Robotica: (2021) volume 39, pp. 1007–1022. © The Author(s), 2020. Published by Cambridge University Press.
doi:10.1017/S0263574720000867

Real-time Unmanned Aerial Vehicle Cruise Route Optimization for Road Segment Surveillance using Decomposition Algorithm

Xiaofeng Liu†*, Jian Ma‡, Dashan Chen¶ and Li-Ye Zhang§

†*School of Automotive and Transportation, Tianjin University of Technology and Education, Tianjin 300222, China*
‡*School of Civil Engineering, Suzhou University of Science and Technology, Suzhou 215011, Jiangsu, China*
E-mail: 9764634@qq.com
¶*School of Railway Transportation, Shanghai Institute of Technology, Shanghai 201418, China*
E-mail: logtop@126.com
§*Institute of High Performance Computing, A*STAR, Singapore 138632, Singapore*
E-mail: zhangly@ihpc.astar.edu.sg

(Accepted August 12, 2020. First published online: September 14, 2020)

SUMMARY
Unmanned aerial vehicle (UAV) was introduced for nondeterministic traffic monitoring, and a real-time UAV cruise route planning approach was proposed for road segment surveillance. First, critical road segments are defined so as to identify the visiting and unvisited road segments. Then, a UAV cruise route optimization model is established. Next, a decomposition-based multi-objective evolutionary algorithm (DMEA) is proposed. Furthermore, a case study with two scenarios and algorithm sensitivity analysis are conducted. The analysis result shows that DMEA outperforms other two commonly used algorithms in terms of calculation time and solution quality. Finally, conclusions and recommendations on UAV-based traffic monitoring are presented.

KEYWORDS: Unmanned aerial vehicle; Traffic monitoring; Route planning; Multi-objective optimization; Road segment.

1. Introduction

Due to its unique advantages of wide view, flexibility, and mobility, unmanned aerial vehicle (UAV) (a.k.a. drone) has been widely used in the fields of environmental monitoring, meteorology, power facility inspection, traffic monitoring, and so on.[1] In recent years, some countries have deregulated its flight policy. For example, in 2016, the Civil Aviation Administration of China approved that UAV indoor fights, UAV flights within visual line of sight, and UAV flights at the areas of few population were exempted from flight regulation and flight application permission.[2] In addition, the drone operations have become easier, and the drone sensing technology is maturing with a significant decrease in its price, which have promoted its applications in the fields of commercial business and scientific research.

* Corresponding author. E-mail: microbreeze@126.com

In the field of traffic monitoring, UAVs can be loaded with different sensors (e.g., infrared cameras, radars, and high-resolution cameras) to collect traffic information and monitor traffic infrastructure, such as road condition monitoring, traffic incident detection, bridge inspection, traffic information acquisition, and so on.[3–6] Besides the traditional manual observation, computer vision technology has been gradually used to conduct traffic monitoring, owing to its high accuracy and traffic analysis speed. Coifman et al. proposed a real-time computer vision system for vehicle tracking and traffic surveillance.[7] Puri stated a survey of UAV traffic surveillance and summarized its worldwide development situation.[8] Romanoni et al. compared two Monte Carlo algorithms for 3D vehicle trajectory reconstruction in roundabouts.[9] Liu et al. used a UAV to take pictures of the traffic accident scene and then adopted multi-stereo technology to reconstruct 2D and 3D traffic accident scene.[10] These studies put forward the UAV applications in the field of intelligent transportation systems.

In urban areas, most roads are installed with traffic detectors (e.g., loop detectors, video cameras, infrared cameras, and automatic vehicle identification readers) to conduct traffic monitoring. However, due to limited budget, some urban roads and several rural roads are not installed with any traffic detectors; thus, it is difficult to collect traffic information and conduct effective traffic management. In this context, UAVs are introduced to conduct traffic surveillance for these roads in this study. In real-life traffic monitoring, the number of UAVs available and UAV maximum flight distance are always limited; moreover, the number of monitored roads may be large, and UAV monitoring task is uncertain due to the unexpected traffic accident or congestion. Therefore, it is necessary to conduct real-time UAV cruise route optimization so as to find the optimal routes as soon as possible. Previous studies mainly focused on deterministic UAV cruise route planning and paid more attention to the modeling and optimization algorithms.

As for UAV cruise route optimization model, some studies extend and improve some optimization models such as traveling salesman problem,[11] multiple traveling salesman problem,[12] integer linear programming (ILP),[13] vehicle routing problem (VRP),[14] Voronoi diagram,[15] artificial potential field method,[16] partially observable Markov decision processing.[17] In ref. [14], UAV cruise route optimization problem was formulated as a VRP with time window and time precedence requirements, which has better ability to describe the complicated traffic monitoring problem. Moreover, non-dominated sorting genetic algorithm II (NSGA-II) was selected for optimization. In ref. [18], an optimization method combining VRP and ILP was proposed to conduct different UAV cruise tasks of various time windows for a fleet of UAVs. In ref. [19], the temporal relationship among UAV scheduling constraints and different UAV cruise tasks was considered, and a cooperative multi-task allocation problem model was proposed to describe more complex UAV cruise route planning.

As for UAV cruise route optimization algorithm, they are mainly grouped into two categories, that is, the centralized algorithms and the distributed algorithms. The most used centralized algorithms include dynamic programming, enumeration method, and branch-and-bound method. Moreover, more interests have been paid on heuristic and intelligent optimization algorithms such as genetic algorithm,[11] Tabu search,[12] particle swarm optimization,[20] and ant colony optimization.[21] It is easy to understand the centralized algorithms, and they can acquire the optimal UAV cruise routes theoretically. However, there is a huge computational challenge for them when the number of nodes increases obviously. Moreover, the self-adaptation and cooperative competition of the centralized algorithms are not good enough, and the obtained results by them may be ordinary in the dynamic condition.[22] While the distributed algorithms can grasp the information interaction among different UAVs, which is beneficial for solving the large size and dynamic UAV cruise route optimization problem. There are two types of distributed algorithms, the first type decomposes the original optimization problem into several sub-problems and optimizes these sub-problems simultaneously through adaptation and cooperation among each other; the second type emphasizes the individual's local awareness and response interaction to achieve the global self-organization. The main distributed algorithms include auction algorithm, contract net, dynamic distributed constraint optimization, and multi-objective evolutionary algorithm based on decomposition (MOEA/D). The distributed algorithms have some advantages, such as simple calculation, low calculation complexity, and good robustness; therefore, they are usually adopted to solve large size and complex task allocation problems.[23]

There are three important parts of real-time UAV cruise route optimization for road segment surveillance: (1) the road segment should not be simplified as a node, and it is a link which has a length from its start point to its end point. (2) The traffic monitoring task is uncertain, and traffic

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

accident or congestion may occur on some road segments randomly, and thus there are some unexpected/new monitoring targets. Therefore, it is essential to adjust UAV cruise routes in real time for quick surveillance response. (3) Optimization speed is important for real-time UAV cruise route planning. Most studies do not consider the length of road segment and always simplify it as a node, which is far away from the real-life situation. Few studies take this into account. Liu et al. proposed a UAV route planning optimization approach for road segment surveillance, and NSGA-II was adopted to find optimal UAV cruise routes.[24] However, the surveillance task is static and deterministic. In addition, NSGA-II is one of the centralized algorithms, which has high computational complexity; thus, it is difficult to acquire the optimal UAV cruise routes quickly.

In summary, previous studies have focused on static or deterministic UAV cruise route optimization problem, and centralized algorithms are always used for optimization, which is not helpful to obtain the optimal solutions as soon as possible. Therefore, this study aims at the real-time UAV cruise route planning problem for road segment surveillance, with the objectives of minimizing the number of UAVs used and minimizing the total cruise distance. Then, a decomposition-based evolutionary algorithm is proposed to find the optimal routes as soon as possible. The contributions of this research include the modeling of the real-time UAV rerouting for road segment surveillance, which considers the length of road segment, and the decomposition-based multi-objective evolutionary algorithm (DMEA), which improves the solution quality and optimization speed. Specifically, DMEA proposes an insertion comparison method to generate feasible UAV cruise route chromosome for road segment; additionally, DMEA adopts the order-based crossover operator and inverse mutation operator, to enhance the algorithm search ability.

