*, which is a route-vector with all drone routes R_d, $d \in D$. GCH is called line 1, returning the initial solution s which is stored in s^* in line 2. The variable i, which counts the number of iterations without solution improvement, is initialized in line 3. Lines

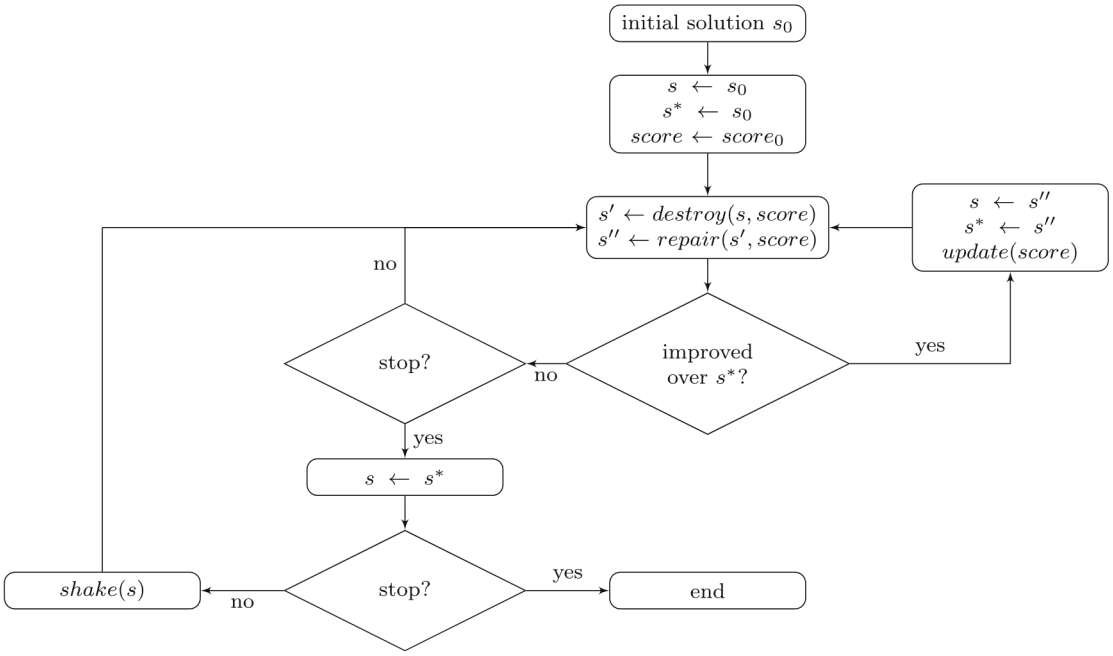

FIGURE 3 Generic ALNS flowchart.

4 and 5, two arrays w_r and w_a are created and initialized to 0. Their size is respectively equal to the number of removal and insertion operators. The main purpose of these arrays is to keep the probability of each operator being selected. This probability is computed by dividing the number of times the application of an operator led to an improvement in s^* by the total number of times s^* has been updated. At each iteration, a removal and an insertion operator are randomly selected according to w_r and w_a. Thus, a more successful operator is likely to be selected more frequently. The loop in lines 6–34 is performed until no solution improvement is found after *it_ALNS* iterations. The variable *j*, which counts the number of local iterations without solution improvement, is initialized in line 7. The loop in lines 8–23 is repeated until the maximum number *it_LS* of iterations without any improvement is reached. In line 9, a removal operator σ_r is randomly chosen according to the adaptive criterion previously explained. Then, σ_r is applied to s in line 10, generating a partial solution s'. In line 12, an insertion operator σ_a is randomly selected, also following the adaptive criterion. Subsequently, σ_a is used to s' in line 13, producing a new solution s''. These two operations are repeated until s'' is feasible. If the value of s'' is better than the one of s (line 15), the variables s, $w_r[\sigma_r]$ and $w_a[\sigma_a]$ are updated in lines 16–18, while j is reset line 19. Otherwise, the value of j is increased line 21. After the inner loop, if the value of s is better than the one of s^* (line 24), s^* is updated in line 25 and i is reset in line 26. Otherwise, the value of i is increased (line 28). The shaking procedures are applied to s^* and the resulting solution is stored in s lines 31 and 32. The shaking is repeated until s is feasible. Finally, the incumbent solution s^* is returned in line 35.

An extended version of ALNS, referred to here as E-ALNS, is also proposed. It consists in adding a sequence of additional moves after the ALNS destroy/repair loop. Its aim is to further intensify the solution search. These new moves consist of two swap approaches based on the well-known *k*-opt move. In the first swap, each pair of nodes belonging to the same trip is a candidate for switching. The feasible swap with the best evaluation is selected. The second approach focuses on the scans: the amount of scans for each pair of nodes within the same trip is considered for swapping. Then, the node sequence is not modified. The feasible neighbor with the best evaluation is chosen. The best solution returned by these methods is then compared with the incumbent and the better one is retained. It is worth mentioning all the moves are done in order to increase the time dedicated to the drone scans since this directly impacts the objective function. Thus, E-ALNS is obtained from ALNS by inserting the following two lines between lines 23 and 24:

$$s \leftarrow nodeSwap(s),$$
$$s \leftarrow scanSwap(s).$$

5 | NUMERICAL EXPERIMENTS

The computational experiments are carried out on an Intel Core i7-1165G7 with 2.80 GHz clock and 32 GB of RAM, running Ubuntu Linux 20.04 LTS. GCH, ALNS and E-ALNS are implemented in C++ and compiled with GNU g++ 9.3.0. Five

Algorithm 2. ALNS pseudocode for the PDRP

Input: G, B, C, R, H
Output: s^*
$s \leftarrow GCH(G, B, C, R, H)$
$s^* \leftarrow s$
$i \leftarrow 0$
$w_r \leftarrow 0$
$w_a \leftarrow 0$
while $i < it\_ALNS$ **do**
  $j \leftarrow 0$
  while $j < it\_LS$ **do**
    $\sigma_r \leftarrow chooseRemovalOperator(w_r)$
    $s' \leftarrow applyRemovalOperator(\sigma_r, s)$
    do
      $\sigma_a \leftarrow chooseInsertionOperator(w_a)$
      $s'' \leftarrow applyInsertionlOperator(\sigma_a, s')$
    while s'' *not feasible*;
    if $cost(s'') > cost(s)$ **then**
      $s \leftarrow s''$
      $w_r[\sigma_r] \leftarrow w_r[\sigma_r] + 1$
      $w_a[\sigma_a] \leftarrow w_a[\sigma_a] + 1$
      $j \leftarrow 0$
    else
      $j \leftarrow i + 1$
    end
  end
  if $cost(s) > cost(s^*)$ **then**
    $s^* \leftarrow s$
    $i \leftarrow 0$
  else
    $i \leftarrow i + 1$
  end
  do
    $s \leftarrow randomTripReplacement(s^*)$
    $s \leftarrow routeRebuild(s)$
  while s *not feasible*;
end
return s^*

runs are done for each instance with different seeds for the pseudorandom number generator for both ALNS and E-ALNS in all experiments, and the average values are reported. Section 5.1 presents the computational tests on theoretical instances. Computational experiments on a case study of the Beirut Port Explosion, in Lebanon, on August 4, 2020 [7], are described in Section 5.2.

