Qiao, T., Sambo, Y. A., Imran, M. A. and Ahmed, W. (2020) Drone Trajectory Optimization using Genetic Algorithm with Prioritized Base Stations. In: IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, Online Only, 14-16 Sep 2020, ISBN 9781728163390 (doi:10.1109/CAMAD50429.2020.9209291).

Drone Trajectory Optimization using Genetic Algorithm with Prioritized Base Stations

Tianrui Qiao[1], Yusuf A. Sambo[2], Muhammad A. Imran[2] and Wasim Ahmad[2]
[1] Glasgow College UESTC, University of Electronic Science and Technology of China, Chengdu, China.
[2] James Watt School of Engineering, University of Glasgow, Scotland, United Kingdom
Email: wasim.ahmad@glasgow.ac.uk

***Abstract*—Drones have been widely applied to perform emergent tasks in the post-disaster scenario, due to their unique characteristics such as mobility, flexibility, and adaptivity to altitude. However, drones have limited energy capacity, which presents a major drawback in flight time and affects their performance in such scenarios. Hence, trajectory optimization has become a critical research problem for such applications of drones. In this paper, we present an optimal trajectory design for a single drone to ferry data from temporary Base Stations (BSs) deployed within a disaster zone to a fixed gateway node that is the point of origin and final destination for the drone flight. We have used a Genetic Algorithm (GA)-based approach that takes into account the shortest distance traveled and least time spent by the drone during flight. We also examine the case where some BSs have delay requirements that are unknown to the drone in advance. Simulation results show that the performance of our proposed GA-based approach matches that of the benchmark exhaustive search algorithm and the difference in computational time between the 2 algorithms increases with the number of BSs. Accordingly, our proposed algorithm has 96.4% lower computational time complexity compared to the benchmark exhaustive search algorithm when there are 12 BSs in the disaster area.**

***Index Terms*—drone, genetic algorithm, trajectory optimization, prioritized base Stations, time-sensitive data.**

I. Introduction

In post-disaster scenarios, critical communications infrastructure could be damaged leaving large parts of the network cut-off, thereby affecting the operations of first responders. Network recovery is often very slow in this case, sometimes taking days or even weeks to get the network back to normal. Hence, there is a need for innovative deployment techniques to provide temporary coverage before repair works are completed. Drones are increasingly playing an important role in post-disaster management due to their unique characteristics, such as unmanned system, mobility, flexibility, and adaptivity to altitude [1]. Drones can act as communication relays between users and workable infrastructures, and also can function as devices of collecting and delivering data [2]. However, energy limitation of drones result in a major drawback in flight time, which significantly influences performance [3]. Drone path planning, therefore, becomes a critical performance evaluation criterion in drone-assisted networks.

Several algorithms, such as Artificial Neural Network (ANN) [4], ant colony [5], simulated annealing algorithm [6], game theory [7] and Genetic Algorithm (GA), have been designed to perform path planning tasks. GA has been applied widely due to its good performance in locating near optimal solutions in large space within a short computation time [8]. GA is a random global search and optimization method designed by imitating the evolutionary mechanism of natural organisms, whose essence is an efficient, parallel, and global search method [9].

The problem of trajectory optimization has been considered under many different conditions. In [10], the authors have implemented GA, adding an extra step of local optimization, to find the shortest closed flyable path for Unmanned Aerial Vehicle (UAV) while escaping the coverage zones of the radars. In [1], an adaptive evolutionary planner based on GA was adopted to generate a path for UAV in an environment with unknown obstacles. Scenarios considered in [11] are similar to one presented in [1] but it has a distinctive mechanism for avoiding-obstacles, it adds extra corner points of the intersected forbidden zones in the pre-optimized path. A commonality among these studies is that they consider the path planning problem in the environments with obstacles, where the objective of the optimization is the minimization of the length of the trajectory. Other restrictive factors such as energy limitations are not taken into consideration.

There are also some researches trying to achieve the purpose of joint optimization for the routes of drones. In [12], we have proposed a novel trajectory design based on GA to find the optimal path with the least energy consumption requirement. In [13], the researchers have introduced an iterative algorithm that applies the block coordinate descent and successive convex optimization techniques to realize a joint optimization of throughput, multi-user communication scheduling, drones' trajectory, and power control. However, in these studies, all BSs are assumed as peers, which means their messages to be delivered by the drone have the same delay requirements. This problem is what we address in this paper.