The remainder of this study is organized as follows: In Section 2, the real-time UAV cruise route optimization problem for road segment surveillance is described, and the mathematical modeling is stated. Section 3 proposes a DMEA to conduct optimization. In Section 4, a case study is implemented, and the effectiveness of the proposed algorithm is verified through several simulated scenarios and algorithm comparisons. Finally, Section 5 presents the conclusions and recommendations.

2. Modeling

2.1. Problem description

For the sake of traffic surveillance, UAVs take flights according to the planned cruise routes, to acquire the traffic information and detect traffic incident/accident. Suddenly, traffic incident or traffic congestion occurs on some road segments, and these segments need aerial surveillance. In this condition, new/unexpected road segments appear and initial UAV cruise routes should be adjusted in real time. To better demonstrate the problem, Fig. 1 is presented. In Fig. 1, a road segment is denoted as a link, and each link has its start and end nodes. In this study, the critical link is defined as the link in which a UAV is currently monitoring, or a link which a UAV is heading to. When new links appear, critical links should be identified immediately, and their nodes are denoted as critical nodes. The goal of this optimization problem is to acquire the optimal routes as soon as possible.

As Fig. 1 shows, at initial time t_0, the road network has six links and a base, and several UAVs are available in this base. In addition, there are three possible UAV routes, that is, 0-4-3-1-2-5-6-0, 0-7-8-0, and 0-11-12-10-9-0. At time t_1, a new link from node 13 to node 14 appears. Meanwhile, the route 0-7-8-0 is finished, and the remaining routes are not finished. At time t_1, a UAV is flying from node 12 to node 10, and the other one is monitoring the link from node 4 to node 3. At this time, the link from node 10 to node 9 and the link from node 4 to node 3 are critical links. At time t_2, new UAV cruise routes are generated. In Fig. 1, at time t_1, the visited nodes are {7, 8, 11, 12}, the unvisited nodes are {1, 2, 5, 6, 13, 14}, and the critical nodes are {3, 4, 9, 10}. It should be noted that the possible UAV route (e.g., 0-4-3-1-2-5-6-0) cannot be reversed for flight navigation in the above model due to two reasons. The first reason is that a UAV sub-route (e.g., 0-4-3) of route 0-4-3-1-2-5-6-0 may have been visited at designated time (e.g., $t = 0.1$ h); in other words, the route direction cannot be changed. The second reason is that if the possible UAV route (e.g., 0-4-3-1-2-5-6-0) can be reversed for navigation, that is, 0-6-5-2-1-3-4-0, the critical link of this reversed route may change at designated time (e.g., the link from node 6 to node 5, not the link from node 4 to node 3); thus, the possible optimized UAV routes will be different after optimization. Generally, the

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

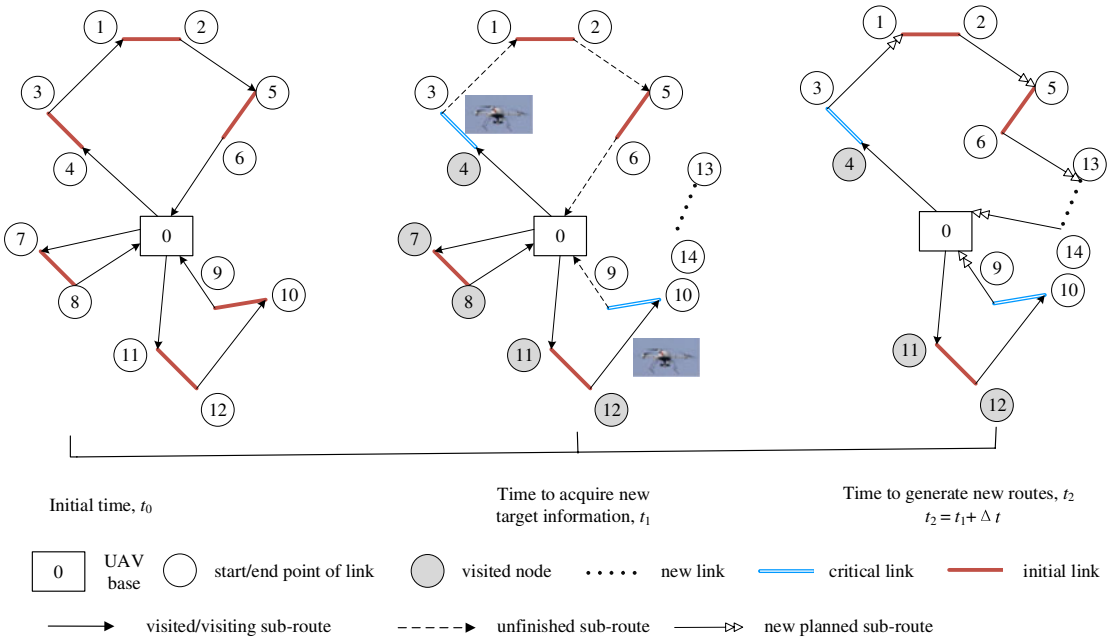

Fig. 1. Diagram of link re-routing.

visited, unvisited, and critical links should be considered together when planning UAV cruise routes. Since each link has its start and end nodes, UAV cruise route planning problem can be converted into a node-visiting optimization problem. As for the new optimized routes, they must originate from UAV base or critical nodes, go through unvisited nodes, and return to UAV base.

2.2. *Model definition*

At the time of new link(s) appearing, some nodes have been visited, and they must be deleted before modeling (e.g., node 7 and node 8 in Fig. 1). Then, the remaining ones are grouped as three categories: visited, unvisited, and critical nodes. Suppose that the set of critical and unvisited nodes is denoted as N_{cu}; the set of UAV base, critical and unvisited nodes are denoted as N_{0cu}; the set of critical, unvisited, and visited nodes is denoted as N_{cuv}; the set of UAV base, critical, unvisited, and visited nodes is denoted as N_{0cuv}; the set of UAVs is denoted as K; and the set of start and end nodes of road segments is represented by an integer number sequence, and they are denoted as N_s and N_E, respectively, additionally, $N_E = N_s + 1$; node pair (i, j) of each UAV route has flight distance and flight time, which are denoted as d_{ij} and t_{ij}, respectively. The number of UAVs available is denoted as N, and UAV maximum flight distance is denoted as D.

In real-life UAV flight, wind disturbance may have impact on the use of UAV energy and the UAV maximum flight distance. For idealized modeling, two optimization objectives are stated. The first objective is to minimize the total cruise distance, and the second objective is to minimize the number of UAVs used.

$$\text{Min}\, f_1 = \sum_{k\in K} \sum_{i\in N_{ocuv}} \sum_{j\in N_{ocuv}} x_{kij} d_{ij} \tag{1}$$

$$\text{Min}\, f_2 = \sum_{k\in K} \sum_{j\in N_{cuv}} x_{koj} \tag{2}$$

The constraints are listed as follows:

$$\sum_{j\in N_{cuv}} x_{koj} = 1, \quad \forall k \in K \tag{3}$$

$$\sum_{i\in N_{cuv}} x_{kio} = 1, \quad \forall k \in K \tag{4}$$

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

Equations (3)–(4) state that UAVs depart from and return to the base.

$$\sum_{k\in K}\sum_{j\in N_{ocu}} x_{kij} = 1, \quad \forall i \in N_{cu} \tag{5}$$

Equation (5) ensures that for each critical or unvisited node, only one UAV leaves it.

$$\sum_{k\in K}\sum_{i\in N_{ocu}} x_{kij} = 1, \quad \forall j \in N_{cu} \tag{6}$$

Equation (6) ensures that for each critical or unvisited node, only one UAV arrives at it.

$$\sum_{k\in K}\sum_{i\in N_{ocu}} x_{kih} = \sum_{k\in K}\sum_{j\in N_{ocu}} x_{khj}, \quad \forall h \in N_{cu} \tag{7}$$

Equation (7) ensures that for each critical or unvisited node h, its entering UAV must leave it.

$$\sum_{k\in K, j=i+1} x_{kij} = 1, \quad \forall i \in N_s \tag{8}$$

$$\sum_{k\in K, j=i-1} x_{kij} = 1, \quad \forall i \in N_E \tag{9}$$

Equations (8)–(9) ensure that the start and end nodes of one road segment are linked directly in one UAV cruise route.

$$\sum_{i\in N_{ocuv}}\sum_{j\in N_{ocuv}} x_{kij} \times d_{ij} \leqslant D, \quad \forall k \in K \tag{10}$$

Equation (10) states that UAV maximum flight distance should not be exceeded.

$$\sum_{k\in K}\sum_{j\in N_{cuv}} x_{koj} \leqslant N \tag{11}$$

Equation (11) states that the number of UAVs available should not be exceeded.

