to appear in IEEE Vehicular Technology Conference (VTC-Fall) 2018

Time Optimal Multi-UAV Path Planning for Gathering ITS Data from Roadside Units

Hamidullah Binol*, Eyuphan Bulut†, Kemal Akkaya*, and Ismail Guvenc‡
*Department of Electrical and Computer Engineering, Florida International University, Miami, FL
†Department of Computer Science, Virginia Commonwealth University, Richmond, VA
‡Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC
Email: {hbinol, kakkaya}@fiu.edu, ebulut@vcu.edu, iguvenc@ncsu.edu

***Abstract*—In this paper, we address the problem of path planning for multiple unmanned aerial vehicles (UAVs), to gather data from a number of roadside units (RSUs). The problem involves finding time-optimal paths for multiple UAVs so that they collectively visit all the RSUs, while also exchanging information at their own point when they fly from a starting point to the final location. We solve the problem by applying modified evolutionary methods based on genetic algorithm (GA) and harmony search (HS). The modified search methods seek to determine the overall shortest path utilizing various evolutionary operators regarding each UAV which has identical properties at the start location. Numerical results are introduced under different scenarios and the performances of the proposed algorithms are evaluated.**

***Index Terms*—Drone, evolutionary computation, genetic algorithm, harmony search, intelligent transportation system (ITS), path planning, RSUs, Unmanned aerial vehicles.**

I. Introduction

Use of unmanned aerial vehicles (UAVs) for intelligent transportation system (ITS) applications can improve efficiency of ITS services and reduce deployment costs [1], [2]. In particular, one of the applications of UAVs in ITS domain involves intelligent traffic management exploiting them as flying roadside units (RSUs). Such kind of operation demands multiple UAVs to fly in coordination to accomplish a specific mission. For example they can gather data from ground vehicles, or periodically from other ground RSUs that gather/store data from vehicles passing by. In rural areas, connecting such RSUs to a network may be challenging, and hence UAVs can be an effective way of collecting data from them.

Deployment and trajectory optimization of single/multiple UAVs have been considered in various different contexts in the recent literature. In [3], trajectory planning has been studied for multiple drones to minimize delay in search and reconnaissance applications, while [4] explores trajectory planning within a smart city surveillance framework. An energy-efficient trajectory optimization technique for a single drone, where the drone identifies the ideal locations to stop and collect data from a cluster of sensors, is proposed in [5]. The proposed method clusters the sensors in the first step, and subsequently applies the traveling salesman solution into the cluster centers, to find the optimum drone trajectory.

In [6], penalized weighted k-means and particle swarm optimization techniques are used to find optimum fixed locations of docking stations, so that UAVs can reach them at a reasonable time in case of need. In [7], [8], dynamic programming has been used to find optimum trajectory of a UAV to maintain good connectivity with a cellular network, while [9] uses deep reinforcement learning to address a similar problem. Genetic algorithm is utilized in [10], [11] to jointly find optimum deployment locations and interference coordination parameters for LTE-Advanced UAVs while serving mobile users.

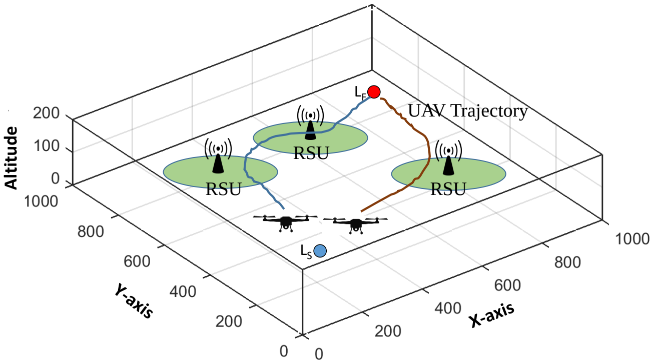


Fig. 1: Three dimensional view of a sample scenario with three RSUs and two UAVs that require to travel from a starting location (L_{S}) to a final point (L_{F}).