In this paper, we address the problem of path planning under a scenario of drone-assisted post-disaster communication where the network infrastructure in a given area has been destroyed but there is a gateway node with access to the core network. A drone then needs to visit and collect data from all the BSs in the disaster area and ferry them back to a gateway node. In our study, we consider two cases, in the first case, the only factor taken into consideration is the path length. For the second case, we introduce BS prioritization due to delay requirements that are unknown by the drone before taking off. Data from those prioritized BSs must be ferried back to the core network within certain time limitations, otherwise, the data would be lost after the time expires. The best route obtained by GA is the shortest one that would minimize the travel time of the drone.

The remainder of the paper is organized as follows. Section II provides an overview of the system deployed. Section III presents the implementation steps of GA along with the description of the nature of operators applied and their corresponding principles. A detailed description of a different set of scenarios considered and simulation results produced for those scenarios are presented in Section IV. Finally, conclusions are drawn in Section V.

II. System Overview

Consider a post-disaster scenario, where all communication links between the BSs in the disaster area and the core network are broken. One BS is assumed to have a workable access to the core network, acting as a gateway node. All BSs, except the gateway BS, can deliver messages only through some external assisting devices. In such scenario, single drone can be used to visit all BSs in the disaster area to collect data and then transport it back to the gateway node periodically with an objective to build a temporary communication network and maintain necessary communication. Figure 1 provides an example of BSs topology and ideal flight trajectory. The solid segments represent routes that the drone has covered, and the dashed ones indicate paths waiting to be travelled.

A. Topology of BSs

The BSs and the gateway node are generated within a square region with each side of length 1000 meters. A Cartesian coordinate system is drawn on this areas which divide this area into four equal sub-areas each represented by one of the quadrant of the Cartesian plane as shown in Figure 1. For a clearer view and less computational complexity, the gateway node is assumed to be located at the center of the region, where coordinates of the Cartesian plane are (0,0). Locations of other BSs are generated randomly.

B. Flight Model

All situations are considered in a two-dimensional map, and thus the altitude of drone is not taken into consideration. It is assumed that the drone moves in a straight line with a constant speed of v. The amount of time needed for the drone to take off and land are neglected. The time spent on travelling directly from node (x_i,y_i) to node (x_j,y_j) is calculated by (1).

$$t_{i,j} = \frac{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}{v} (s) \tag{1}$$

The factor of energy limitation is not taken into consideration, it means drone has enough energy to visit all BSs without a need of flying back to the gateway node for replenish.

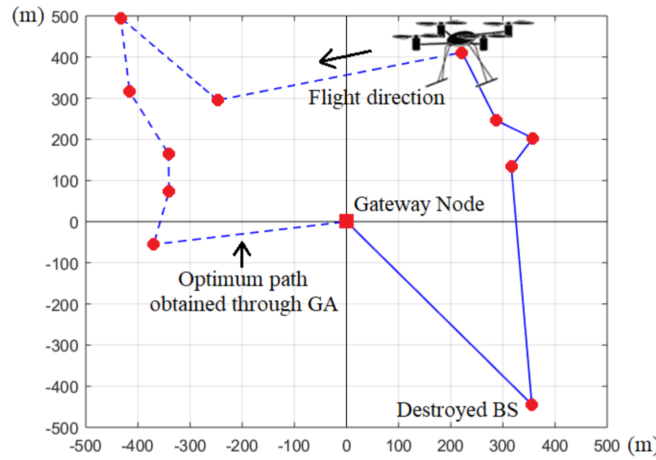

Fig. 1. System implementation.

III. Implementation of Genetic Algorithm

The searching process of GA starts from a population, which consists of several solutions presented by individuals. The population evolves towards a better performance through the procedure of selection, crossover and mutation. The optimal solution is selected according to the fitness after a certain number of generations. Figure 2 indicates the working logic of GA.

A. Representation of Chromosomes

A chromosome is essentially a collection of genes whose genotype, a kind of gene combination, determines the external expression (phenotype) of an individual. Each BS and the gateway node are given a number, which is regarded as a gene. The varying arrangement of genes makes up a chromosome or a trajectory.