The decision variable is given by

$$x_{kij} = \begin{cases} 1, & \text{route}(i,j) \text{ is cruised by UAV } k \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

Equation (12) states that the decision variable equals to 1 when route (i, j) is cruised by UAV k; otherwise, it equals to 0.

The major contributions of the model include Eqs. (7) and (8)–(9). Equation (7) considers each critical or unvisited node and states that its entering UAV must leave it. Equation (7) guarantees that the new road segments, which have unvisited nodes, will be visited by UAVs. Equations (8)–(9) consider the length of a road segment and guarantee that one road segment will be visited in one UAV cruise route. In addition, it should be noted that the proposed model is not only applicable for UAV traffic surveillance but also suitable for other surveillance tasks, for example, vehicle allocation for logistics distribution, vessel deployment for maritime search, and so on.

3. Algorithm

The proposed model is a multi-objective optimization problem, and the commonly used solving methods include weighted sum and Pareto optimization. The weighted sum method assigns different weights to different objective functions and then linear combination is adopted to convert the multi-objective optimization problem into a single-objective optimization problem. However, it is difficult to determine the weights. Pareto optimality is widely used to solve the multi-objective optimization problem, and several evolutionary algorithms have been developed, such as the NSGA,[25] the Pareto archived evolutionary strategy,[26] and the strength Pareto evolutionary algorithm (SPEA).[27] These algorithms use archive or tournament pool to keep population diversity and adopt Pareto optimality to search an optimization front set for objective functions. As for SPEA, the overall

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

population is consisted of an internal and external population, and the external population is proposed to store non-dominated solutions at each generation. Then, all non-dominated solutions in the overall population are assigned fitness according to the number of solutions they dominated; moreover, the dominated solutions are assigned fitness worse than the worst fitness of any non-dominated solution. As for NSGA, it is criticized for specifying the sharing parameter, lacking the elitism and high computational complexity.[28] To overcome these disadvantages, NSGA-II is proposed, which conducts fast non-dominated sorting and crowded distance evaluation to choose elitist chromosomes, and uses tournament population to improve the diversity. Due to these advantages, NSGA-II is the most popular algorithm to solve the multi-objective optimization. In recent years, MOEA/D,[29,30] sub-population genetic algorithm II,[31] and hypervolume-based optimization[32] are developed. Additionally, some swarm intelligence algorithms are developed,[33–35] such as hybrid particle optimization algorithm, hybrid ant colony algorithm, and flower pollination algorithm.

Since the research is dynamic and multi-objective based, calculation time and solution quality are important. As for the calculation time, there is a huge computational challenge for the centralized algorithms, especially when the size of links is large, and it is preferred to use distributed algorithms for obtaining the optimized solutions as soon as possible. Compared with other algorithms, MOEA/D is one of the distributed algorithms and decomposes the optimization problem into several single optimization sub-problems and optimizes these sub-problems simultaneously; thus, MOEA/D algorithm has the advantages of solving complicated Pareto set shapes and low computational complexity,[36] which is beneficial to plan UAV cruise routes in real time. As for multi-objective optimization and solution quality, it is preferred to adopt Pareto technique and elitist strategy to approach the solution Pareto Fronts. Therefore, both the decomposition strategy, Pareto technique, and elitist strategy are adopted in this study, and a decomposition-based multi-objective optimization algorithm is proposed. As for the multi-objective decomposition, Tchebycheff approach is one of the commonly used methods, it can alter the weight vectors to deal with nonconcave Pareto Fronts.[29,36] Therefore, Tchebycheff approach is used to implement the decomposition strategy.

3.1. Tchebycheff approach

In Tchebycheff approach, a single objective optimization sub-problem is described as follows.

$$\text{Minimize } g(x|\lambda, z^*) = \max_{1\le i\le m} \{\lambda_i |f_i(x) - z_i^*|\} \tag{13}$$

$$\text{Subject to } x \in \Omega \tag{14}$$

where m is the number of objective functions, $\lambda = (\lambda_1, \ldots, \lambda_m)$ is the set of weight vectors, $f(x)$ is the objective function, z^* is the reference point, and Ω is the decision space. For all $i = 1, \ldots, m$, $\lambda_i \geqslant 0$, $\sum_{i=1}^{m} \lambda_i = 1$, and $z_i^* = \min\{f_i(x)|x \in \Omega\}$. When the optimal solutions of these sub-problems are obtained, this approach can provide a good approximation to the Pareto Front of the optimization problem.

3.2. Algorithm framework

The proposed algorithm consists of feasible UAV cruise route chromosome generation, UAV sub-route division for each chromosome, Tchebycheff value calculation for each chromosome, chromosome crossover and mutation, and neighborhood chromosome updating. The framework of this proposed algorithm is shown in Fig. 2. The main contributions of the proposed algorithm include feasible UAV cruise route chromosome generation, UAV sub-route division, and Tchebycheff value calculation for chromosomes, which are marked in gray in Fig. 2.

As Fig. 2 shows, the algorithm procedure can be summarized as follows. First, the algorithm generates initial UAV cruise route chromosomes and designates a reference point and the set of weight vectors. In this set, each weight vector corresponds to one chromosome. Second, the algorithm chooses T closest weight vectors for each weight vector, that is, each weight vector has T closest weight vectors or neighborhood chromosomes. The selection of closest weight vectors or neighborhood chromosomes is illustrated in Table I. In Table I, there are five UAV route chromosomes, and each chromosome has a weight vector and a neighborhood set, both of them are generated randomly so as to keep the search diversity. In addition, each chromosome has two objective values $\{f_1, f_2\}$. Taking the first UAV route chromosome as an example, the size of neighborhood set is three,

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

Table I. Selection of closest weight vectors or neighborhood chromosomes.

No.	Weight vectors for $\{f_1, f_2\}$	Neighborhood set	UAV route chromosome
1	0.645, 0.355	{1, 2, 5}	3-4-9-8-7-1-5-2-6
2	0.332, 0.668	{2, 3, 4}	3-5-1-6-2-7-8-9-4
3	–	–	–
4	–	–	–
5	0.496, 0.504	{2, 4, 5}	5-7-8-4-3-1-6-2-9