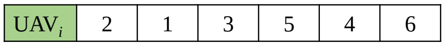
To our best knowledge, multi-UAV trajectory optimization problem for gathering data from a number of RSUs has not been addressed in the literature. In this paper, as summarized in Fig. 1, we tackle the problem of finding time-optimal paths for multiple UAVs so that they collectively visit the coverage areas (not necessarily the exact RSU locations) of all the RSUs, when they fly from a starting point to the final location. In order to solve this problem, we utilize modified evolutionary methods based on genetic algorithm (GA) and harmony search (HS), which seek to determine the overall shortest path utilizing various evolutionary operators regarding each UAV. In particular, UAVs have identical properties at the start location, and at the end of the trip we consider that: 1) each UAV should have a similar path length, or 2) the total trajectory distance of all UAVs should be as short as possible. Numerical results show that our proposed multi-UAV trajectory planning approach can effectively span multiple RSUs to collect data in a time-efficient manner.

II. System Model and Problem Formulation

A. Problem Description

The classic path planning problem requires solving for a route between a source and a destination, to visit a given set of points while avoiding forbidden area(s). In our scenario, the critical constraint is to visit *wireless coverage areas* of RSUs, in order to periodically gather information from them. Fig. 1 demonstrates an example scenario with three RSUs and

This work was made possible by NPRP grant # 9-257-1-056 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made here are solely the responsibility of the authors.



UAV_i	2	1	3	5	4	6

Fig. 2: A sample chromosome structure for assignment and ordering problems of path planning.

two UAVs. The aim of this paper is to design an algorithm that finds an optimal route between a given starting and a final location within a certain mission time for each UAV. The hypotheses handled to illustrate the problem are as follows:

- There is no overlap between the coverage areas of RSUs.
- The starting and final positions are the same for each UAV, and neither is in the range of any RSU.
- Each RSU should be visited by a single UAV for information exchange. It is not mandatory for a UAV to reach the exact RSU location if data exchange is finalized at a point within the coverage area of an RSU. We consider the coverage range of an RSU to be 100 meters.
- UAVs involved in the problem have identical physical characteristics, e.g., battery capacity. Accordingly, the mission time should be as close as possible for each UAV (as will be captured by (2)), or the total mission time should be minimized (as will be captured by (8)).

Under these assumptions, the proposed solution includes two main steps. The problem is considered to be the multiple single-UAV path-planning problems in the first step. The *first step* is actually an assignment problem to identify *"which RSU will be visited by which UAV"* and determine the visiting order of RSUs for each UAV. The basic of this step is to solve fixed-terminal open multiple Traveling Salesmen Problem (M-TSP) [12] which is a variant of the popular traveling salesman problem (TSP) [13]. The *second step*, on the other hand, controls the provision of the data exchange constraint. After the point at which the necessary data exchange is done, this step will lead to the next determined RSU.

B. Problem Formulation

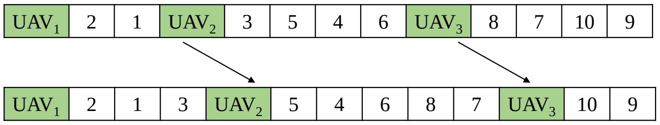
This subsection formally describes the problem addressed in this paper. We consider a network of N RSUs by the three dimensional (3D) geographical position model denoted as (x_n, y_n, h_n), where $n = 1, ..., N$. There are K UAVs to be traveled with a constant speed to collect the data of the RSUs. We denote the time-dependent location of *k*th UAV as $(x_k(t), y_k(t), h_k(t))$. The UAV motion distance *d* from a location to nth RSU thus can be expressed as follows:

$$d_{kn} = \sqrt{(x_k(t) - x_n)^2 + (y_k(t) - y_n)^2 + (h_k(t) - h_n)^2} \ .$$

For simplicity and without loss of generality, we assume that the altitude of the N RSUs and K UAVs are identical ($h_n = h_k$). Each UAV has one data transmission interface which is adopted to download the ITS data while flying from a start location L_S, to a final location L_F. The two fixed terminal points are same for each UAV and can be considered that they represent the locations of docking stations where a UAV can charge its battery.

The trajectory distance of *k*th UAV is T_k, with $0 < T_k < T_k^{\max}$, where $T_k^{\max}$ denotes the maximum travel capacity of the *k*th UAV without battery replenishment. To provide collaborative and time-efficient path planning, we consider the average UAV flying time cost factor, defined as:

$$\bar{T} = \frac{1}{K}\sum_{i=1}^{K} T_i \ . \tag{1}$$



UAV_1	2	1	UAV_2	3	5	4	6	UAV_3	8	7	10	9
UAV_1	2	1	3	UAV_2	5	4	6	8	7	UAV_3	10	9

Fig. 3: An illustration of the operator of altering RSU allocations between UAVs.