For all the instances, the individual probability of identification p_c for each node $c \in C$ is arbitrarily chosen in the interval $(0, 1]$. The stopping criteria it_{ALNS} and it_{LS} are set to respectively 20 and 1 iterations without improvement on the incumbent solution. The weight factor α_c is set to 1. Both E_O and T_O are set to 1 and T_R is instance-dependent.

5.1 | Theoretical instances

The theoretical instance set contains 20 grid graphs with up to 900 nodes. Time horizons of 300, 900, 3600, and 7200 time units are considered (they correspond to 5, 15, 60, and 120 min respectively). The instances contain up to four drone bases, located

TABLE 2 Comparing GCH, ALNS, and E-ALNS on the theoretical instance-set.

Instance							GCH		ALNS		E-ALNS	
ID	$\|N\|$	H	$\|D\|$	R	T_R	Q	% scan	Time	% scan	Time)	% scan	Time
$t-1$	25	300	1	100	100	314	87.83	0.01	**94.00**	0.08	**93.45**	0.10
$t-2$	25	300	1	100	100	330	73.84	0.01	83.47	0.06	83.89	0.07
$t-3$	25	300	2	100	100	306	**99.17**	0.01	**99.95**	2.23	**99.95**	2.16
$t-4$	25	300	2	100	100	310	**96.46**	0.01	**99.56**	2.30	**99.62**	1.79
$t-5$	100	900	1	300	300	1338	79.10	0.01	80.13	0.21	80.60	0.31
$t-6$	100	900	1	300	300	1481	72.80	0.01	77.43	0.35	77.49	0.48
$t-7$	100	900	2	300	300	1316	**95.57**	0.02	**98.28**	1.42	**98.29**	1.31
$t-8$	100	900	2	300	300	1481	**93.55**	0.02	**96.51**	0.93	**96.38**	0.85
$t-9$	400	3600	1	900	900	5821	50.83	0.07	54.66	1.34	55.30	6.25
$t-10$	400	3600	1	900	900	5894	52.52	0.07	56.23	2.11	56.80	6.58
$t-11$	400	3600	2	900	900	5810	81.90	0.14	86.36	5.45	86.89	14.51
$t-12$	400	3600	2	900	900	5883	79.68	0.13	83.90	5.15	84.70	13.40
$t-13$	400	3600	4	900	900	5796	**97.32**	0.33	**99.69**	19.52	**99.69**	20.88
$t-14$	400	3600	4	900	900	5845	**97.06**	0.34	**99.21**	14.48	**99.18**	17.31
$t-15$	900	7200	1	900	900	13 170	56.88	0.37	58.40	5.94	58.53	58.97
$t-16$	900	7200	1	900	900	13 447	58.61	0.38	60.45	7.91	60.32	55.20
$t-17$	900	7200	2	900	900	13 140	85.47	0.83	86.60	26.75	87.10	82.75
$t-18$	900	7200	2	900	900	13 429	85.26	0.85	86.83	26.75	87.35	79.70
$t-19$	900	7200	4	900	900	13 099	**97.89**	1.95	**99.78**	356.78	**99.76**	369.72
$t-20$	900	7200	4	900	900	13 397	**97.91**	1.99	**99.63**	412.19	**99.64**	316.69
Average							81.98	0.38	85.05	44.45	85.25	52.45

at the grid corners. Ranges of 100, 300, and 900 time units are taken into account. For each node $c \in C$, the number of targets Q_c is randomly drawn using a uniform discrete distribution $U[0, 30]$.

The performance of GCH, ALNS and E-ALNS are assessed on this set and the results are reported in Table 2. The first six columns show, respectively, the instance identifier *ID*, the number $|N|$ of nodes, the time horizon H, the number $|D|$ of drones, the range R, the recharge operation T_R, and the number Q of targets for each instance. The average rate (*%scan*) of expected identified targets and the average running time (*time*) in seconds are shown for each method in columns 7 to 12. Bold values mean at least 90% of the targets are identified in the solution.

Results indicate that, on average, GCH achieves an expected location of almost 82% of the targets, while both ALNS and E-ALNS obtain an expected identification of approximately 85% of the targets. Furthermore, all methods are able to solve the instances in reasonable times, with GCH running in less than two seconds. The slowest runs of ALNS and E-ALNS take 7 min, on average. One can also observe that ALNS and E-ALNS improvement over GCH can be up to 10.05%. It can be explained by the fact that the objective function in PDRP is impacted by the number of node visits, and not directly by the chosen arcs as it is the case for VRPs. Thus the order the nodes are visited does not count except that it may take more time and then limit the number of nodes scanned in a trip due to the range constraint. A solution from a heuristic might have less time available for node scans due to myopic arc selection, which means some nodes will be less scanned. This removes the gains in identification from the last observations only, which are significantly less than the gain in the first scans. By computing a better sequence of node visits, solutions in ALNS are given more time for node scans and thus can add some more observations at nodes, whose benefit is lower than the initial observation. Although slower than GCH, the use of ALNS and E-ALNS is justified in the post-disaster context, since an increase of approximately 3% in the average outcome can be translated into the localization of tens or hundreds of survivors and victims. It means that additional time spent running ALNS or E-ALNS may lead, for example, to a faster rescue or relief operation. It is also worth mentioning that fleets with at most four drones are able to scan at least 90% of the targets in instances with up to 900 vertices, demonstrating the efficiency of the heuristics on this instance set.

Note that further improvements on the route length, and thus on the number of observations, could be made by allowing the drones to perform the observations from a given range, for instance from the neighbor nodes. This feature is modeled in the close-enough TSP, for which efficient approaches have been proposed in [14].

5.2 | Case study: Beirut port explosion

On August 4, 2020, around 17:45 (local time), a huge explosion happened in the Port of Beirut, Lebanon [7]. The explosion was caused by a fire that broke out at a warehouse containing a large amount of ammonium nitrate, which had been confiscated