Let $C_{k,l}$ represent the l^{th} gene on the k^{th} chromosome in the population, $\forall l$ = 1, 2, ···, N and $\forall k$ = 1, 2,···, M, where N denotes the number of genes in each chromosome, i.e., the total number of nodes, and M represents the number of chromosomes. Therefore, the flight of the drone can be described as travelling from the gateway node to all the BSs and then returning to the gateway node, i.e., $C_{k,N}$ through $C_{k,1}$, $C_{k,2}$, ···, $C_{k,N-1}$ and back to $C_{k,N}$, for any trajectory C_k. Before the start of evolution, M individuals are generated randomly to form an initial population, $\{C_1, C_2, \cdots, C_M\}$ based on which the following genetic processes operates.

B. Fitness Value

The fitness value measures the degree of adaptation of an individual to its living environment. The larger it is, the more it adapts to the environment, and thus less likely, it gets eliminated [10]. In this paper, fitness evaluates the performance of a route in terms of its total length, which is defined as

$$D_k = \sum_{i=1}^{N-1} d_{k,i}, \tag{2}$$

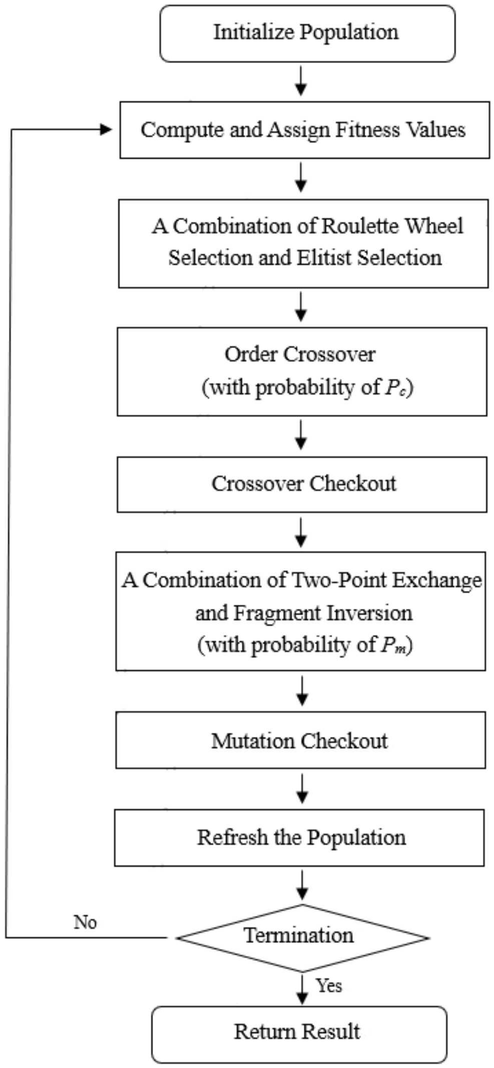

Fig. 2. Flow chart of GA

such that

$$d_{k,i} = \sqrt{(x_{k,i+1} - x_{k,i})^2 + (y_{k,i+1} - y_{k,i})^2}, i \in [1, N-1] \tag{3}$$

where $d_{k,i}$ represents the distance between BS i and BS i+1, whose coordinates are $(x_{k,i}$,$y_{k,i})$ and $(x_{k,i+1}$,$y_{k,i+1})$ respectively, on the k^{th} trajectory. D_k denotes the total length of route C_k.

The fitness value of the k^{th} route, denoted as $f(C_k)$, is expressed as (4).

$$f(C_k) = \frac{1}{D_k} \tag{4}$$

C. Selection

The basic idea of Roulette wheel selection is that the probability of each individual being selected is directly proportional to its fitness. A Roulette wheel will be established according to their cumulative probabilities. The larger the fitness value is, the more likely it is, the individual with this value will be

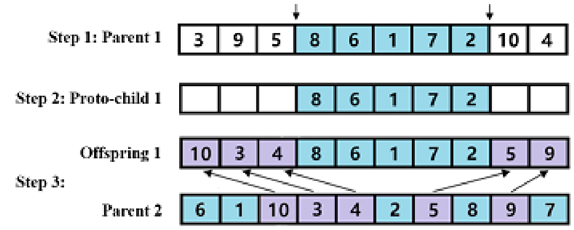

Fig. 3. Graphic explanation of OX

selected for the next generation. This process will be repeated for M times, until the population size reaches the original value [14].

