




Biobjective UAV routing for a mission to visit multiple mobile targets

Nail Karabay¹ · Murat Köksalan^{1,2} · Diclehan Tezcaner Öztürk³ 

Received: 2 August 2022 / Accepted: 23 March 2023 / Published online: 12 April 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

We address the route planning problem of an unmanned air vehicle (UAV) that operates in a two-dimensional hostile terrain monitored with radars. In this terrain, there are a number of targets that are planned to be visited to collect intelligence. We consider a setting where a UAV starts from a base, visits the targets, and returns to the base. Targets may move during the UAV's mission and their movement directions are unpredictable in advance. We consider two objectives: to minimize distance and to minimize radar detection threat. These objectives are conflicting in the regions monitored by radars. The constructed routes comprise the visiting order to the targets and the trajectories used between the visited pairs of targets. There are many efficient trajectories between the target pairs and many efficient visiting orders of the targets due to the two conflicting objectives. As a result, there are many efficient routes that are combinations of efficient trajectories and efficient orders of visits. Due to the dynamic nature of the problem, there is a need to select the route to follow in real time. To respond to these challenges, we develop an algorithm that finds a preferred route of a route planner (RP) quickly. We characterize the efficient trajectories between target pairs approximately and utilize the RP's preferences to choose the preferred efficient trajectories and to construct a preferred route. During the flight of the UAV, targets keep moving. We update the route every time the UAV reaches a new target. We also develop a heuristic approach in case the problem needs to be solved faster. We demonstrate our algorithms on 5, 9, and 15-target problems.

Keywords Unmanned air vehicles · Real-time routing · Multiobjective optimization · Continuous space

✉ Diclehan Tezcaner Öztürk
diclehanozturk@hacettepe.edu.tr

Extended author information available on the last page of the article

1 Introduction

Unmanned air vehicles (UAVs) are unpiloted aircraft used for both military and civilian purposes such as reconnaissance, surveillance against crime, agricultural applications, weather forecast, hurricane detection, transportation, and minimizing hazardous effects of natural disasters. UAVs are far less costly in initial and operational expenses (fuel and maintenance) and do not need on-board pilots, in contrast with conventional aircraft. They have high-resolution video-recording capabilities. Owing to these advantages, the use of UAVs has become widespread in recent years.

In a military context, route planning for UAVs comprises determining the path the UAV follows while visiting a number of targets in a hostile terrain. This terrain is typically monitored by radars to protect the targets from hostile intrusion. Radars pose a threat to UAVs as there is a possibility that they could be destroyed if detected by a hostile force. Criteria, such as the total distance of the route, flight duration, flight altitude, average distance from the radar sites, safety, radar detection threat, or risk posed by a UAV to third parties on the ground could be considered. Although many studies consider a single criterion, there are an increasing number of studies that consider multiple criteria (see Rojas Vilorio et al. 2021, for a review). There are recent studies that include a second criterion in addition to a travel distance-based criterion. In this study, we consider two conflicting objectives: minimizing the total distance of the route and minimizing the radar detection threat. The first objective is to minimize the distance traveled by the UAV, and the second objective is to minimize a detection measure for the route based on a cumulative measure of the probability that the radars will detect the UAV on its route. These are two relevant objectives in military applications of UAV route planning.

The studies that consider similar criteria (such as fuel consumption, path length, risk of threat, distance to radar sites), mainly consider a static environment where conditions stay the same throughout the mission of the UAV. Li et al. (2006) linearly aggregate path length and threat exposure into a single objective and search for the path of a UAV using particle swarm optimization. Zhenhua et al. (2008) consider similar objectives and find alternative routes for a UAV using an ant colony system without aggregating the objectives. Swartzentruber et al. (2009) use particle swarm optimization to construct paths for a UAV using a linear aggregation of fuel consumption, risk of threats, and reconnaissance criteria. They find alternative paths by varying the weights of the criteria in the linear aggregation. Pfeiffer et al. (2009) develop efficient paths for a UAV between an origin and a destination minimizing the flight duration and the probability of being detected in excess of a threshold number of times. Dolicanin et al. (2018) linearly aggregate fuel consumption and threat exposure and use a brain storm optimization algorithm to structure the routes of a UAV. Tezcaner and Köksalan (2011) develop an interactive approach to find the best route of a route planner (RP) who is assumed to have an implicit underlying linear preference function of travel distance and radar detection threat. They consider the routing of a

single UAV visiting multiple targets in a hostile terrain. Dasdemir et al. (2020) also consider the same problem and search for efficient routes close to the reference points specified by the RP to represent their preferences using an evolutionary algorithm. In all these approaches, the UAV follows the route selected at the beginning of the mission ignoring any changes that might occur in the environment during the mission.