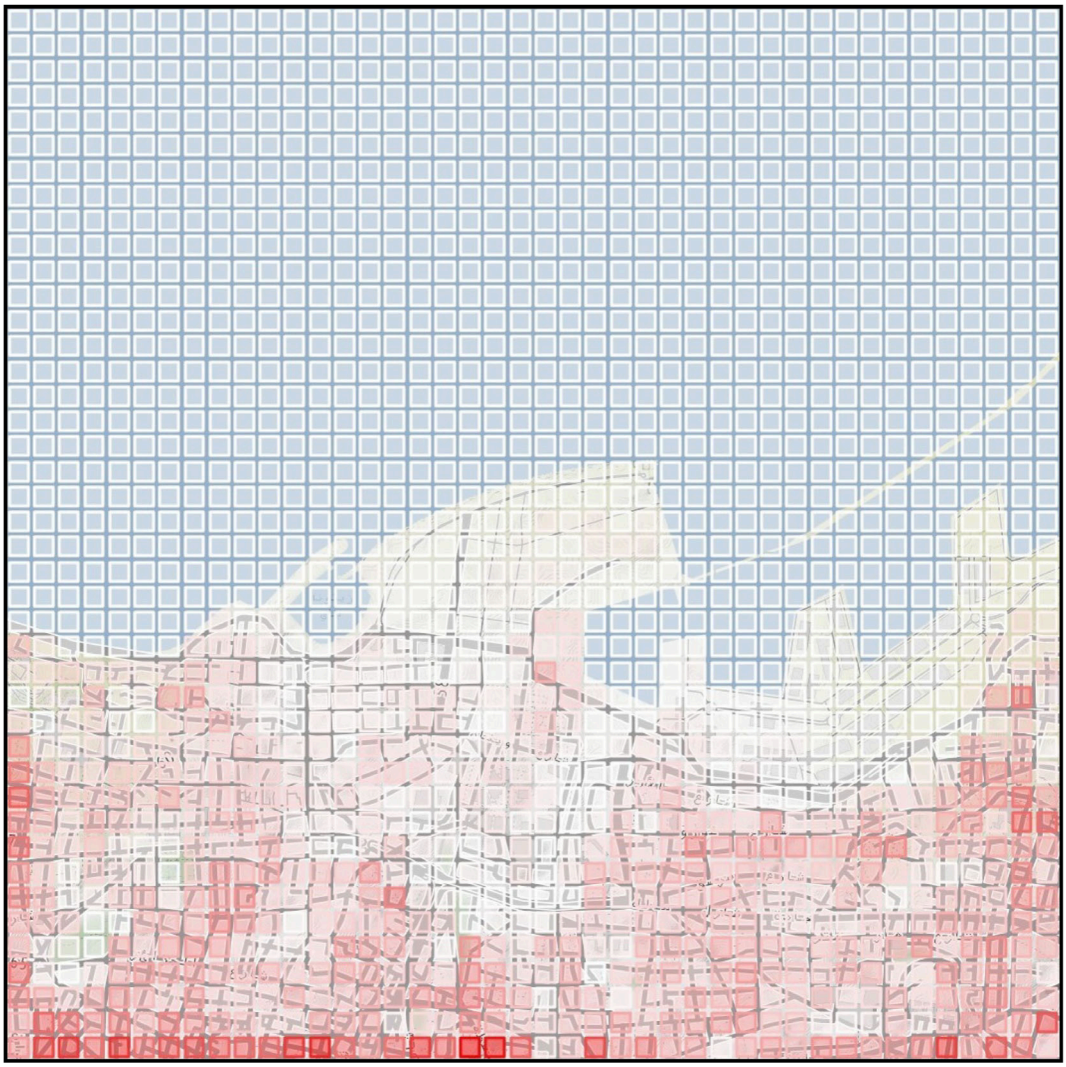

FIGURE 4 Map of the region around the port of Beirut, Lebanon, before removing the water tiles.

by the Lebanese authorities and was being stored without proper safety measures. The blast was so powerful that it shook the whole country of Lebanon. It was also felt in other Middle East countries, as well as parts of southeastern Europe. According to [7], the burst produced a magnitude 3.3 seismic shock. It is considered one of the most powerful accidental and non-nuclear explosions in world history. This catastrophe resulted in approximately 218 casualties, 7000 injuries, and fifteen billion dollars in infrastructure damage. Furthermore, it left an estimated 300 000 people homeless.

The second instance set contains realistic instances built from the map of the region around the port of Beirut shortly after the 2020 explosion. We assume that all existing structures within a 3 km radius of the blast focal point were affected, as displayed in Figure 4. Following the process of transforming a map into an instance transformation process as described in Section 3, the damaged area map is divided into 100×100 m^2 square cells. The color intensity corresponds to the value of the heatmap in each cell. It is computed as the amount of buildings in each cell. As can be seen in Figure 4, the sea takes up a large section around the focal point. Thus, a preprocessing is done on the original map to remove all water tiles. This results in a complete graph with 1050 nodes (approximately a 42×25 grid), presented in Figure 5. For each instance, the number of targets Q_c of each cell c is randomly drawn using the uniform distribution $U(0, 20)$ times the number of buildings in the cell.

The first experiment on these realistic instances focuses on analyzing the impact of the time horizon as well as the number of drones and bases on finding more targets. The results are reported in Table 3. The first seven columns display the instance identifier *ID*, the number $|N|$ of nodes, the time horizon H, the number $|B|$ of bases, the number $|D|$ of drones, the range R, the recharge operation T_R, and the number Q of targets for each instance, in that order. Columns 7 through 12 present, for each method, the average percentage of expected identified targets (*%scan*) and the average running time (*time*) in seconds. Bold values mean at least 90% of the targets are identified in the solution.

Results point out that, on average, GCH obtains an expected identification of approximately 68% of the targets, while both ALNS-based approaches achieve an expected identification of almost 72.5% of the targets. It is also worth noting that, on

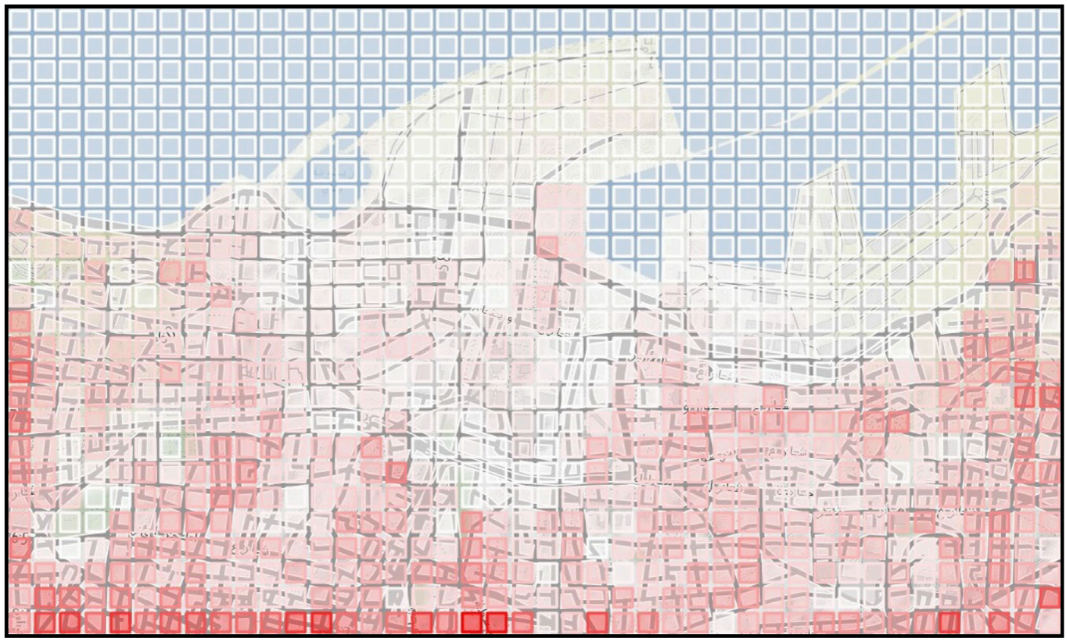

FIGURE 5 Map of the region around the port of Beirut, Lebanon, after deleting the squares containing only water.