Whereas in the elitist selection the best individual in the population to date (elitist) is directly copied to the next generation and replace the worst one without crossover and mutation. Therefore, the traits of the best individual will be maintained [15]. The elitist selection is performed before roulette wheel selection.

D. Crossover

In order to avoid gene repetition and achieve higher efficiency, Order Crossover (OX) is chosen as the crossover operator. A gene fragment on one parent is randomly chosen and directly transferred to its offspring. Same genes on the other parent are found out and the left ones fill up the offspring's vacancies in order. Figure 3 illustrates how Offspring 1 is generated and the other one is available in the similar way. Crossover procedure is applied with a certain probability P_c. The parents will be replaced by the obtained offsprings only if they have better fitness values.

E. Mutation

Two-point exchange and fragment inversion are chosen as the mutation operators, where two genes are arbitrarily selected to swap their positions at first, and then the sequence of the randomly chosen gene fragment is inverted. Figure 4 details the mutation procedure. Mutation will be applied if the randomly generated number is smaller than a probability P_m. The mutated individual will replace the original one in the parents pool of next generation only if its fitness value is larger.

IV. Simulation Setup and Results

In this section, we present our simulation setup and compare the performance of our algorithm against exhaustive search method, which is described in the next subsection. All results are obtained using Monte Carlo simulation.

A. Exhaustive Search

In this project, the basic idea of exhaustive search is to find out the trajectory with the shortest length, searching space is the set of all possible trajectories. Its advantage is obvious, which is direct operating logic, but at the expense of complexity as the number of BSs increases. Exhaustive search method is used here as benchmark for our algorithm.

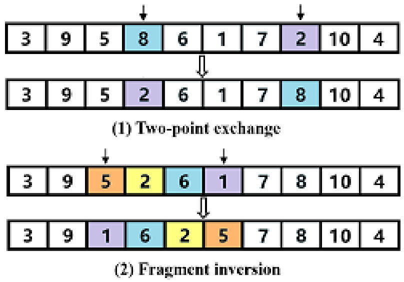

Fig. 4. Graphic explanation of mutation.

The following describes the exhaustive search algorithm, assuming the total number of BSs is N.

- Use the function *perms*([1 2 ··· (N - 1)]) in MATLAB to obtain all possible permutations of a total number of (N - 1) BSs. Then a matrix $\mathbf{T}$ with a dimension of (N - 1)! × (N - 1) is returned. Because the drone is always required to take off and fly back to the gateway node, the integer assigned to the gateway node is not taken into consideration when the random arrangement is applied, since it is always at the end of any qualified chromosome.
- Add an integer N, the assigned number of gateway node, to the end of each row in matrix $\mathbf{T}$ to ensure that the drone will finally return to the gateway node.
- For matrix $\mathbf{T}$, each row represents a possible closed path, starting from and ending at the gateway node. The total length of each route is calculated, and the results are stored in matrix $\mathbf{X}$.
- Find the smallest value from all elements in matrix $\mathbf{X}$, which is the searching space.
- Return the shortest path length obtained and the computation time consumed in order for the following comparation.

B. Case 1: Original Case

In this scenario, we apply GA to calculate the shortest trajectory length for a total number of [7, 8, 9, 10, 11, 12] BSs. When we deploy more BSs, the size of the disaster area also increases, which are [1000 × 1000, 1100 × 1100, 1200 × 1200, 1300 × 1300, 1400 × 1400, 1500 × 1500] m^2. Their corresponding population sizes are [100 200 300 400 500 600] and generation numbers are [50 60 70 80 90 100]. P_c and P_m are set as 0.90 and 0.08 respectively, which are kept constant for all cases. We compare the performance of GA and exhaustive search at the same number of BSs in terms of the shortest path length obtained and the computational time preseneted in Figure 5 and 6 respectively.