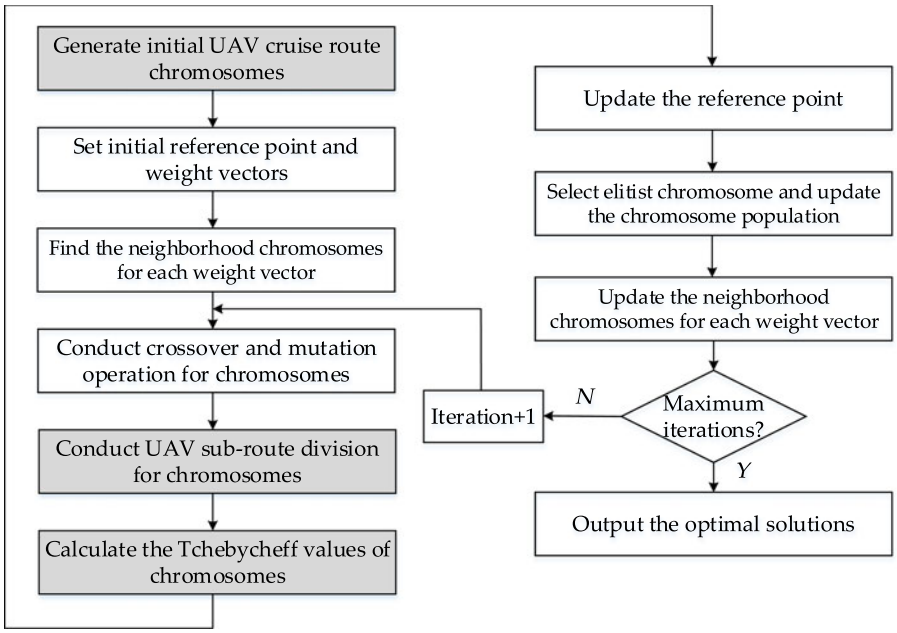

Fig. 2. Framework of the proposed algorithm.

and its neighborhood set is {1, 2, 5}. Once T is supposed to be two, the algorithm will select two neighborhood numbers from this set randomly, that is, 2 and 5 in the first row of Table I. This means that number "2" corresponds to the second chromosome, and number "5" corresponds to the fifth chromosome. In other words, the second and fifth chromosomes are the neighborhoods of the first weight vector or chromosome. Third, the algorithm conducts crossover and mutation operation for chromosomes so as to improve the population diversity and algorithm local search ability. Fourth, the algorithm conducts UAV sub-route division for chromosomes and calculates their Tchebycheff values. Then, the algorithm updates the new reference point and selects the elitist chromosome. Next, the algorithm compares the Tchebycheff values of elitist chromosome and its neighborhood chromosomes one by one. If the value of neighborhood chromosome is greater, it is substituted by the elitist chromosome so as to renew the chromosome population. Meanwhile, each weight vector's neighborhood chromosomes are updated. Finally, the algorithm implements the iteration optimization and outputs the optimized solutions.

Detailed steps of the proposed algorithm are given as follows.

Step 1: Initialization

Step 1.1 Generate an initial UAV cruise route chromosome population $\{x^1, x^2, \ldots, x^N\}$ (see Section 3.3) and calculate the objective values (i.e., the total cruise distance and the number of UAVs used) of chromosome x^i, which is denoted as $FV^i = F(x^i)$.

Step 1.2 Set initial reference point $z = (z_1, z_2, \ldots, z_m)$ where $z_j = \min_{1 \le i \le N} f_j(x^i)$.

Step 1.3 Generate initial weight vectors $\{\lambda^1, \ldots, \lambda^i, \ldots, \lambda^N\}$, calculate their Euclidean distances, and select T closest weight vectors for each weight vector λ^i. The neighborhood set of λ^i is denoted as $B(i) = \{i_1, \ldots, i_T\}$, and its closest weight vectors is denoted as $\{\lambda^{i_1}, \ldots, \lambda^{i_T}\}$.

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

Step 2: Optimization

For $i = 1, \ldots, N$, do

Step 2.1 Conduct chromosome crossover. Choose two random numbers in $B(i) = \{i_1, \ldots, i_T\}$, and each number corresponds to one chromosome; thus, two chromosomes are selected. Then, order-based crossover is conducted to generate two new chromosomes (see Section 3.5).

Step 2.2 Conduct chromosome mutation. Choose one random number in $B(i) = \{i_1, \ldots, i_T\}$, and this number corresponds to one chromosome; thus, one chromosome is selected. Then, inverse mutation is conducted to generate a new chromosome (see Section 3.5).

Step 2.3 Update the objective values of the new population, and the chromosome with minimum objective values is selected as the elitist chromosome, which is denoted as y. Next, update the reference point z.

For each $j = 1, \ldots, m$, if $z_j > f_j(y)$, then $z_j = f_j(y)$.

Step 2.4 Update the neighborhood chromosomes.

For each $j \in B(i)$, if $g(y|\lambda^j, z) \leqslant g(x^j|\lambda^j, z)$, then set $x^j = y$ and $FV^j = F(y)$.

Step 3: Output optimization result

When the number of maximum iterations is reached, output the optimized UAV cruise routes and their objective values.

3.3. Feasible UAV cruise route chromosome generation

For satisfying the constraints of Eqs. (5)–(9), the start and end nodes of a link must be connected directly in one UAV cruise route. Moreover, integer-based target arrangement method[37] is adopted to represent the feasible UAV cruise route chromosome, which uses an integer number sequence to represent the sequence of unvisited links/nodes. Suppose that the number of unvisited links is M, an odd number sequence is used to represent the start nodes of unvisited links, and it is denoted as S (e.g., 5 and 7). Then, each number of this sequence is added with 1, and this new sequence is denoted as E to represent the end nodes (e.g., 6 and 8). Based on S and E, an intermediate matrix R is first introduced to store the initial start nodes of unvisited or new links, so as not to change the node structure of S. Then, R is used to store the optimized feasible UAV cruise route chromosome. An insertion comparison method is proposed to generate feasible chromosome, which is shown as follows.

$R = S$;
For $i = 1{:}1{:}M$
$S_1 = []; S_2 = []$;
In set R, insert $E(i)$ in the front of $S(i)$, then a new set S_1 is generated;
In set R, insert $E(i)$ behind $S(i)$, then another new set S_2 is generated;
Calculate the route lengths of S_1 and S_2; if the former is shorter than the latter, $R = S_1$; otherwise, $R = S_2$;
End
Output the feasible UAV cruise route chromosome R.

To better demonstrate the method, an unvisited link "5-6" and a new link "7-8" are presented as an example. Suppose that the start and end node sets are denoted as $S = \{5, 7\}$ and $E = \{6, 8\}$, respectively. First, "6" is inserted in the front of "5" and behind "5," respectively, and then we get two UAV routes, that is, 6-5-7 and 5-6-7. If the route length of 6-5-7 is shorter, the selected route is denoted as 6-5-7. Then, "8" is inserted in the front of "7" and behind "7," respectively, and we get two UAV routes, that is, 6-5-8-7 and 6-5-7-8. If the route length of 6-5-7-8 is shorter, the selected route is denoted as 6-5-7-8, and 6-5-7-8 is used to represent the feasible chromosome. This method ensures that the start and end nodes of each link are connected directly, and the length of the selected route is small, which is beneficial to generate feasible UAV cruise route chromosome.

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

3.4. UAV sub-route division

To obtain the objective values of a UAV route chromosome, a link insertion method is proposed to conduct UAV sub-route division, and its steps are presented as follows.

Step 1: At time t_1, some UAV routes have been finished and then delete them. Next, determine the unfinished UAV routes and the new links. Then, generate a UAV route chromosome according to Section 3.3.

Step 2: Select one link of the chromosome generated in Step 1 and insert it into an unfinished UAV route. Then, a new UAV sub-route is generated, and its length should not exceed UAV maximum flight distance. If the length constraint is satisfied, select another link of the above chromosome and insert it into the abovementioned unfinished UAV route. Otherwise, insert the link into another unfinished UAV route.

Step 3: If all the unfinished UAV routes are inserted and there are some links still remaining in the chromosome, these remaining links are used to generate new UAV sub-routes.