TABLE 3 Comparing GCH, ALNS, and E-ALNS on the realistic instance-set based on the Beirut Port explosion.

Instance								GCH		ALNS		E-ALNS	
ID	$\|N\|$	H	$\|B\|$	$\|D\|$	R	T_R	Q	% scan	Time	% scan	Time	% scan	Time
$r-1$	1050	3600	1	1	900	900	29 513	18.96	0.20	21.93	1.92	21.14	1.52
$r-2$	1050	3600	1	1	900	900	30 004	20.21	0.16	20.70	1.30	20.44	0.88
$r-3$	1050	7200	1	1	900	900	29 513	31.23	0.18	33.04	5.39	33.18	5.03
$r-4$	1050	7200	1	1	900	900	30 004	30.83	0.19	33.17	6.21	32.47	2.33
$r-5$	1050	3600	2	2	900	900	29 829	40.22	0.21	43.31	3.29	43.90	4.42
$r-6$	1050	3600	2	2	900	900	29 869	42.45	0.19	44.07	2.13	44.77	3.18
$r-7$	1050	7200	2	2	900	900	29 829	60.33	0.20	68.70	31.94	69.24	24.40
$r-8$	1050	7200	2	2	900	900	29 869	62.96	0.19	66.74	16.58	66.80	18.25
$r-9$	1050	3600	2	4	900	900	29 829	60.02	0.19	69.71	34.33	69.19	28.06
$r-10$	1050	3600	2	4	900	900	29 869	61.78	0.19	68.30	39.14	67.84	23.64
$r-11$	1050	7200	2	4	900	900	29 829	83.03	0.23	88.11	92.30	88.20	99.97
$r-12$	1050	7200	2	4	900	900	29 869	80.93	0.23	86.83	84.25	87.35	118.83
$r-13$	1050	3600	4	4	900	900	29 881	60.39	0.19	70.23	23.47	70.01	25.32
$r-14$	1050	3600	4	4	900	900	29 677	61.82	0.18	68.31	35.92	67.03	19.33
$r-15$	1050	7200	4	4	900	900	29 881	85.56	0.25	89.79	97.74	89.91	151.53
$r-16$	1050	7200	4	4	900	900	29 677	82.28	0.24	88.29	110.61	88.51	134.38
$r-17$	1050	3600	4	8	900	900	29 881	85.55	0.26	**90.09**	112.13	**90.21**	135.80
$r-18$	1050	3600	4	8	900	900	29 677	80.47	0.23	89.06	111.87	88.79	110.04
$r-19$	1050	7200	4	8	900	900	29 881	**96.99**	0.34	**99.74**	784.20	**99.73**	779.28
$r-20$	1050	7200	4	8	900	900	29 677	**96.15**	0.34	**99.37**	668.17	**99.39**	844.11
$r-21$	1050	3600	4	16	900	900	29 881	**96.11**	0.30	**99.30**	422.31	**99.31**	534.86
$r-22$	1050	3600	4	16	900	900	29 677	**95.68**	0.32	**99.14**	583.78	**99.30**	593.60
$r-23$	1050	7200	4	16	900	900	29 881	**99.88**	0.48	**99.98**	934.69	**99.98**	1468.00
$r-24$	1050	7200	4	16	900	900	29 677	**99.87**	0.53	**99.98**	1108.60	**99.98**	1466.73
Average								68.07	0.25	72.41	221.34	72.36	274.73

TABLE 4 Evaluation of the impact of the number of iterations in ALNS and E-ALNS.

		Number of iterations											
		100		200		400		600		800		1000	
ID	Method	% scan	Time	% scan	Time	% scan	Time	% scan	Time	% scan	Time	% scan	Time
$r-9$	ALNS	69.92	67.74	70.38	134.43	70.76	262.78	70.87	394.92	70.89	522.51	70.97	659.35
$r-9$	E-ALNS	69.87	72.33	70.05	149.24	70.25	295.05	70.75	449.59	70.75	593.92	70.79	740.13
$r-10$	ALNS	68.31	67.49	69.08	134.88	69.33	265.89	69.37	399.46	69.42	532.29	69.42	666.05
$r-10$	E-ALNS	68.17	66.87	68.95	137.92	69.40	279.69	69.57	421.03	69.71	559.07	69.92	699.22
$r-15$	ALNS	89.48	185.26	90.10	383.00	90.10	749.82	90.10	1100.02	90.18	1462.69	90.19	1832.57
$r-15$	E-ALNS	90.01	257.05	90.46	479.56	90.46	902.18	90.47	1438.69	90.47	2056.42	90.47	2440.27
$r-16$	ALNS	88.81	165.11	89.02	360.97	89.27	687.99	89.49	999.99	89.49	1181.64	89.55	1629.64
$r-16$	E-ALNS	89.19	210.35	89.27	446.35	89.47	821.16	89.53	1262.08	89.64	1642.31	89.64	2103.46

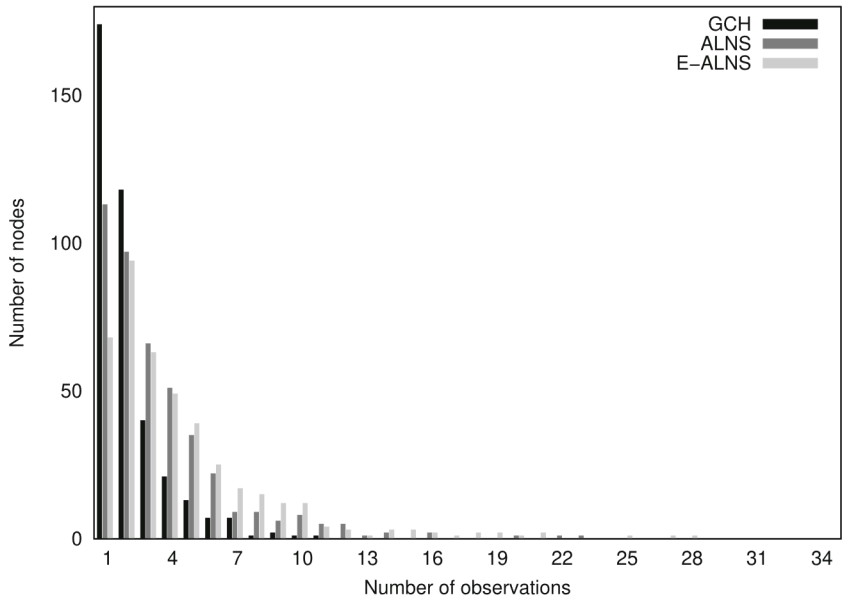

FIGURE 6 The distribution of the number of observations per node.

average, GCH solves all realistic instances in less than 1 s, while ALNS and E-ALNS take up to 25 min to compute their solution. This means that, after the data of a disaster is obtained, drones can start looking for survivors and victims within half an hour. Furthermore, it can be noticed that setting new bases on the map without increasing the number of drones does not increase the quality of the solution. Similar outcomes were also observed in other works which dealt with post-disaster management problems [16, 40]. It is also worth noting that a larger drone fleet leads to better outcomes. Finally, one can see that fleets of four drones are able to get an expected scan of at least 80% of the targets in a two-hours operation (7200 time units), while fleets of 16 drones are able to scan almost all targets in the same operation time. This indicates that the proposed approaches are a promising tool for post-disaster management.