The objective of our trajectory optimization and RSU assignment task is to minimize the variance of the population including the mission distances of all UAVs:

$$\min \quad \frac{1}{K}\sum_{i=1}^{K}\left(T_i - \bar{T}\right)^2 \tag{2}$$

$$\text{s.t.} \quad \sum_{k} c_k^n \geq 1, \forall n \tag{3}$$

$$\sum_{n} c_k^n = 1, \forall k \tag{4}$$

$$\big(x_k(0), y_k(0)\big) = (L_\text{S}.x, L_\text{S}.y) \tag{5}$$

$$\big(x_k(T), y_k(T)\big) = (L_\text{F}.x, L_\text{F}.y) \tag{6}$$

where,

$$c_k^n = \begin{cases} 1, & \text{if } k\text{th UAV visit } n\text{th RSU,} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The main constraint for a UAV based on (3) is that it should visit the coverage area of at least one RSU. Moreover, we require each UAV to travel a minimum amount of predetermined distance within an RSU's coverage area (not shown in the constraints above), during which we assume that the RSU data will be downloaded at the UAV.

As an alternative to (2), we also define another cost function considering the problem of minimizing the total path length by all UAVs as follows:

$$C_\text{PL} = \min \sum_{i=1}^{K} T_i, \tag{8}$$

where still the constraints given by (3)−(6) are considered.

C. Evolutionary Operators

To determine the visiting order of RSUs, a chromosome design is presented. Each RSU has a unique label that is randomly assigned to a UAV at initial solution. The example chromosome is depicted in Fig. 2. According to the given chromosome structure, the mission of the i^{th} UAV is to visit six RSUs in total and reach the destination. In this scenario, after the launch, the UAV will fly to the RSU labeled 2, then respectively communicate with RSUs of labels $(1 - 3 - 5 - 4 - 6)$.

The proposed evolutionary path optimizer employs three crossover operators and one mutation operator. The crossover operators manipulate the chromosome of a UAV and mutation operator alters the RSU allocations between UAVs. The features of these operators are summarized as follows.

Flip the bits: This operator flips the order of selected bits in a chromosome representation of path. The illustration in Fig. 4 demonstrates a possible effect of this evolutionary operator on a sample path planning. The input chromosome (upper left one) represents the visiting sequence of RSUs for UAV_i, which is $(2 - 1 - 3 - 5 - 4 - 6)$. A sub-segment of the input

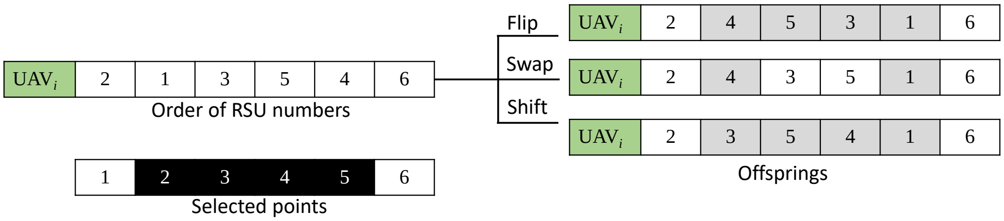


Fig. 4: Evolutionary operators: Flip, Swap, and Shift. Drawing demonstrates possible effects of the evolutionary operators on a candidate path. The original (input) path of a UAV is given at the left. Randomly-selected segment of input chromosome to utilize the operators is highlighted in black color.

chromosome, starts from the 2$^{\text{nd}}$ bit and ends at the 5$^{\text{th}}$ in this case, is randomly chosen before application of any operator. Then the output chromosome is produced by flipping the order of RSU labels contained in the sub-segment.

Swap the bits: Changes the orders of two randomly picked points in path (See Fig. 4).

Shift the bits: Moves the first gene in a selected sub-chromosome to the end of this random selection (See Fig. 4).

Changing the break points: Mutates the whole planning by shifting the RSU assignments. In an example given by Fig. 3, the input chromosome (upper one) of the UAV$_2$ gives the first RSU (labeled as 3) to UAV$_1$ and takes the RSUs 8 and 7 from UAV$_3$. This operator increases the probability of finding the optimum solution. At initial step of search process, $K-1$ break points are randomly chosen among the $(2, ..., N-1)$, inclusive. Hence, K dummy assignments are established for K UAVs beforehand.