To better demonstrate the proposed method, two unfinished UAV routes (e.g., 0-10-9-3-4-15-16-0 and 0-2-1-12-11-14-13-0) and two new links (e.g., 6-5 and 7-8) are given as an example. Suppose that in the unfinished UAV routes, 10-9 is the visited link and 3-4 and 2-1 are the critical links. Therefore, the unvisited nodes are {15, 16, 12, 11, 14, 13, 6, 5, 7, 8}. In this condition, a random UAV route chromosome can be denoted as 11-12-5-6-8-7-13-14-15-16. Then, it begins to conduct UAV sub-route division. First, link 11-12 is inserted into 0-10-9-3-4, and a new UAV route 0-10-9-3-4-11-12-0 is generated. If the route length does not exceed UAV maximum flight distance, link 5-6 is inserted, and a new UAV route 0-10-9-3-4-11-12-5-6-0 is generated. If the route length does not exceed UAV maximum flight distance, the feasible UAV route is 0-10-9-3-4-11-12-0. Second, link 5-6 is inserted into 0-2-1, and a new UAV route 0-2-1-5-6-0 is generated. If the route length does not exceed UAV maximum flight distance, then link 8-7 is inserted, a new UAV route 0-2-1-5-6-8-7-0 is generated, if the route length exceeds UAV maximum flight distance, the feasible UAV route is 0-2-1-5-6-0. Third, nodes 8, 7, 13, 14, 15, and 16 are remaining. If these nodes are used to generate a new UAV route 0-8-7-13-14-15-16-0 and the route length does not exceed UAV maximum flight distance, then 0-8-7-13-14-15-16-0 is another feasible UAV route. Therefore, there are three feasible UAV routes, that is, 0-10-9-3-4-11-12-0, 0-2-1-5-6-0, and 0-8-7-13-14-15-16-0. Hence, the number of UAVs used is three, and the total cruise distance can be acquired.

3.5. Crossover and mutation

Crossover is used to keep the population diversity. Order-based crossover operator is adopted to generate new chromosomes. First, the mated section of two chromosomes is designated, and a link's start and end nodes must be connected directly in this mated section. Second, the mated section of one chromosome is put in the front of the other chromosome. Third, the repeated numbers in the rear are deleted, to ensure that one link is cruised only once. For example, there are two parent UAV cruise route chromosomes, 12345678 and 56872143. Suppose that the matched sections are 3456 and 8721, respectively. Then, 8721 is put in the front of the first chromosome, and 3456 is put in the front of the second chromosome, and two new chromosomes are generated, that is, 8721‖12345678 and 3456‖56872143. Next, the repeated numbers in the rear are deleted; therefore, two children chromosomes are represented as 87213456 and 34568721, respectively.

Mutation is used to improve the algorithm local search ability. Inverse mutation is adopted to generate new chromosome. First, the inverse section is designated, and the start and end positions of the inverse section must be an odd number and an even number, to ensure that one link is cruised only once. Second, the numbers in this section are inversed. For example, a chromosome is represented as 123456, and the start and end positions of inverse section are "3" and "6," respectively, then the mutation chromosome can be represented as 126543.

4. Case Study

4.1. Description of case study

A road network is used to simulate the UAV cruise route rerouting optimization, which is shown in Fig. 3. For verifying the effectiveness of the proposed approach, two scenarios are presented as

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

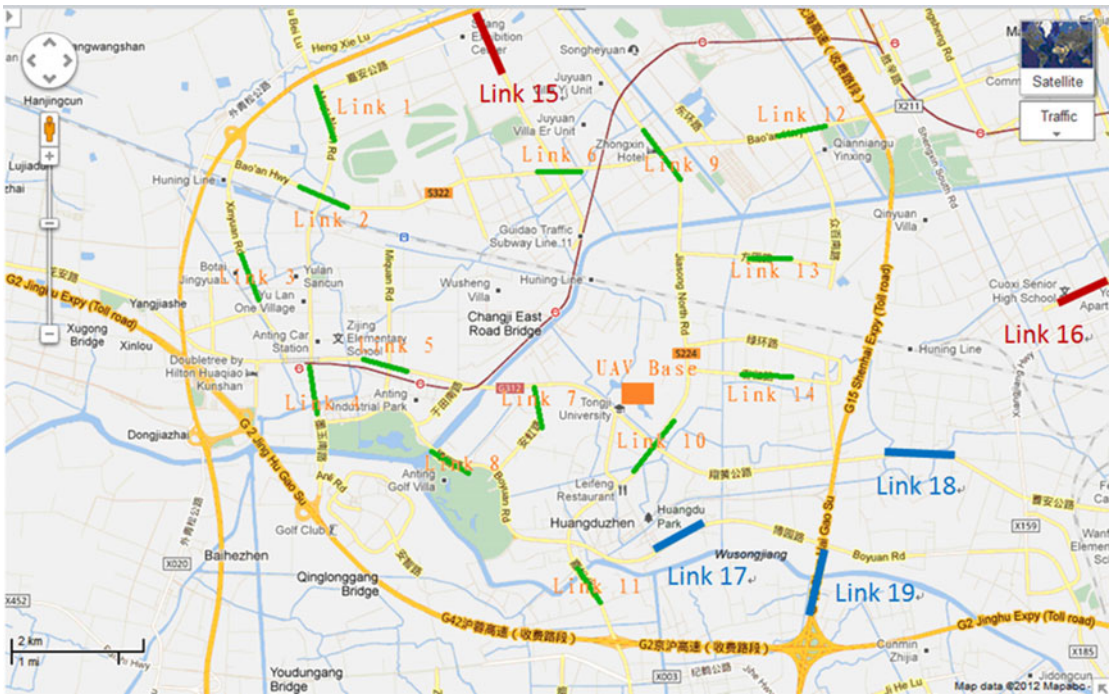

Fig. 3. Distribution of road segments.

follows: (1) there are 14 links, and UAVs need to monitor the traffic situation of these links based on the optimized cruise routes at time $t = 0$ h; (2) there are an exhibition center and a high school near link 15 and link 16, respectively, traffic congestion is likely to occur on these links; therefore, link 15 and 16 are selected. Moreover, there are a park, an intersection, and a bridge near links 17, 18, and 19, respectively, traffic congestion or accident is also likely to occur on these links; therefore, links 17, 18, and 19 are selected. In addition, links 15, 16, 17, 18, and 19 are supposed to appear at time $t = 0.4$ h, and the road network has 19 links in this scenario. At time $t = 0.4$ h, there are several links appearing simultaneously; thus, UAV cruise route scheme should be adjusted in real time in scenario 2. For the sake of clarity, the geographical positions of 19 links are extracted and their coordinates are listed in Table II, in which each link has two points, that is, the start point and end point, and each point has a point ID. In addition, UAV maximum flight distance is set at 20 km, and UAV flight speed is set at 30 km/h.

4.2. Optimization analysis

The optimization was implemented at the MATLAB platform, and the parameters of the proposed algorithm were set as follows: the population size was 100, the iteration number was 300, the crossover rate was 0.8, the mutation rate was 0.1, the number of inverse mutation was 1, and the number of neighborhood weight vectors was 5. The proposed algorithm was optimized for 15 times to avoid algorithm randomness. The optimized UAV cruise routes of scenario 1 are shown in Fig. 4.

Fig. 4 shows that there are two UAV cruise routes in scenario 1, the first route is 0-13-14-16-15-10-9-8-7-6-5-4-3-2-1-11-12-0, and the second route is 0-18-17-23-24-26-25-28-27-22-21-19-20-0. At time $t = 0.4$ h, one UAV is flying from node 1 to node 11, and the other UAV is flying from node 27 to node 22; therefore, link 11-12 and link 22-21 are critical links. Since link 11-12 is the critical link, and link 11-12 will be visited in the first route. In addition, the length of the route 0-13-14-16-15-10-9-8-7-6-5-4-3-2-1-11-12-0 is 18.79 km, and no more links can be inserted into this route; otherwise, UAV maximum flight distance (i.e., 20 km) will be exceeded. In this condition, it is thought that the first route has been finished.

At time $t = 0.4$ h, the visited nodes are {13, 14, 16, 15, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 11, 12, 18, 17, 23, 24, 26, 25, 28, 27}, the critical nodes are {21, 22}, and the remaining nodes are used to generate feasible UAV cruise route chromosome. Fifteen best solutions were selected from the solutions of initial population, and these initial solutions were compared with the optimized solutions, which are

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

Table II. Coordinates of different links and UAV base.