Table 4 shows the impact of the number of iterations for ALNS and E-ALNS on the results for four instances, $r-9$, $r-10$, $r-15$, and $r-16$. One can see the improvement is quite low, while the CPU time grows linearly with the number of iterations: the gain is around 1% (except for $r-10$ with E-ALNS for which it is around 1.75%) for a tenfold increase in CPU time. When the number of iterations is low, ALNS is on par with E-ALNS. However, for larger values E-ALNS provide better results, with a larger CPU time due to the additional moves.

Figure 6 displays the distribution of the number of observations per node for the solutions found by GCH, ALNS and E-ALNS. The instance considered is $r-12$ with 1050 nodes, time horizon 7200, 2 bases and 4 drones, and 29 869 targets to identify. For the three solutions, around 600 nodes (57% of the nodes) are not visited. They correspond to nodes with no targets, for instance the nodes corresponding to areas with no buildings. For the remaining nodes, one can note a different distribution between the solution computed by GCH and the two other solutions. In the former, most visited nodes receive 1, or 3 observations and none takes more than 10 observations. For the two other solutions, the number of observations is significantly larger, and some nodes have up to 20 observations. This can be explained by the fact that GCH adds new scans on the current node until

TABLE 5 Evaluation of the drone range variation.

Instance								GCH		ALNS		E-ALNS	
ID	$\|N\|$	H	$\|B\|$	$\|D\|$	R	T_R	Q	% scan	Time	% scan	Time	% scan	Time
$r-12$	1050	7200	2	4	900	900	29 869	80.93	0.23	86.83	84,25	87.35	118.83
$r-12$	1050	7200	2	4	1800	900	29 869	94.99	0.38	97.48	127.10	97.37	124.60
$r-12$	1050	7200	2	4	3600	900	29 869	97.06	0.41	99.41	370.21	99.62	6304.23
$r-12$	1050	7200	2	4	7200	900	29 869	98.12	0.42	99.98	165.28	99.98	5820.63
Average								92.78	0.37	96.06	186.71	96.08	3092.07

TABLE 6 Impact of the drone range on the time a drone spends travelling, scanning, and recharging.

Instance								Scan		Travel		Recharge	
ID	$\|N\|$	H	$\|B\|$	$\|D\|$	R	T_R	Q	Time	% time	Time	% time	Time	% time
$r-12$	1050	7200	2	4	900	900	29 869	317.25	4.75	2564.77	38.32	3827.49	56.93
$r-12$	1050	7200	2	4	1800	900	29 869	661.50	9.80	4285.14	63.49	1802.28	26.71
$r-12$	1050	7200	2	4	3600	900	29 869	761.75	11.36	5056.04	75.23	900.94	13.41
$r-12$	1050	7200	2	4	7200	900	29 869	895.00	12.67	6167.76	87.32	0.64	0.01
Average								658.88	9.64	4518.43	66.09	1632.84	24.26

the drone moves to another one. Then, it becomes less interesting to come back to the node, with few targets left. And, as a consequence, the metaheuristics can perform more scans in the solutions by improving the drone routing.

Table 5 reports the impact of increasing the drone range on the solutions. The goal of this experiment is to assess the interest of using other technologies (for instance hydrogen-powered drones) in order to benefit from a larger range. Here, the range of the drones, converted in time units, varies 900 (15 min) to 7200 (2 h) in Table 5. The first eight columns of this table are similar to those in Table 3. The average percentages of identified targets (*%scan*) and average running times (*time*) for GCH, ALNS and E-ALNS are presented in columns 8 to 13. One can observe that increasing the drone range leads to a higher percentage of expected scanned targets in all algorithms, as the drones spend comparatively less time recharging at the base. However, E-ALNS takes, on average, almost 2 h to solve the realistic instances with ranges of 1 and 3 h. It means that the additional step introduced in this method struggles to handle solutions with larger trip sizes. Hence, it is recommended using ALNS if the range of the drone fleet exceeds 1 h (3600 time units), as this algorithm computes solutions as good as E-ALNS in approximately 3 min on average.

Table 6 illustrates the impact of the drone range on the traveling, scanning, and recharging times in instance $r-12$, using the GCH heuristic. This analysis aims at evaluating the distribution of time allocation for drones across distinct tasks as their operational range increases. Each row corresponds to a specific drone range R, measured in time units. The first eight columns of this table are similar to those in Table 3. Columns 8 to 13 provide information on the average time and on the time ratio dedicated to scanning, traveling between nodes, and recharging. In instances with a range $R \geq 1800$ time units, the drones predominantly allocate their time to travels between nodes. On the other hand, in the scenario with a drone range $R = 900$, the drones spend most of the time recharging. Additionally, it is observed that, on average, the time ratio a drone actively spends scanning a node is 4.75%. However, with a range set to $R = 7200$ time units, the average scan ratio per drone increases to 12.67%. Notably, the increase in drone range has a more significant impact on travel time than on scanning time.

The results of the experiment presented in Table 7 depict the impact of the correlation between two consecutive scans at a node. This correlation is given by a variable $\alpha \in [0, 1]$, and represents the similarities/discrepancies between two consecutive scans of a same node. If $\alpha = 0$, the images are similar, meaning that the same targets will be identified. In this case, the problem is a variation of the team orienteering problem, as each node only needs to be visited at most once and profits associated to visited nodes have to be maximized. If $\alpha = 1$, the two pictures are significantly different, which means the target identification is different. Any intermediate value implies that some of the targets found in a new image were previously scanned, while others were not. The first eight columns of Table 7 are similar to Table 3 and α is the correlation factor. The average percentage of expected identified targets (*%scan*) and the average running time (*time*) in seconds are shown for each method in columns 9 to 14. Bold stands for the highest percentage of expected value achieved by each approach. One can notice that the worst solutions are produced when $\alpha = 0$, since each visited node does not need to be scanned more than once. It is also worth mentioning that intermediate values of α produce better solutions than $\alpha = 1$. The best outcomes for GCH, ALNS and E-ALNS are found when $\alpha = 0.8$, $\alpha = 0.8$, and $\alpha = 0.5$, respectively. This means that avoiding excessive searches on certain nodes allows the drones to cover a larger region, leading to a higher number of targets scanned during a two-hour operation. However, in an unlimited

TABLE 7 Impact of the variation in the correlation between two successive scans of a same node.