III. Multi-UAV Path Planning for Visiting RSUs

A. Genetic Algorithm

Genetic Algorithm (GA) as a population-based metaheuristic optimization method that exploits concepts from biology, genetics, and evolution [14], [15]. The classical GA first produces an initial population, then it evolves the solutions through three actions, i.e., selection, crossover, and mutation. In this paper, we utilize the following operators instead of classical ones: flip, swap, and shift. The pseudo-code of the GA that is used in this paper is given in Algorithm 1.

Algorithm 1: GA based Path Planning for UAV

```
create the initial population randomly
evaluate the fitness value of each member using (2) or (8)
while not stopping condition do
    split population randomly into sub-groups and select
      the best member for each of them;
    hold the best candidate and manipulate the rest of the
      new group exploiting operators in Section II-C;
end
return the best solution in the population
```

B. Harmony Search

As a relatively new population-based optimization technique, the Harmony Search (HS) algorithm has an analogy with music improvisation process where musicians improvise the pitches of their instruments to obtain a better state of harmony [16]. In the HS algorithm, each solution and a set of these solutions are respectively called a harmony and harmony memory (HM). The improvisation process is mimicked in each variable selection of the basic HS algorithm in three rules: (*i*) picking any value from the memory; (*ii*) picking an adjacent value from the memory; and (*iii*) picking a random solution from the possible value range. Using harmony memory consideration rate (HMCR), and pitch adjusting rate (PAR), the pseudo-code of the HS that is proposed in our paper is given in Algorithm 2.

Algorithm 2: HS based Path Planning for UAV

```
create the initial population randomly
evaluate the fitness value of each member using (2) or (8)
while not stopping condition do
    split population randomly into sub-groups and select
      the best member for each subgroup;
    if $\mathrm{rand}() < HMCR$ then
        pick the best value from harmony memory
        if $\mathrm{rand}() < PAR$ then
            apply the flip operator
        else
            apply the swap operator
        end
    else
        generate random RSU assignments to all UAVs
    end
end
return the best solution in the harmony memory
```

IV. Numerical Results

In this section we present results for the considered path planning techniques, using a map that corresponds to Doha, Qatar. The road network of Doha is obtained by the OpenStreetMap [17] which is a cooperative project to build free editable maps. Fig. 5 shows the road network with RSU deployments utilized in the experiments for major roads in north-west of Doha. The study area is approximately 8,898 meters in longitude and 8,684 meters in latitude. Our experiments are conducted on a machine with an Intel Core i5-2410M CPU at 2.30GHz and 8GB DDR-III RAM. Some of the challenges in simulations include determining the reasonable parameter values such as *HMCR* and *PAR* values in HS, and population/sub-group size in both HS and GA. We set the key parameters of HS as *HMCR* $= 0.8$ and *PAR* $= 0.4$, which are close to their values used in the recent literature [18].

In this work, the population size and the actual number of individuals in a sub-group are experimentally selected as 80 and 8, respectively. Since offsprings are created with a non-random process in suggested GA, the size of sub-groups is more important than that of HS. As emphasized