From the experimental results, a much higher working efficiency of GA can be observed, especially when the number of BSs is getting larger. Figure 5 shows that our proposed algorithm has the same performance as the exhaustive search method in obtaining the shortest path lengths. From Figure 6, it can be seen that, for fewer BSs (7 to 10 BSs),the exhaustive search takes less time to find the results as compared to time taken by our proposed algorithm. However, as the

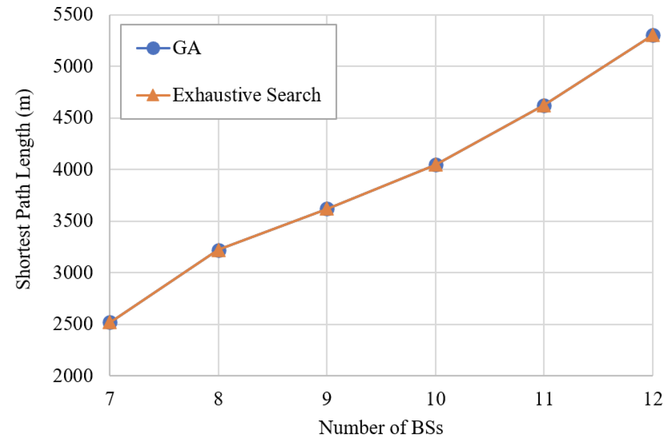

Fig. 5. Comparison between our proposed GA and exhaustive search in terms of the shortest path length obtained.

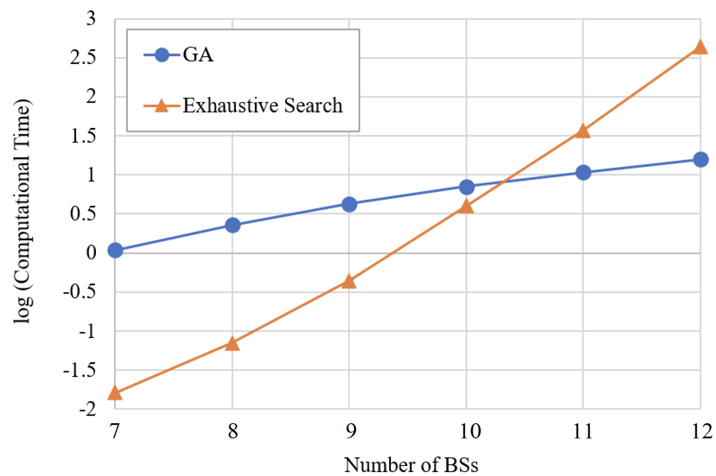

Fig. 6. Comparison between our proposed GA and exhaustive search in terms of the computational time.

number of BSs increases, our algorithm requires much less time to return the results and exhaustive search may not be practicable anymore. When there are 12 BSs in the disaster area, the computational time required for exhaustive search to obtain results is approximately 30 times longer than that for our proposed GA. Therefore, under real scenarios, GA is more efficient and practical than exhaustive search to solve optimization related problems.

C. Case 2: BSs with Priority

In this case, time-sensitive BSs are assumed to exist. It is assumed that the shortest path is computed before the drone takes off and it does not know in advance which BSs have priority. It means, the drone will only know whether the BS is prioritized or not after arriving within its coverage. The drone starts following trajectory generated before until it gets to the priority BS.

In simulation, it is assumed that a quarter of BSs, chosen randomly, have priority denoted as $\{T_{p1}, \cdots, T_{pc}, ..., T_{p\frac{N}{4}}\}$ with time limitations $\{l_{p1}, ..., l_{pc}, ..., l_{p\frac{N}{4}}\}$ assigned to them accordingly, where $l_{pc} \in (t_{pc,N}, t_{max})$ such that $t_{max} = \max\{t_{1,N}, t_{2,N}, \cdots, t_{N-1,N}\}$ and $t_{pc,N}$ indicates the time needed for the drone to fly back to the gateway T_N from the pc^{th} prioritized BS T_{pc}.

Six situations are proposed based on different actions of drone after arriving at the prioritized BSs and whether GA would be reapplied to generate the shortest trajectory among the unvisited BSs or not. After reaching the time-sensitive BSs, the drone can either directly return to the gateway node (Situation 1 & 2), or check if it can visit extra BSs within the expiration time among unvisited ones following the order of the pre-planned trajectory (Situation 3 & 4), or check among all remaining BSs (Situation 5 & 6). Situations with same action after arriving at BSs with priority differ from each other in terms of whether GA would be used to re-plan the route or not. Their corresponding detailed descriptions are as follows:

Algorithm: BSs with Priority

(1) Generate the shortest trajectory T using GA, taking off from and return to the gateway node T_N through BS T_1, T_2, $\cdots$, T_{N-1}.