A more realistic version of this problem is to take into account the dynamic nature of the environment. Some approaches consider pop-up or mobile threats by linearly aggregating several objectives into a single objective and employ heuristic approaches to find approximate solutions (see Zheng et al. 2005, Besada-Portas et al. 2010, Peng et al. 2013 for examples of routing multiple UAVs, and Yao and Zhao 2015, for an example of routing a single UAV). Studies that model the realistic case of mobile targets are rare. Qu and Xi (2010) determine the route of a fleet of UAVs for targets that move in known directions. They use a mixed integer linear program that minimizes the total route length to determine the coordinates of the points at which the UAVs meet their targets. Chen and Xu (2016) find the path of a UAV that visits several moving targets in a radar-monitored terrain where the visiting sequence is known in advance. They linearly aggregate the fuel consumption and radar threat objectives, restrict the path of the UAV to move through the edges of a graph they form based on the locations of the radars, and solve shortest path problems between consecutive targets on this graph.

Natalizio et al. (2020) introduce a different context where UAVs are used to film a sports event. They optimize a linear aggregation of the two criteria: the travel distance and timeliness of arrival to capture an action. They find the routes of a team of UAVs as the locations of the actions (to be filmed) keep changing.

Another dimension in multiobjective UAV route planning is to consider the existence of multiple trajectories when traveling between a pair of targets. In the classical multiobjective traveling salesperson problem (TSP), a single connection is assumed to exist between any two targets (target pairs). However, under multiple objectives, this is not a realistic setting since different trajectories could favor different objectives at different levels, leading to multiple efficient trajectories. In a continuous space where the UAV can move to any point in the terrain, this implies the existence of multiple efficient trajectories between target pairs. The problem, therefore, requires searching for efficient trajectories to use between target pairs as well as determining the visiting order to the targets. A few studies allow for multiple trajectories between target pairs using the objectives of minimizing flight distance and detection threat. Tezcaner and Köksalan (2011) introduced this problem for a discrete terrain where a UAV is restricted to move between imaginary grid points that are assumed to be uniformly located over the terrain. Tezcaner Öztürk and Köksalan (2016) and Köksalan and Tezcaner Öztürk (2017) used different approaches for the same problem, also considering a discrete terrain. Discretizing the terrain limits the possible moves of the UAV compared to the actual situation that allows the UAV to move to any point in the space. Tezcaner Öztürk and Köksalan (2023), Türeci (2017), and Dasdemir et al. (2020) worked on this more realistic version of the problem where the UAV is routed in a continuous space. Tezcaner Öztürk and Köksalan (2023) develop methods to generate the nondominated frontier of the routes for this

problem. For the same problem structure, Türeci (2017) develops interactive algorithms to find the most preferred solution of a RP having different preference functions of the two objectives. Dasdemiir et al. (2020) develop a reference point-based evolutionary algorithm that approximates the preferred regions of the nondominated frontier and apply the algorithm to the biobjective UAV route planning problem in continuous space.

In this paper, we consider a more general case than the problem setting of Tezcaner Öztürk and Köksalan (2023), Türeci (2017), and Dasdemiir et al. (2020). We form the route of a single UAV that has a mission to visit a set of targets by moving freely in continuous space considering flight distance and detection threat objectives. These targets are typically located on ground vehicles that move at a much slower speed relative to the UAV. As is commonly done in the literature, we assume that the UAV moves to the location of each target. In practice, the exact location the UAV needs to move could change slightly depending on its mission. In general, it needs to move close enough to accomplish its mission. Within the context of this study, such slight differences do not affect the routing decisions. Once the UAV reaches the vicinity of a target, it can make small adjustments to get to the desired location to achieve its mission. We consider multiple trajectories between target pairs and, different from the previous studies, a dynamic environment where the targets change their locations throughout the mission of the UAV. We assume that the movements of the targets are unknown to us in advance. Our approach starts with finding an initial route considering the observed locations of the targets when the UAV is at the base. The route is constructed taking the RP's preferences into account. When the UAV is en route to the first target, all targets keep moving in directions unknown in advance but can be observed by the UAV. The UAV locks on to the next target and moves toward it, adjusting its route as necessary. At the point the UAV completes its reconnaissance mission for that target, it observes the new locations of the remaining targets and updates its route based on the observed locations. Our algorithms keep constructing and updating preferred routes of the RP, taking the changing conditions into account each time the UAV reaches a target. To the best of our knowledge, this is the only study that considers a dynamic environment in a continuous space while allowing for multiple trajectories that could be used between target pairs.