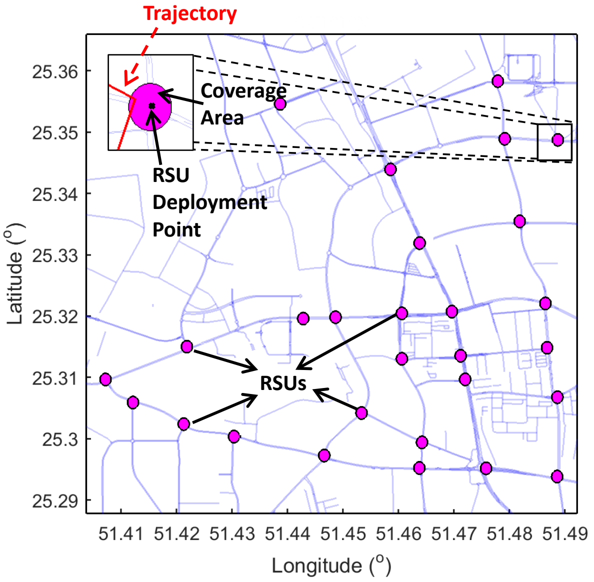
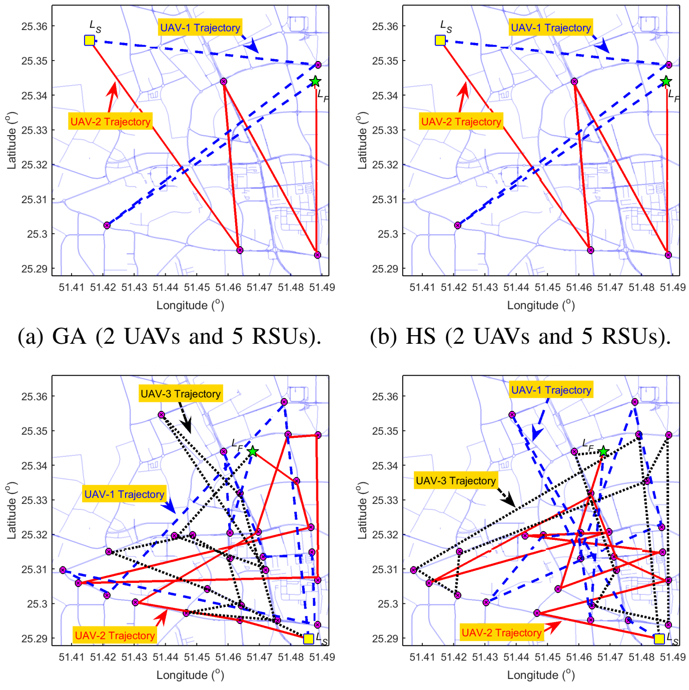


Fig. 5: Illustration of 28 RSUs deployed at major intersections of road network at North West of Doha, Qatar. An example for changing direction of a UAV within an RSU coverage area, after pulling the data from the RSU, is also illustrated.



(a) GA (2 UAVs and 5 RSUs).

(b) HS (2 UAVs and 5 RSUs).

(c) GA (3 UAVs and 28 RSUs).

(d) HS (3 UAVs and 28 RSUs).

Fig. 6: Trajectories by GA and HS on two different scenarios.

in Algorithm 1, the first member of each new generation is the best parent of current sub-group. The subsequent three children are generated by using flip, swap, and shift operators. The order of RSUs in the fifth member is the same as the best member, but RSU assignments between UAVs are randomly regenerated in this step. And finally, the last three members are provided with the handling of evolutionary operators and re-assignment together.

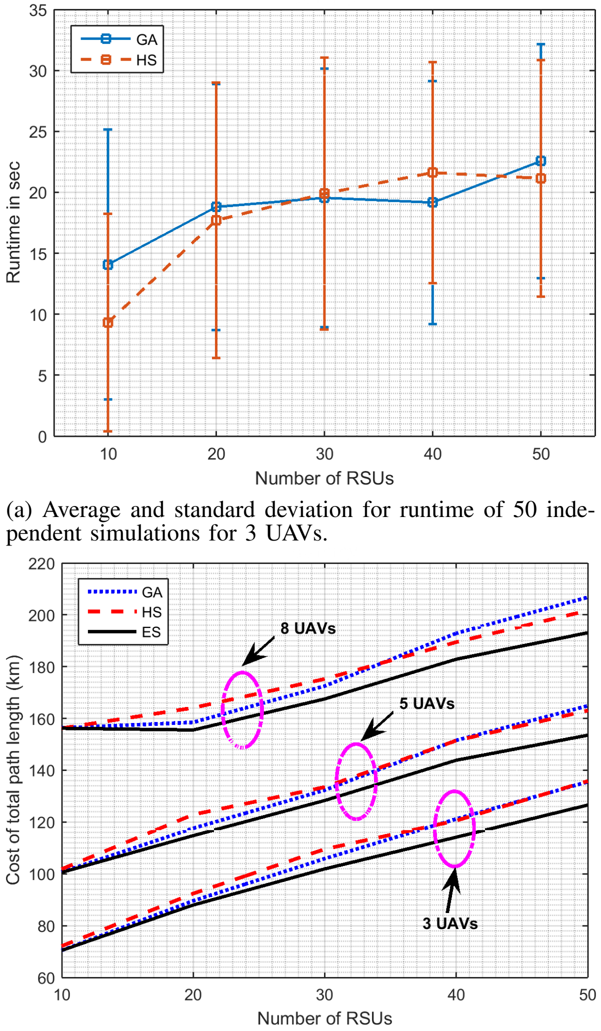
For simplicity, in the first scenario we have adopted 2 UAVs for the mission over the area where 5 RSUs are deployed. As depicted in Figs. 6(a)-(b), the same paths are obtained with GA and HS with same costs calculated by (2); however, HS reached the solution significantly faster than GA, i.e., needed iterations to reach convergence are respectively 2 and 137. An interesting observation is, the found paths seem longer than can be and have a zigzag style. This is because the optimization constraint aims to maintain equal mission time interval for all UAVs. By increasing the number of UAVs to 3 and RSUs to 28 in Figs. 6(c)-(d), different path plans are generated by GA and HS. We notice that the HS outperforms the GA in terms of cost-effectiveness and convergence speed. Specifically, the minimum cost value obtained by HS was 3.2258×10^{-06} at 1210 iterations, while it was 8.3259×10^{-06} at 1600 iterations by GA. Note that in all maps, UAVs may make sharp turns at the stop points, which can be avoided with more granularity added to the problem environment.