Instance								GCH		ALNS		E-ALNS	
ID	$\|N\|$	H	$\|B\|$	$\|D\|$	R	α	Q	% scan	Time	% scan	Time	% scan	Time
$r-12$	1050	7200	2	4	900	0	29 869	58.76	0.22	63.32	323.79	63.52	451.54
$r-12$	1050	7200	2	4	900	0.2	29 869	74.94	0.24	87.55	121.35	87.38	112.72
$r-12$	1050	7200	2	4	900	0.5	29 869	79.69	0.26	87.52	111.29	**87.90**	107.59
$r-12$	1050	7200	2	4	900	0.8	29 869	**82.36**	0.28	**88.06**	128.65	87.64	116.90
$r-12$	1050	7200	2	4	900	1	29 869	80.93	0.23	86.83	84.25	87.35	118.83
Average								75.34	0.28	82.76	153.86	82.76	181.52

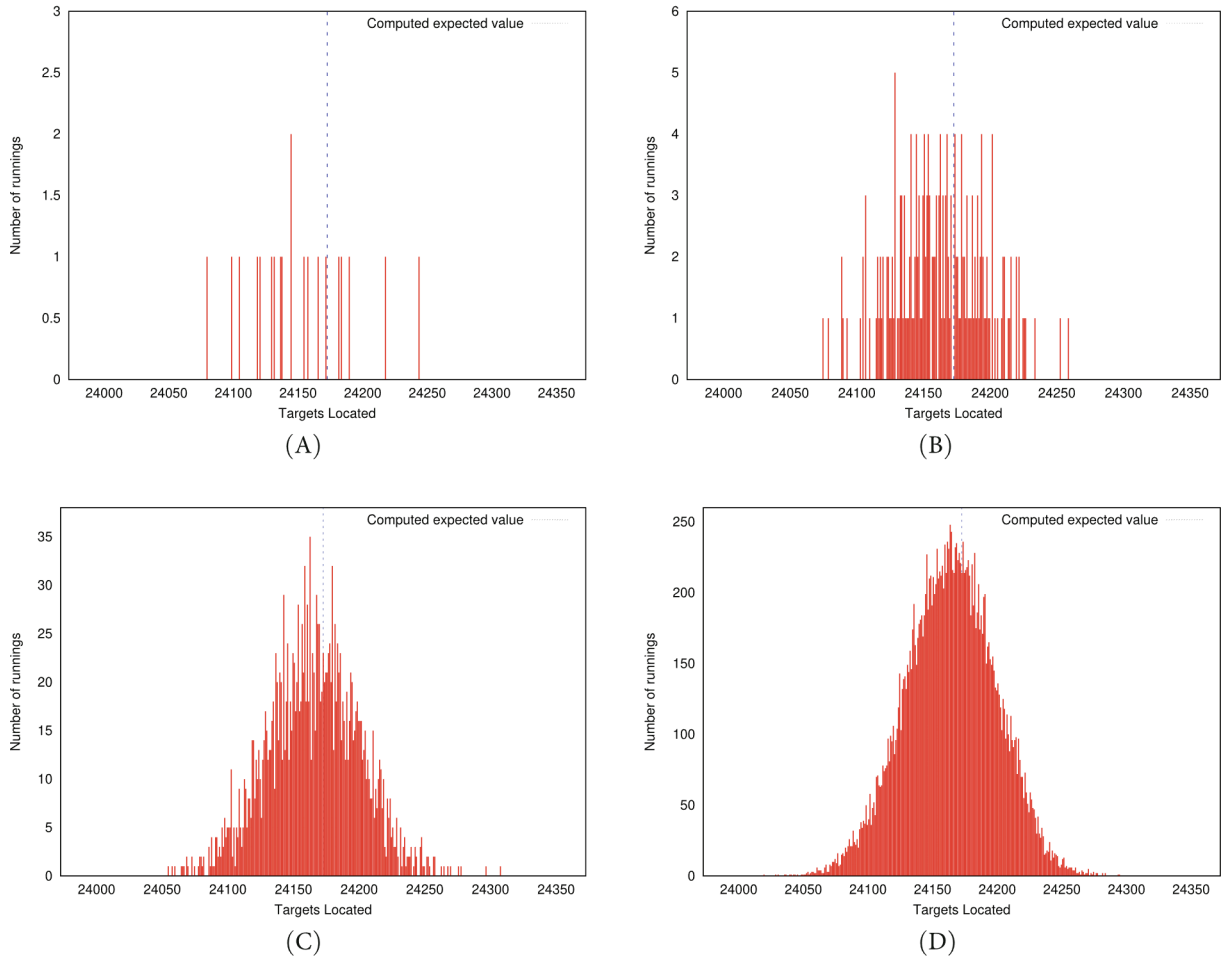

FIGURE 7 The distribution of the number of targets found after (A) 20, (B) 200, (C) 2000, and (D) 20 000 simulations.

time horizon, $\alpha = 1$ should lead to the best solutions. Alternative diversification mechanisms should be investigated to address the counter-intuitive fact that $\alpha = 1$ does not produce the best results in a 2 h time horizon.

5.3 | Convergence towards the expected value

The graphs in Figure 7 show the relevance of the expected values from the solutions when evaluated numerically by running several simulations. To evaluate this impact, we select the representative instance $r-12$ with four homogeneous drones, a 900 time units range, where these drones are equally divided into two different bases. The time horizon is set to 2 h (7200 s), and the total number of targets in this instance is 29 869. The expected value obtained by GCH for $r-12$ is 80.93%, which represents approximately 24 174 scanned victims.

Given the solution computed by GCH, ALNS or E-ALNS, one simulation run consists in following the route of each drone. For each observation on the nodes of the route, a random number is generated for each unidentified target to check if it becomes identified. The total number of targets identified is kept. The idea is to check the convergence of the simulations towards the

computed expected value. Figure 7A–D illustrate the distributions after 20, 200, 2000, and 20 000 runs, respectively. In the four figures, the x-axis presents the number of targets, while the red columns on the y-axis represent the number of times each value was returned by the simulations. Furthermore, the blue dotted line displays the expected value of the solution computed for the instance. As shown in the histograms in Figure 7A–D, an increase in the number of simulations leads to a better approximation to a normal distribution centered on the expected value. This behavior is clearly observable in Figure 7C,D, which show the results for 2000 and 20 000 runs, respectively. Finally, it is worth mentioning that the mean values $\overline{m_k}$, where $k = \{20, 200, 2000, 20\,000\}$ runs, are $\overline{m_{20}} = 24152.00$, $\overline{m_{200}} = 24162.79$, $\overline{m_{2000}} = 24166.44$, and $\overline{m_{20\,000}} = 24165.40$, while the standard deviations σ_k are $\sigma_{20} = 40.04$, $\sigma_{200} = 33.76$, $\sigma_{2000} = 34.97$, and $\sigma_{20\,000} = 34.93$.

These results show using expected values in our model is meaningful. This provides routing solutions for drones that capture the variability of the scanning operations, as implemented in the simulations.