(2) **for** c=1:N **do**

calculate $t_{c,N}$ for BS T_c using formula (1).

end for

(3) Randomly select $\frac{1}{4}\times N$ BSs, $\{T_{p1}, \cdots, T_{p\frac{N}{4}}\}$, to give priority, and then assign them corresponding time limitations.

(4) The drone begins to travel along trajectory T.

(5) **if** the drone arrives at BS in $\{T_{p1}, \cdots, T_{p\frac{N}{4}}\}$ **then**

Situation 1:

(6) it will directly fly back to the gateway node T_N

(7) the drone will continue flying along the route T from the next unvisited BS in the trajectory

Situation 2:

(6) same as *Step* 6 in Situation 1

(7) applying GA to re-plan the shortest path among all unvisited BSs

(8) the drone will follow the newly generated shortest path

Situation 3:

(6) checking if the drone can visit extra BSs within the expiration time among unvisited BSs following the order of T

(7) the drone will continue visiting the next remained BSs along T

Situation 4:

(6) same as *Step* 6 in Situation 3

(7) GA will be applied to re-generate the shortest route

(8) the drone will continue visiting the next unvisited BSs, following the lately generated shortest path in (7)

Situation 5:

(6) the drone will check whether it can visit more BSs within the time limitation among all unvisited BSs, i.e., not following the order of T

(7) it will continue flying to the next unvisited BSs in the order of T

Situation 6:

(6) same as *Step* 6 in Situation 5

(7) GA will be operated to re-generate the shortest trajectory among unvisited BSs

(8) the drone will continue its task following the revised trajectory obtained in (7)

end if

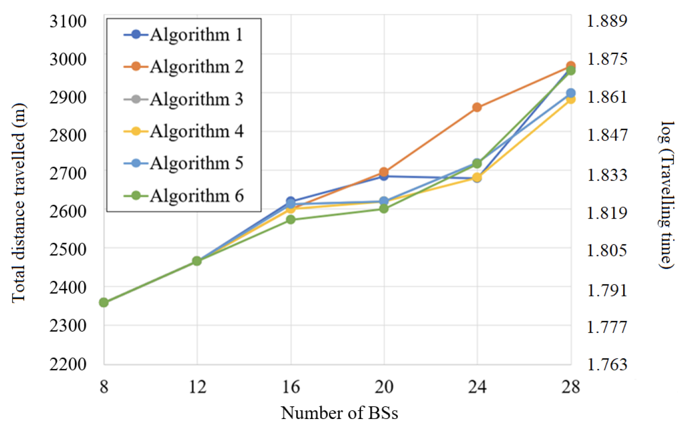

Fig. 7. Situations comparison for case 2 in terms of the total length travelled and the total time spent on travelling.

These algorithms are simulated through MATLAB with a total number of [8 12 16 20 24 28] BSs. The percentage of BSs with priority is considered as constant, which is 25% in this project. Assume the velocity of drone, v, is constant and equal to 40m/s. If GA is applied to re-generate the shortest path, the genetic parameters are assigned values as: population size=[200 600 1000 1400 1800 2200], generation number=[60 100 140 180 220 260], P_c=0.90 and P_m=0.08. Figure 7 reveals the experimental results.

From Figure 7, conclusions can be drawn about the performance of different situations for solving the problem of time-sensitive BSs. When the number of BSs in the disaster area is relatively small (8 and 12 BSs in this experiment), different situations can obtain the same results. This shows that due to the small number of BSs, the distances among them are too large to visit extra BSs within the expiration time. When the number of BSs gets larger, performance differences among the methods proposed are revealed. Situation 1 and 2 seem to have the lowest efficiency, and the second one is even worse because of travelling approximately 8% more distances It is so, probably because of the extra uncertainty introduced by the re-application of GA. When there are more BSs in the dedicated area, the performance of Situation 4 exceeds that of Situation 6, saving approximately 5% time spent on travelling. That may be due to the increased distortion introduced to the original trajectory by taking all possible unvisited BSs into consideration, while Situation 4 maintains the initial structure with a higher extent.

Therefore, when the BSs in the dedicated zone are rare, Situation 1 is preferred because of its lowest computation complexity without worsening the results. With the increase of the number of BSs, Situation 6 has a better performance by visiting more possible BSs on the way back while flying to the gateway node from the BS with priority. When the scenario is very complicated, Situation 4 performs best due to the highest efficiency without too much distortion on the original route.