The performance of GA and HS techniques in terms of mean and standard deviation of execution times are plotted in Fig. 7(a). We report the runtime for the MATLAB simulations. The analysis shows that HS is better than GA in the lower-level complexities of the framework, e.g, for the situation where 10 and 20 deployed RSUs are searched. On the other hand, GA can establish faster decision making on higher number of RSUs, particularly around 10% faster for 40 RSUs. Furthermore, in Fig. 7(b), we exhaustively enumerate all possible path combinations on the Doha map for various number of UAVs and RSUs and then compare them against our path planning procedures in terms of cost performance given by (8). We again record the mean of 50 runs and find that for relatively lower number of RSUs, our algorithms produce paths that are very close to the optimal paths. With the increasing number of UAVs, the total path lengths generated by GA and HS are much longer than that by the exhaustive search (ES). Further analysis shows that GA achieves better performance under 30 RSUs, whereas HS noticeably outperforms the GA in case of 50 RSUs for 5 UAVs, and 40 and 50 RSUs for 8 UAVs. We believe that HS generates better solutions when the search space is large, thanks to its random assignment of RSUs to UAVs that happens with a probability of ($1 - \mathit{HMCR}$), as illustrated in Line-13 of Algorithm 2.

Finally, we study the difference of the obtained trajectory distances produced by applying the two different objective functions introduced in (2) and (8). Fig. 8 shows the results for 2 UAVs and 5 RSUs found with an exhaustive search. We observe that both UAV$_1$ and UAV$_2$ share the same amount of RSUs for different costs. Nevertheless, UAV$_1$ should fly 24,028 meters for (2), but only 8,1261 meters for (8), while UAV$_2$ should travel 25,620 meters for (2) but 18,378 meters for (8) to accomplish the overall mission. Hence, we can say both system-wide and individual efficiency could be achieved for servings RSUs when using (8).

V. Conclusion

This paper introduces our findings on the path planning problem for multiple unmanned aerial vehicles (UAVs) for collecting data from a number of pre-deployed roadside units (RSUs) considering several scenarios. We assume that the battery capacity of a UAV and/or mission time are not adequate to visit all RSUs. We therefore formulated two problems: one assumes that each UAV has similar travel distance, while the other aims to optimize total path length. To solve these



(a) Average and standard deviation for runtime of 50 independent simulations for 3 UAVs.

(b) Average cost of total path search for 50 independent runs over various number of UAVs and RSUs.

Fig. 7: Comparisons of runtime and cost performance of GA, HS, and ES methods for different scenarios.

problems, we propose two modified metaheuristic-based approximate solutions with different evolutionary operators. Our results show that the proposed HS algorithm outperforms the GA in terms of cost-effectiveness when the problem becomes more complicated, and in convergence time when the search process is relatively straightforward.

References

[1] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang, and X. Shen, "Drone assisted vehicular networks: Architecture, challenges and opportunities," *IEEE Network Mag.*, 2018.

[2] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 22–28, 2017.

[3] D. Kim, L. Xue, D. Li, Y. Zhu, W. Wang, and A. O. Tokuta, "On theoretical trajectory planning of multiple drones to minimize latency in search-and-reconnaissance operations," *IEEE Trans. Mobile Comp.*, vol. 16, no. 11, pp. 3156–3166, 2017.