UAVs are commonly used for military missions that have characteristics similar to the problem structure we consider. Pursuing mobile targets is an important mission of UAVs (Glade 2000), and our approach captures the fundamental aspects of such missions well. A specific application of UAVs that can be modelled with our problem structure is gathering intelligence from static or mobile targets using sensors (cameras, recording devices, etc.). Mufalli et al. (2012) explicitly address the problem of which sensors to place on a fleet of UAVs and they determine the routes of the UAVs to maximize the information collected from static targets. Our approach addresses the dynamic version of this problem under two objectives for a UAV that has a fixed set of sensors.

The paper unfolds as follows. We present definitions and state our problem in Sect. 2. We develop solution mechanisms and an algorithm in Sect. 3 and demonstrate it in Sect. 4. We conclude and discuss possible future work in Sect. 5.

2 Problem definition

We first introduce our notation and definitions.

Let p denote the number of objectives and assume without loss of generality that they are all of minimization type. Let \mathbf{x} be a solution (that is, the decision variable vector indicating the route to follow in our case) and X be the feasible set. $z(\mathbf{x}) = (z_1(\mathbf{x}), z_2(\mathbf{x}), \dots, z_p(\mathbf{x})) \in Z$ denotes the objective function vector corresponding to solution $\mathbf{x} \in X$, where $z_k(\mathbf{x})$ is the performance of \mathbf{x} in objective k , and Z is the image of X in the objective function space.

We take the following definitions directly from Tezcaner Öztürk and Köksalan (2016).

Definition 2.1 A solution $\mathbf{x} \in X$ is efficient if there does not exist $\mathbf{x}' \in X$ such that $z_k(\mathbf{x}') \leq z_k(\mathbf{x})$ for $k = 1, 2, \dots, p$ and $z_k(\mathbf{x}') < z_k(\mathbf{x})$ for at least one objective. If there is such an \mathbf{x}' , \mathbf{x} is said to be inefficient. All efficient solutions constitute the efficient frontier (set).

Definition 2.2 If a solution $\mathbf{x} \in X$ is efficient, then $z(\mathbf{x})$ is said to be nondominated, and if \mathbf{x} is inefficient, and then $z(\mathbf{x})$ is said to be dominated. All nondominated points constitute the nondominated frontier (set).

Definition 2.3 An extreme efficient solution \mathbf{x} is an efficient solution that has the minimum possible value in at least one of the objectives.

We next explain the problem setting, the structure we create to represent the trajectories between target pairs, and the route we construct.

2.1 The problem setting

We consider the same problem setting as in Tezcaner Öztürk and Köksalan (2023), Türeci (2017), and Dasedemir et al. (2020), where the route of a UAV is constructed in continuous space. In these studies, the UAV is assumed to start from a base, visit each target once, and return to the base. They minimize distance and radar detection threat objectives. Tezcaner Öztürk and Köksalan (2023) generate the nondominated frontier of this problem. Türeci (2017) and Dasedemir et al. (2020) find preferred solutions using exact and heuristic approaches, respectively. They all assume static targets. In contrast, we consider mobile targets that are ground vehicles that move at slow speeds relative to the speed of the UAV. Our approach can work for different relative speeds for the ground vehicles and the UAVs.

The exact movements of the targets are unknown to us, but we assume that the UAV observes the targets throughout its mission. The initial route is constructed when the UAV is at the base for the observed locations of the targets. The UAV then locks on to the next target and flies towards it adjusting its route according to the movements of that target. Once the UAV reaches the point it can collect

information, and we update the remaining legs of the route based on the new locations of the unvisited targets. We further discuss the details of constructing the route later.

During its travel toward the targets, the UAV can move to any point in a two-dimensional terrain that is monitored by radars. The UAV tries to minimize both the distance traveled and the radar detection threat. The total distance traveled is the sum of the lengths of all the paths the UAV traverses. We measure the possibility of detection using the radar detection threat measure developed in Gudaitis (1994). Based on this measure, the effectiveness of radar in detecting a UAV depends on how close it is. If the UAV is within a certain distance, it is detected with certainty. This detection probability decreases proportional to its distance from the radar and reaches 0 beyond a certain distance. The region of effectiveness of the radar is shown in Fig. 1a with two concentric circular areas assuming that the radar is located at the center. The detection probability is 1.0 in the inner circular region (the core region) depicted with a grey shading. The outer region is the remaining white area in the circular radar region, where the detection probability starts at 1.0 at the circumference of the core region and gradually decreases toward the circumference of the outer region, and reaches a value of 0.0 at the outer circumference. To create an overall measure of radar detection threat objective, we accumulate all detection probabilities on the route of the UAV, as the overall risk to the UAV depends on the instantaneous probability of detection weighted with the duration of being exposed to that probability.