6 | CONCLUDING REMARKS AND PERSPECTIVES

In this study, the PDRP is proposed, which aims at searching and identifying victims by means of drones, for difficult operations in inaccessible, contaminated or dangerous areas after major disasters. Several additional issues make PDRP differ from other problems found in the literature. For instance, this problem considers anti-collision constraints, a probabilistic identification model, and it allows multiples visits, while maximizing the expected number of identified victims.

Heuristics methods have been developed: a constructive heuristic, refereed to as GCH, an ALNS and an E-ALNS. GCH provides a quick initial solution that can be used in emergencies (immediately after an event). The goal of ALNS and E-ALNS is to improve the outcome given by GCH, with the aim of identifying a higher number of survivors or victims, which would enable a more accurate and efficient rescue operation. Computation results indicate that all methods produce high-quality solutions in reasonable running times. Yet, E-ALNS performed better than the other methods for theoretical instances, while ALNS returns the best solutions on the set of realistic instances. This could be explained by the fact that abstract instances do not fully capture the features from real situations. The results on larger instances coming from the port of Beirut explosion show that these methods are able to handle realistic size data. In summary, ALNS seems to be a good compromise between the quality of the solution and the computation time.

This study opened several avenues of research. We are investigating a dynamic heatmap, which real time updated information from the situation on the ground, in order to better control the search of victims through the drone flight. For such a purpose, on-line algorithms need to be developed with a specialized group able to crew and program drones flights. Another forthcoming study involves a decentralized approach, where drone swarms work cooperatively to maximize the number of identified victims in an area affected by an industrial disaster. Some open questions were also raised, such as: are there some better suited objective functions for the focused goal? What are the special characteristics that make the instances more difficult to be addressed? Last but not least, exact methods could also be developed, even if it would be difficult to deploy them in practices due to the execution time, they still remain a way to assess the solutions quality produced by the heuristics.

ACKNOWLEDGMENTS

This research is supported by the RIN Émergent LIS project, funded by Région Normandie, program FEDER-FSE/IEJ, Haute-Normandie, France.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Amadeu Almeida Coco https://orcid.org/0000-0001-7257-7368
Christophe Duhamel https://orcid.org/0000-0001-6522-7343
Andréa Cynthia Santos https://orcid.org/0000-0003-2497-4072
Matheus Nohra Haddad https://orcid.org/0000-0002-7809-0611

REFERENCES

[1] G. S. C. Avellar, G. A. S. Pereira, L. C. A. Pimenta, and P. Iscold, *Multi-UAV routing for area coverage and remote sensing with minimum time*, Sensors **15** (2015), no. 11, 27783–27803.

[2] N. Boysen, S. Fedtke, and S. Schwerdfeger, *Last-mile delivery concepts: A survey from an operational research perspective*, OR Spectr. **43** (2021), no. 1, 1–58.

[3] R. Z. B. Bravo, A. Leiras, and F. L. C. Oliveira, *The use of UAVs in humanitarian relief: An application of POMDP-based methodology for finding victims*, Prod. Oper. Manag. **28** (2019), no. 2, 421–440.

[4] M. Cannioto, A. D'Alessandro, G. L. Bosco, S. Scudero, and G. Vitale, *Brief communication: Vehicle routing problem and UAV application in the post-earthquake scenario*, Nat. Hazards Earth Syst. Sci. **17** (2017), no. 11, 1939–1946.

[5] A. R. Cassandra, *Exact and approximate algorithms for partially observable Markov decision processes*, Brown University, Providence, RI, 1998.

[6] I. M. Chao, B. L. Golden, and E. A. Wasil, *The team orienteering problem*, Eur. J. Oper. Res. **88** (1996), no. 3, 464–474.

[7] M. A. Cheaito and S. Al-Hajj, *A brief report on the Beirut port explosion*, Mediterr. J. Emerg. Med. Acute Care **1** (2020), no. 4.

[8] S. Chowdhury, M. Marufuzzaman, H. Tunc, L. Bian, and W. Bullington, *A modified ant colony optimization algorithm to solve a dynamic traveling salesman problem: A case study with drones for wildlife surveillance*, J. Comput. Des. Eng. **6** (2019), no. 3, 368–386.

[9] S. Chowdhury, O. Shahvari, M. Marufuzzaman, X. Li, and L. Bian, *Drone routing and optimization for post-disaster inspection*, Comput. Ind. Eng. **159** (2021), 107495.

[10] W. P. Coutinho, J. Fliege, and M. Battarra, *Glider routing and trajectory optimisation in disaster assessment*, Eur. J. Oper. Res. **274** (2019), no. 3, 1138–1154.

[11] D. C. Dang, R. N. Guibadj, and A. Moukrim, *An effective PSO-inspired algorithm for the team orienteering problem*, Eur. J. Oper. Res. **229** (2013), no. 2, 332–344.

[12] G. De Castro Pena, A. C. Santos, and C. Prins, *Solving the integrated multi-period scheduling routing problem for cleaning debris in the aftermath of disasters*, Eur. J. Oper. Res. **306** (2022), no. 1, 156–172.

[13] E. Demir, T. Bektaş, and G. Laporte, *An adaptive large neighborhood search heuristic for the pollution-routing problem*, Eur. J. Oper. Res. **223** (2012), no. 2, 346–359.

[14] A. Di Placido, C. Archetti, and C. Cerrone, *A genetic algorithm for the close-enough traveling salesman problem with application to solar panels diagnostic reconnaissance*, Comput. Oper. Res. **145** (2022), 105831.

[15] H. Duan, X. Zhang, J. Wu, and G. Ma, *Max-min adaptive ant colony optimization approach to multi-UAVs coordinated trajectory replanning in dynamic and uncertain environments*, J. Bionic Eng. **6** (2009), no. 2, 161–173.

[16] C. Duhamel, A. C. Santos, D. Brasil, E. Châtelet, and B. Birregah, *Connecting a population dynamic model with a multi-period location-allocation problem for post-disaster relief operations*, Ann. Oper. Res. **247** (2016), no. 2, 693–713.

[17] C. Fikar, M. Gronalt, and P. Hirsch, *A decision support system for coordinated disaster relief distribution*, Expert Syst. Appl. **57** (2016), 104–116.

[18] E. Galceran and M. Carreras, *A survey on coverage path planning for robotics*, Robot. Auton. Syst. **61** (2013), no. 12, 1258–1276.

[19] K. Glock and A. Meyer, *Mission planning for emergency rapid mapping with drones*, Transp. Sci. **54** (2020), no. 2, 534–560.

[20] F. Guerriero, R. Surace, V. Loscri, and E. Natalizio, *A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints*, Appl. Math. Model. **38** (2014), no. 3, 839–852.

[21] L. A. Haidari, S. T. Brown, M. Ferguson, E. Bancroft, M. Spiker, A. Wilcox, R. Ambikapathi, V. Sampath, D. L. Connor, and B. Y. Lee, *The economic and operational value of using drones to transport vaccines*, Vaccine **34** (2016), no. 34, 4062–4067.