[4] H. Kim, L. Mokdad, and J. Ben-Othman, "Designing UAV Surveillance Frameworks for Smart City and Extensive Ocean with Differential Perspectives," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 98–104, Apr. 2018.

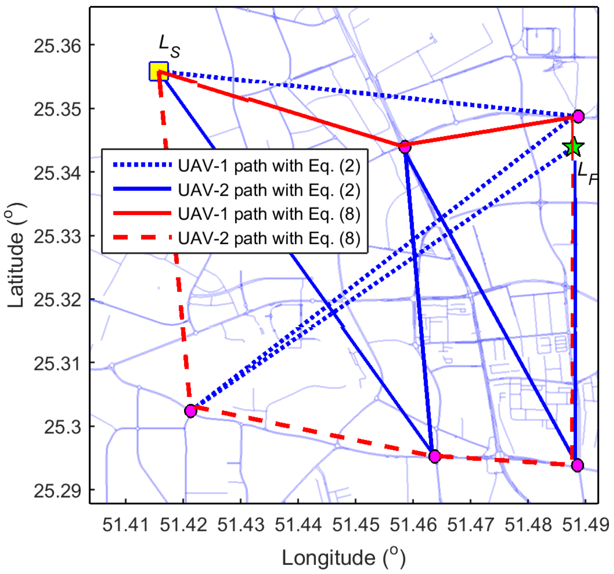


Fig. 8: Optimal trajectories for 2 UAVs and 5 RSUs with different cost functions in (2) and (8), respectively.

[5] M. B. Ghorbel, D. Rodriguez-Duarte, H. Ghazzai, M. J. Hossain, and H. Menouar, "Energy efficient data collection for wireless sensors using drones," in *Proc. IEEE Vehic. Technol. Conf. (VTC)*, Porto, Portugal, 2018, pp. 1–6.

[6] H. Ghazzai, H. Menouar, and A. Kadri, "On the placement of UAV docking stations for future intelligent transportation systems," in *Proc. Vehic. Technol. Conf. (VTC)*, 2017, pp. 1–6.

[7] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, 2017.

[8] E. Bulut and I. Guvenc, "Trajectory optimization for cellular-connected uavs with disconnectivity constraint," *Proc. IEEE Int. Conf. Commun. (ICC) Workshops*, May 2018.

[9] U. Challita, W. Saad, and C. Bettstetter, "Deep reinforcement learning for interference-aware path planning of cellularconnected uavs," in *Proc. Int. Conf. Commun. (ICC)*, Kansas City, MO, 2018.

[10] A. Kumbhar, I. Güvenç, S. Singh, and A. Tuncer, "Exploiting LTE-Advanced HetNets and FeICIC for UAV-assisted public safety communications," *IEEE Access*, vol. 6, pp. 783–796, 2018.

[11] A. Merwaday, A. Tuncer, A. Kumbhar, and I. Guvenc, "Improved throughput coverage in natural disasters: Unmanned aerial base stations for public-safety communications," *IEEE Vehic. Technol. Mag.*, vol. 11, no. 4, pp. 53–60, 2016.

[12] H. Ergezer and K. Leblebicioğlu, "3d path planning for multiple uavs for maximum information collection," *J. Intelligent & Robotic Syst.*, vol. 73, no. 1-4, pp. 737–762, 2014.

[13] L. Tang, J. Liu, A. Rong, and Z. Yang, "A multiple traveling salesman problem model for hot rolling scheduling in shanghai baoshan iron & steel complex," *European J. Operational Research*, vol. 124, no. 2, pp. 267–282, 2000.

[14] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[15] H. Binol, A. Bal, and H. Cukur, "Differential evolution algorithm-based kernel parameter selection for fukunaga-koontz transform subspaces construction," in *High-Performance Computing in Remote Sensing V*, vol. 9646. International Society for Optics and Photonics, 2015, p. 96460T.

[16] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer methods in Applied Mechanics and Eng.*, vol. 194, no. 36-38, pp. 3902–3933, 2005.

[17] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2018.

[18] X. Kong, L. Gao, H. Ouyang, and S. Li, "A simplified binary harmony search algorithm for large scale 0–1 knapsack problems," *Expert Systems with Applications*, vol. 42, no. 12, pp. 5337–5355, 2015.