Link ID	*X* axis/km	*Y* axis/km	Point ID	Link ID	*X* axis/km	*Y* axis/km	Point ID
1	−4.823	4.173	1	11	−0.296	−3.655	21
	−4.608	3.733	2		−0.146	−4.095	22
2	−4.697	2.821	3	12	2.465	3.463	23
	−5.157	2.981	4		2.935	3.563	24
3	−5.681	2.387	5	13	2.009	1.564	25
	−5.551	1.927	6		2.509	1.564	26
4	−5.004	0.141	7	14	2.263	0	27
	−4.944	−0.349	8		2.763	0	28
5	−3.831	0	9	15	−2.625	4.473	29
	−3.331	−0.09	10		−2.314	4.201	30
6	−1.353	2.858	11	16	7.348	1.213	31
	−0.863	2.898	12		7.792	1.476	32
7	−1.629	−0.292	13	17	0.512	−3.421	33
	−1.669	−0.772	14		0.924	−2.914	34
8	−3.313	−0.665	15	18	4.312	−1.023	35
	−2.883	−0.775	16		4.813	−1.023	36
9	0.329	3.18	17	19	2.794	−3.855	37
	0.569	2.75	18		2.853	−3.321	38
10	0.391	−1.165	19	UAV base	0	0	0
	0.631	−0.745	20				

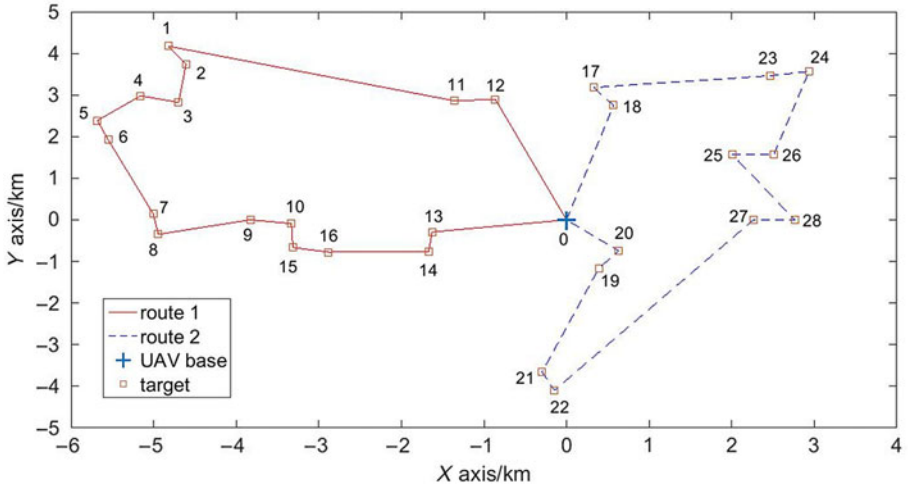

Fig. 4. Optimized UAV cruise routes of scenario 1.

listed in Table III. Based on the optimization result of scenario 1, the dynamic UAV cruise route optimization was conducted, and the optimized routes of scenario 2 are shown in Fig. 5.

It can be seen from Table III that the optimized solutions of the proposed algorithm are better than the initial optimal solutions. Compared with the optimal total cruise distances of initial solutions, optimized solutions of two scenarios decrease by −41.28% and −1.23%, respectively. In addition, compared with the average total cruise distances of initial solutions, optimized solutions of two scenarios decrease by −42.00% and −5.79%, respectively. Furthermore, compared with the maximum total cruise distances of initial solutions, optimized solutions of two scenarios decrease by −41.77%, and −7.71%, respectively. This demonstrates that the proposed algorithm is feasible and effective for UAV cruise route planning.

To demonstrate the real-time UAV route adjustment, Fig. 6 is presented to describe the UAV rerouting. In Fig. 6(a), UAV route 1 is finished at time $t = 2255$ s, and UAV route 2 is finished at time $t = 2340$ s. In Fig. 6(b), new links 15, 16, 17, 18, 19 appear at time $t = 0.4$ h (i.e., 1440 s), and the

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

Table III. Optimization result comparison of two scenarios.

Scenario	Parameter	Optimal total cruise distance/km	Optimal number of UAVs used	Average total cruise distance/km	Average number of UAVs used	Maximum total cruise distance/km	Maximum number of UAVs used
1	Initial solutions	66.03	4.00	71.50	4.40	77.71	5.00
	Optimized solutions	38.77	2.00	41.47	2.80	45.25	3.00
	Decrease degree	−41.28%	−50.00%	−42.00%	−36.36%	−41.77%	−40.00%
2	Initial solutions	58.65	4.00	61.67	4.00	63.91	4.00
	Optimized solutions	57.93	4.00	58.1	4.00	58.98	4.00
	Decrease degree	−1.23%	0.00%	−5.79%	0.00%	−7.71%	0.00%

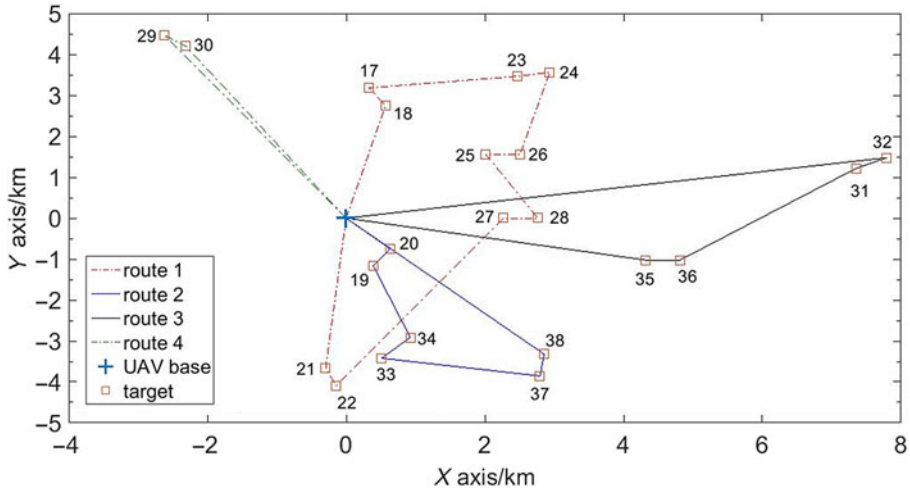

Fig. 5. Optimized routes of scenario 2.

second, third, and fourth UAV routes begin at time $t = 1440$ s. It should be noted that the start time of these routes does not consider the optimization calculation time and the transmission delay time to acquire the information of new links.

In order to verify the effectiveness of the proposed algorithm (DMEA), NSGA-II, and SPEA are selected, which are the most popular benchmark algorithms for optimization comparison.[27,28] Therefore, DMEA is compared with the most commonly used NSGA-II and SPEA. The proposed algorithm parameters are the same as mentioned earlier. The parameters of SPEA were set as follows: the internal population size was 80, the external population size was 20, the iteration number was 300, the crossover rate was 0.8, and the mutation rate was 0.1. The parameters of NSGA-II were set as follows: the population size was 100, the iteration number was 300, the crossover rate was 0.8, the mutation rate was 0.1, the tournament population size was 50, and the number of tournament UAV routes was 2. A computer with windows 8 operation system, Intel Core i7-4870HQ CPU 2.50GHz, and RAM 16.0GB was used, and the optimization comparison was implemented at the MATLAB R2014b platform for 15 times. The optimized solutions of different algorithms are compared and shown in Table IV.

It can be seen from Table IV that these three algorithms can find the optimal solutions in two scenarios, the optimal total cruise distances of two scenarios are 38.77 and 57.93 km, respectively, and the optimal number of UAVs used of two scenarios are 2 and 4, respectively. On the other hand, the proposed algorithm outperforms SPEA and NSGA-II in terms of the percentage of finding optimal solutions and calculation time. Compared with the percentage of finding optimal solutions using SPEA, the values of the proposed algorithm increase by 50.04% and 9.10%, respectively. Compared with the calculation time using SPEA, the values of the proposed algorithm decrease by 61.27% and 54.33%, respectively. Compared with the percentage of finding optimal solutions using NSGA-II,

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

Table IV. Optimized solution comparison of different algorithms.