[22] V. C. Hemmelmayr, J. F. Cordeau, and T. G. Crainic, *An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics*, Comput. Oper. Res. **39** (2012), no. 12, 3215–3228.

[23] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl, *The electric fleet size and mix vehicle routing problem with time windows and recharging stations*, Eur. J. Oper. Res. **252** (2016), no. 3, 995–1018.

[24] P. Kitjacharoenchai, B. Min, and S. Lee, *Two echelon vehicle routing problem with drones in last mile delivery*, Int. J. Prod. Econ. **225** (2020), 107598.

[25] A. A. Kovacs, S. N. Parragh, K. F. Doerner, and R. F. Hartl, *Adaptive large neighborhood search for service technician routing and scheduling problems*, J. Sched. **15** (2012), no. 5, 579–600.

[26] H. Li, J. Chen, F. Wang, and Y. Zhao, *Truck and drone routing problem with synchronization on arcs*, Nav. Res. Logist. **69** (2022), no. 6, 884–901.

[27] Y. Liu, *An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones*, Comput. Oper. Res. **111** (2019), 1–20.

[28] J. E. Macias, N. Goldbeck, P. Y. Hsu, P. Angeloudis, and W. Ochieng, *Endogenous stochastic optimisation for relief distribution assisted with unmanned aerial vehicles*, OR Spectr. **42** (2020), no. 4, 1089–1125.

[29] G. Macrina, L. D. P. Pugliese, F. Guerriero, and G. Laporte, *Drone-aided routing: A literature review*, Transp. Res. Part C Emerg. Technol. **120** (2020), 102762.

[30] S. T. W. Mara, A. P. Rifai, and B. M. Sopha, *An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops*, Expert Syst. Appl. **205** (2022), 117647.

[31] V. Mersheeva and G. Friedrich, "*Routing for continuous monitoring by multiple micro AVS in disaster scenarios*," *ECAI 2012*, IOS Press, Amsterdam, The Netherlands, 2012, pp. 588–593.

[32] M. Moshref-Javadi and M. Winkenbach, *Applications and research avenues for drone-based models in logistics: A classification and review*, Expert Syst. Appl. **177** (2021), 114854.

[33] A. Nedjati, G. Izbirak, B. Vizvari, and J. Arkat, *Complete coverage path planning for a multi-UAV response system in post-earthquake assessment*, Robotics **5** (2016), no. 4, 26.

[34] B. E. Oruc and B. Y. Kara, *Post-disaster assessment routing problem*, Transp. Res. B Methodol. **116** (2018), 76–102.

[35] S. H. Othman and G. Beydoun, *Model-driven disaster management*, Inf. Manag. **50** (2013), no. 5, 218–228.

[36] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, *Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey*, Networks **72** (2018), no. 4, 411–458.

[37] L. D. P. Pugliese, F. Guerriero, E. Natalizio, and N. R. Zema, "*A biobjective formulation for filming sport events problem using drones*," *2017 9th IEEE Int. Conf. Intell. Data Acquis. Adv. Comput. Syst.: Technol. Appl. (IDAACS)*, Vol **2**, IEEE, New York, 2017, pp. 639–644.

[38] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner, *Networked UAVs as aerial sensor network for disaster management applications*, Elektrotech. Inf. **127** (2010), no. 3, 56–63.

[39] B. Rabta, C. Wankmüller, and G. Reiner, *A drone fleet model for last-mile distribution in disaster relief operations*, Int. J. Disaster Risk Reduct. **28** (2018), 107–112.

[40] M. Rekik, A. Ruiz, J. Renaud, D. Berkoune, and S. Paquet, "*A decision support system for humanitarian network design and distribution operations*," *Humanitarian and relief logistics*, V. Zeimpekis, S. Ichoua, and I. Minis (eds.), Springer, New York, 2013, pp. 1–20.

[41] S. Ropke and D. Pisinger, *An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows*, Transp. Sci. **40** (2006), no. 4, 455–472.
[42] M. Rosalie, G. Danoy, S. Chaumette, and P. Bouvry, *Chaos-enhanced mobility models for multilevel swarms of UAVs*, Swarm Evol. Comput. **41** (2018), 36–48.
[43] C. Rottondi, F. Malandrino, A. Bianco, C. F. Chiasserini, and I. Stavrakakis, *Scheduling of emergency tasks for multiservice UAVs in post-disaster scenarios*, Comput. Netw. **184** (2021), 107644.
[44] D. Sacramento, D. Pisinger, and S. Ropke, *An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones*, Transp. Res. Part C Emerg. Technol. **102** (2019), 289–315.
[45] A. V. Savkin and H. Huang, *A method for optimized deployment of a network of surveillance aerial drones*, IEEE Syst. J. **13** (2019), no. 4, 4474–4477.
[46] D. Schermer, M. Moeini, and O. Wendt, *A matheuristic for the vehicle routing problem with drones and its variants*, Transp. Res. Part C Emerg. Technol. **106** (2019), 166–204.
[47] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, *Sensor planning for a symbiotic UAV and UGV system for precision agriculture*, IEEE Trans. Robot. **32** (2016), no. 6, 1498–1511.
[48] P. Toth and D. Vigo, Eds., *Vehicle routing: Problems, methods, and applications*, MOS-SIAM Series on Optimization, Vol **18**, SIAM, Philadelphia, PA, 2014.
[49] D. R. Viloria, E. L. Solano-Charris, A. Muñoz-Villamizar, and J. R. Montoya-Torres, *Unmanned aerial vehicles/drones in vehicle routing problems: A literature review*, Int. Trans. Oper. Res. **28** (2021), no. 4, 1626–1657.
[50] S. Waharte and N. Trigoni, "*Supporting search and rescue operations with UAVs*," *2010 Int. Conf. Emerg. Secur. Technol.*, IEEE, New York, 2010, pp. 142–147.
[51] T. Wen, Z. Zhang, and K. Wong, *Multi-objective algorithm for blood supply via unmanned aerial vehicles to the wounded in an emergency situation*, PLoS One **11** (2016), no. 5, e0155176.
[52] S. Yahyanejad, D. Wischounig-Strucl, M. Quaritsch, and B. Rinner, "*Incremental mosaicking of images from autonomous, small-scale UAVs*," *2010 7th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, IEEE, New York, 2010, pp. 329–336.
[53] R. Yan, X. Zhu, X. Zhu, and R. Peng, *Optimal routes and aborting strategies of trucks and drones under random attacks*, Reliab. Eng. Syst. Saf. **222** (2022), 108457.
[54] D. Zorbas, L. P. Pugliese, T. Razafindralambo, and F. Guerriero, *Optimal drone placement and cost-efficient target coverage*, J. Netw. Comput. Appl. **75** (2016), 16–31.

How to cite this article: A. Almeida Coco, C. Duhamel, A. C. Santos, and M. N. Haddad, *Solving the probabilistic drone routing problem: Searching for victims in the aftermath of disasters*, Networks. **84** (2024), 31–50. https://doi.org/10.1002/net.22214