V. Conclusions

In this paper, we apply GA to solve drone trajectory optimization problems for post-disaster communication. We first introduce our system design and basic assumptions, and then present how our proposed GA is implemented. We compare the results of our approach against the exhaustive search method used as benchmark and results confirm the accuracy and efficiency of proposed solution. We also show that our algorithm can carry out path planning with the existence of time-sensitive BSs, with different situations proposed and discussed. In the future, we plan to consider the energy requirement of the drone in the algorithm design.

Acknowledgment

This research was funded by EPSRC Global Challenges Research Fund - the DARE project - EP/P028764/1.

References

[1] S. Fu, L. Han, Y. Tian and G. Yang, "Path planning for unmanned aerial vehicle based on genetic algorithm," 2012 IEEE 11th International Conference on Cognitive Informatics and Cognitive Computing, Kyoto, 2012, pp. 140-144.

[2] M. Erdelj, E. Natalizio, K. R. Chowdhury and I. F. Akyildiz, "Help from the Sky: Leveraging UAVs for Disaster Management," in IEEE Pervasive Computing, vol. 16, no. 1, pp. 24-32, Jan.-Mar. 2017, doi: 10.1109/MPRV.2017.11.

[3] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz and Q. Ni, "Drone-Aided Communication as a Key Enabler for 5G and Resilient Public Safety Networks," in IEEE Communications Magazine, vol. 56, no. 1, pp. 36-42, Jan. 2018, doi: 10.1109/MCOM.2017.1700451.

[10] A. Sonmez, E. Kocyigit and E. Kugu, "Optimal path planning for UAVs using Genetic Algorithm," 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, 2015, pp. 50-55.

[4] S. A. Gautam and N. Verma, "Path planning for unmanned aerial vehicle based on genetic algorithm and artificial neural network in 3D," 2014 International Conferenceon Data Mining and Intelligent Computing (ICDMIC), New Delhi, 2014, pp. 1-5.

[5] S. Gao, X. Jiang and K. Tang, "Hybrid algorithm combining ant colony optimization algorithm with genetic algorithm," Proceedings of the 26th Chinese Control Conference, Zhangjiajie, 2007, pp. 701-704.

[6] H. Daryanavard and A. Harifi, "UAV Path Planning for Data Gathering of IoT Nodes: Ant Colony or Simulated Annealing Optimization," 2019 3rd International Conference on Internet of Things and Applications (IoT), Isfahan, Iran, 2019, pp. 1-4.

[7] S. B. Jasna, P. Supriya and T. N. P. Nambiar, "Application of game theory in path planning of multiple robots," 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, 2017, pp. 147-151.

[8] G. Vahdati, M. Yaghoubi, M. Poostchi and M. B. N. S., "A New Approach to Solve Traveling Salesman Problem Using Genetic Algorithm Based on Heuristic Crossover and Mutation Operator," 2009 International Conference of Soft Computing and Pattern Recognition, Malacca, 2009, pp. 112-116.

[9] R. Lakshmi and K. Vivekanandan, "Performance analysis of a novel crossover technique on permutation encoded genetic algorithms," 2014 International Conference on Advances in Engineering and Technology (ICAET), Nagapattinam, 2014, pp. 1-4.

[11] M. Cakir, "2D path planning of UAVs with genetic algorithm in a constrained environment," 2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), Istanbul, 2015, pp. 1-5

[12] Y. A. Sambo, P. V. Klaine, J. P. B. Nadas and M. A. Imran, "Energy Minimization UAV Trajectory Design for Delay-Tolerant Emergency Communication", 2019 IEEE International Conference on Communications Workshops (ICC Workshops), Shanghai, China, 2019, pp. 1-6.

[13] Q. Wu, Y. Zeng and R. Zhang, "Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks," in IEEE Transactions on Wireless Communications, vol. 17, no. 3, pp. 2109-2121, March 2018.

[14] S. Sharma and K. Gupta, "Solving the traveling salesmen problem through genetic algorithm with new variation order crossover," 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), Udaipur, 2011, pp. 274-276, doi: 10.1109/ETNCC.2011.6255903.

[15] S. N. Sivanandam and S. N. Deepa, Introduction to Genetic Algorithms. Springer, Berlin, Heidelberg, 2008.