Scenario	Algorithm	Optimal total cruise distance/km	Optimal number of UAVs used	Percentage of finding optimal solutions	Calculation time/s
1	SPEA	38.77	2.00	13.33%	40.87
	NSGA-II	38.77	2.00	13.33%	40.54
	Proposed algorithm	38.77	2.00	20.00%	15.83
	Improvement degree	0.00%;0.00%	0.00%;0.00%	+50.04%; +50.04%	−61.27%; −60.95%
2	SPEA	57.93	4.00	73.33%	53.54
	NSGA-II	57.93	4.00	80%	52.80
	Proposed algorithm	57.93	4.00	80%	24.45
	Improvement degree	0.00%; 0.00%	0.00%; 0.00%	+9.10%; 0.00%	−54.33%; −53.69%

Note: There are two numbers in the row of improvement degree, they are the improvement percentages compared with SPEA and NSGA-II, respectively.

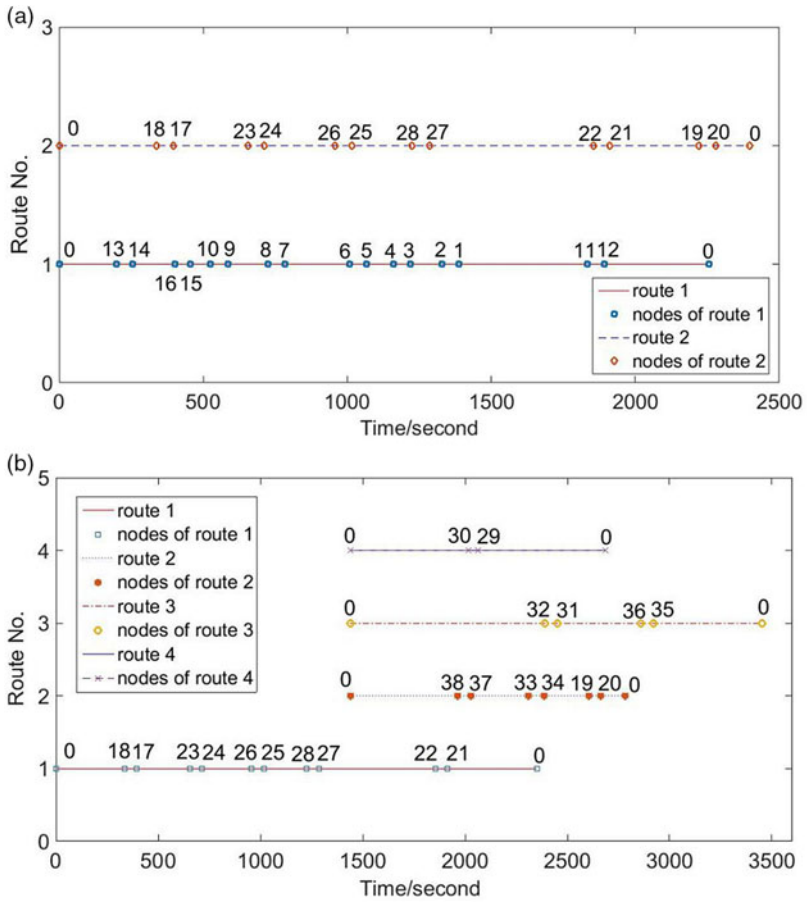

Fig. 6. The sequence diagram of UAV re-routing: (a) scenario 1; (b) scenario 2.

the values of the proposed algorithm increase by 50.04% and 0.00%, respectively. Compared with the calculation time using NSGA-II, the values of the proposed algorithm decrease by 60.95% and 53.69%, respectively. There are three factors accounting for the good performance of the proposed algorithm, the first factor is that neighborhood strategy is adopted, which is helpful to find optimal solutions; the second factor is that the proposed algorithm can alter weight vectors to approach the nonconcave Pareto Front; the third factor is that the proposed algorithm has lower computational complexity. The computational complexities of the proposed algorithm, NSGA-II and SPEA are $O(mNT)$, $O(mN^2)$, and $O(mN^2)$, respectively, in which m is the number of objective functions, N is the population size, and T is the number of neighborhood weight vectors. If the number of objective

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

Table V. Sensitivity analysis of T.

Parameter	Average total cruise distance/km	Average number of UAVs used	Average computation time/s
$T = 3$	58.21	4.00	23.19
$T = 5$	58.10	4.00	24.44
$T = 7$	58.28	4.00	24.49

Table VI. Sensitivity analysis of N and G.

Parameter	Total cruise distance/km			Number of UAVs used			Percentage of finding optimal solution (%)	Average computation time/s
	Max	Mean	Min	Max	Mean	Min		
$N = 50, G = 150$	58.98	58.28	57.93	4.00	4.00	4.00	66.67	5.98
$N = 50, G = 300$	58.98	58.21	57.93	4.00	4.00	4.00	73.33	11.77
$N = 100, G = 150$	58.98	58.10	57.93	4.00	4.00	4.00	73.33	12.00
$N = 100, G = 300$	58.98	58.10	57.93	4.00	4.00	4.00	80.00	24.44

functions and the population size are the same, the computational complexity ratio among the proposed algorithm and other two algorithms is $O(T)/O(N)$. Commonly, T is much less than N (e.g., T and N equal to 5 and 300, respectively, in this case study). Therefore, the proposed algorithm has faster optimization speed.

Generally, the average computation time of two scenarios decreases significantly, and this is helpful to acquire the optimal UAV cruise routes as soon as possible. The good algorithm performance helps to find the best solutions in real time. Moreover, the computation time is closely related to the amount of links and the computer performance. Larger amount of links will increase the computation time.

4.3. Sensitivity analysis

Sensitivity analysis was conducted to test three algorithm parameters, that is, the number of neighborhood weight vectors (T), the population size (N), and the iteration times (G). T was set at 3, 5, and 7, respectively, G was set at 200 and 300, respectively, and N was set at 50 and 100, respectively. Based on scenario 2, sensitivity analysis was implemented for 15 times to avoid algorithm randomness. Analysis result of T is shown in Table V, and analysis result of N and G is shown in Table VI. It can be seen from Table V that when T equals to 5, the average total cruise distance is minimum; additionally, the average computation time increases gradually with T increasing. Table VI shows that the computation time increases significantly and the solution quality becomes better when N increases from 50 to 100, or G increases from 150 to 300.

5. Conclusions and Recommendations

This study proposes a real-time UAV cruise route planning approach for road segment surveillance. The model considers the real-time road segment surveillance and multi-objective optimization, which is more suitable for real-life UAV traffic monitoring problem. A case study with two scenarios is conducted, and the proposed algorithm (DMEA) is compared with the commonly used NSGA-II and SPEA. The study results show that all of them can find the optimal solutions; moreover, DMEA outperforms NSGA-II and SPEA in terms of computation time and the percentage of finding optimal solutions. This indicates that the proposed model and algorithm are promising for real-time UAV cruise route planning in the context of road segment surveillance. Furthermore, sensitivity analysis of three algorithm parameters (i.e., the number of neighborhood weight vectors, the population size, and the iteration times) is conducted, and the analysis results indicate that the number of neighborhood weight vectors has certain impacts on optimized solutions, and increasing the population size and iteration times will improve the quality of optimization solutions. Meanwhile, greater population size and iteration times will increase computation time significantly.

https://doi.org/10.1017/S0263574720000867 Published online by Cambridge University Press

In real-life UAV traffic monitoring, there are some limitations. First, UAV flight height is always constrained due to the consideration of public safety. For example, without flight approval, UAVs less than 7 kg can only take flights below the altitude of 120 m in China.[2] Second, the video quality of UAV cameras depends on weather conditions (e.g., rainy, foggy, windy weather) and the technical performances of cameras. Third, most UAV maximum flight distance is limited; thus, it is difficult for a UAV to cover the whole road network. It is necessary to use multiple UAVs to work together for traffic monitoring; therefore, how to implement the cooperation surveillance of multiple UAVs is an emerging problem. Fourth, there are many high buildings and other infrastructures in the urban downtown area, there are potential collision risks for UAV flights. Flight collision avoidance and the cooperation of multiple UAVs are two concerned issues in the future.

As for the practical applications, the proposed approach can be used to collect the real-time traffic information of large-scale road segments, which is the foundation data of traffic congestion dissemination analysis. In addition, UAVs can carry on some environmental sensors to collect air pollution data of different road segments at different altitudes, these data can describe the air pollution development of road network, and the analysis result will provide important support for making traffic regulation policies. Furthermore, UAVs can be loaded with cameras and 3D radar systems to collect the image and point cloud data of traffic accident scene, then image processing technology and multi-stereo technology can be used to reconstruct the traffic accident scene, which will provide great conveniences for traffic accident investigation in terms of traffic interruption and accident analysis.

Acknowledgments

This research was supported by the Science and Technology Plan Project of Tianjin, China (17ZXRGGX00070, 19YFSLQY00010, 19JCQNJC03400, 20KPHDRC00030) and the Talent Development Fund of Shanghai Institute of Technology, China (10120K209043-A06).

References

1. A. C. Watts, V. G. Ambrosia and E. A. Hinkley, "Unmanned aircraft systems in remote sensing and scientific research: classification and considerations of use," *Remote Sens.* **4**(6), 1671–1692 (2012).
2. Civil Aviation Administration of China, *Air Traffic Control Regulation of Civil Unmanned Aerial Vehicle Systems*, Beijing, China (2016).
3. Y. Xu, G. Yu, X. Wu, Y. Wang and Y. Ma, "An enhanced viola-jones vehicle detection method from unmanned aerial vehicles imagery," *IEEE Trans. Intell. Transp. Syst.* **18**(7), 1–12, (2016).
4. L. Zhang, Z. Peng, D. J. Sun and X. Liu, "A UAV-Based Automatic Traffic Incident Detection System for Low Volume Roads," *Transportation Research Board of the National Academies* (National Research Council, Washington DC, 2013) pp. 542–558.
5. R. Dobson, T. Colling, C. Brooks, C. Roussi, M. Watkins and D. Dean, "Collecting decision support system data through remote sensing of unpaved roads," *Transp. Res. Record J. Transp. Res. Board* **2433**, 108–115 (2014).
6. J. Seo, L. Duque and J. Wacker, "Drone-enabled bridge inspection methodology and application," *Autom. Const.* **94**, 112–126 (2018).
7. B. Coifman, D. Beymer, P. Mclauchlan and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transp. Res. Part C Emerg. Tech.* **6**(4), 271–288 (1998).
8. A. Puri, A Survey of Unmanned Aerial Vehicles (UAV) for Traffic Surveillance (Department of Computer Science and Engineering, University of South Florida, 2005) pp. 1–29.
9. A. Romanoni, L. Mussone, D. Rizzi and M. Matteucci, "A comparison of two Monte Carlo algorithms for 3D vehicle trajectory reconstruction in roundabouts," *Pattern Recogn. Lett.* **51**(C), 79–85 (2015).
10. X. Liu, L. Zhao, Z. Peng, T. Gao and J. Geng, "Use of Unmanned Aerial Vehicle and Imaging System for Accident Scene Reconstruction," *Transportation Research Board of the National Academies (*National Research Council, Washington DC, 2017) pp. 432–450.
11. Q. Peng, Z. Peng and Y. Chang, "Unmanned Aerial Vehicle Cruise Route Optimization Model for Sparse Road Network," *Transportation Research Board of the National Academies (*National Research Council, Washington DC, 2011) pp. 432–445.
12. J. L. Ryan, T. G. Bailey, J. T. Moore and W. B. Carlton, "Reactive Tabu Search in Unmanned Aerial Reconnaissance Simulations," *Simulation Conference Proceedings*, Washington, DC, USA, vol. 871 (1998) pp. 873–879.
13. J. Zhang, L. Jia, S. Niu, F. Zhang, L. Tong and X. Zhou, "A space-time network-based modeling framework for dynamic unmanned aerial vehicle routing in traffic incident monitoring applications," *Sensors* **15**(6), 13874–13898 (2015).
14. X. Liu, Z. Peng, Y. Chang and L. Zhang, "Multi-objective evolutionary approach for UAV cruise route planning to collect traffic information," *J. Central South Univ.* **19**(12), 3614–3621 (2012).

15. Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerospace Sci. Tech.* **16**(1), 47–55 (2012).
16. Y. Chen, G. Luo, Y. Mei, J. Yu and X. Su, "UAV path planning using artificial potential field method updated by optimal control theory," *Int. J. Syst. Sci.* **47**(6), 1407–1420 (2016).
17. S. Ragi and E. K. P. Chong, "UAV path planning in a dynamic environment via partially observable Markov decision process," *IEEE Trans. Aerospace Electr. Syst.* **49**(4), 2397–2412 (2013).
18. J. Wang, Y. Zhang, L. Geng, J. Y. H. Fuh and S. H. Teo, "A heuristic mission planning algorithm for heterogeneous tasks with heterogeneous UAVs," *Unmanned Syst.* **3**(3), 1–15 (2015).
19. N. Ozalp, U. Ayan and E. Oztop, "Cooperative Multi-task Assignment for Heterogonous UAVs," *International Conference on Advanced Robotics,* vol. 64(7) (2015) pp. 599–604 .
20. M. D. Phung, C. H. Quach, T. H. Dinh and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Autom. Const.* **81**, 25–33 (2017).
21. H. R. Boveiri, "An incremental ant colony optimization based approach to task assignment to processors for multiprocessor scheduling," *Front. Inf. Tech. Electr. Eng.* **18**(4), 498–510 (2017).
22. P. Novoa-Hernandez, C. C. Corona and D. A. Pelta, "Self-adaptation in dynamic environments-a survey and open issues," *Int. J. Bio-Inspired Comput.* **8**(1), 1–13 (2015).
23. M. H. Kim, H. Baik and S. Lee, "Response threshold model based UAV search planning and task allocation," *J. Intell. Robot. Syst.* **75**(3), 625–640 (2014).
24. X. Liu, Z. Guan, Y. Song and D. Chen, "An optimization model of UAV route planning for road segment surveillance," *J. Central South Univ.* **21**(6), 2501–2510 (2014).
25. N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.* **2**(3), 221–248 (1994).
26. J. D. Knowles and D. W. Corne, "The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimization," *Proceedings of the 1999 Congress on Evolutionary Computation,* Washington DC (IEEE, 1999) pp. 98–105.
27. E. Zitzler and L. Thiele, "Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.* **4**(3), 257–271 (1999).
28. K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002).
29. Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007).
30. Y. Tan, Y. Jiao, H. Li and X. Wang, "MOEA/D plus uniform design: A new version of MOEA/D for optimization problems with many objectives," *Comput. Operat. Res.* **40**(6), 1648–1660 (2013).
31. P. C. Chang and S. H. Chen, "The development of a sub-population genetic algorithm II (SPGA II) for multi-objective combinatorial problems," *Appl. Soft Comput.* **9**(1), 173–181 (2009).
32. L. While, L. Bradstreet and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Trans. Evol. Comput.* **16**(1), 86–95 (2012).
33. D. Li, K. Li, J. Liang and A. Ouyang, "A hybrid particle swarm optimization algorithm for load balancing of MDS on heterogeneous computing systems," *Neurocomputing* **330**, 380–393 (2019).
34. O. Engin and A. Guclu, "A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems," *Appl. Soft Comput.* **72**, 166–176 (2018).
35. L. Ke, Q. Zhang and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony," *IEEE Trans. Cybern.* **43**(6), 1845–1859 (2013).
36. H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.* **13**(2), 284–302 (2009).
37. X. Liu, L. Gao, Z. Guan and Y. Song, "A multi-objective optimization model for planning unmanned aerial vehicle cruise route," *Int. J. Adv. Robot. Syst.* **13**, 1–8 (2016).