The details of the computation of this objective function are given in Appendix.

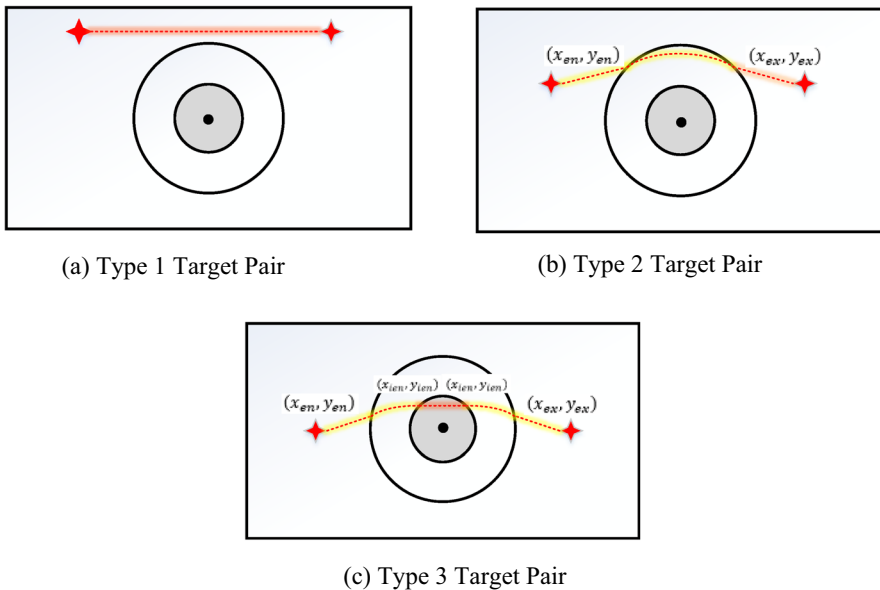


Fig. 1 Types of target pairs (adapted from Tezcaner Öztürk and Köksalan, 2023)

2.2 Trajectories

As the UAV can move to any point in the continuous space, there are multiple efficient trajectories between each target pair the UAV can use, each performing better than any other trajectory in one objective and performing worse in the other objective. These trajectories differ by the points at which they enter the effective radar region, by the paths they follow in the radar region, and by the points they exit the effective radar region. It would be necessary to determine all these components (the entrance and exit points to and from the radar region, and the path to follow in the radar region) in order to structure all efficient trajectories exactly. In a dynamic setting with moving targets, the routes need to be updated in real time, regularly updating the trajectories based on the new locations of the targets. It is not practical to find and update all these trajectories precisely in such a dynamic setting. We utilize a simplified approach that approximates the nondominated frontiers of the efficient trajectories to be used in the updated route. For each target pair, we find several efficient trajectories precisely and approximate the nondominated frontiers of all efficient trajectories by fitting functions that pass through the objective function values of these few trajectories using the method developed by Tezcaner Öztürk and Köksalan (2023). We then select the preferred points of a RP (corresponding to the objective function values of the two objectives) on each of these frontiers and use the corresponding trajectories to form routes. We next explain Tezcaner Öztürk and Köksalan's (2023) representation of the nondominated frontiers between target pairs.

Tezcaner Öztürk and Köksalan (2023) enumerate many trajectories between a target pair for different entrance and exit points to and from the radar region, and the path followed within the radar region. They then eliminate the inefficient trajectories and form the nondominated frontier of the enumerated efficient solutions. They observe that the nondominated frontiers of different target pairs exhibit similar structures. They classify these under three categories based on the positioning of the target pairs relative to the radars. If the shortest path between two targets does not pass through a radar region, there is a single efficient trajectory that simultaneously minimizes both the distance and the radar detection threat objectives when moving between those targets (target pair Type 1, see Fig. 1a). The target pairs for which the shortest path passes through only the outer radar region are categorized as Type 2 (Fig. 1b). Finally, the target pairs for which the shortest path passes through both the outer and the core radar regions are categorized as Type 3 (Fig. 1c).

Tezcaner Öztürk and Köksalan (2023) structure the nondominated frontiers of Types 2 and 3 target pairs as demonstrated in Fig. 2a and b, respectively. The trajectories connecting Type 3 target pairs have two segments: a straight-line segment that is made up of the objective function values of the efficient trajectories passing through both the core and the outer radar regions, and a curved segment that is made up of the objective function values of the trajectories passing through only the outer radar region. An example efficient trajectory between a Type 3 target pair is shown in Fig. 1c. The points (x_{en}, y_{en}) and (x_{ex}, y_{ex}) represent the entrance and exit points to and from the outer radar region, respectively. The entrance and exit points to and

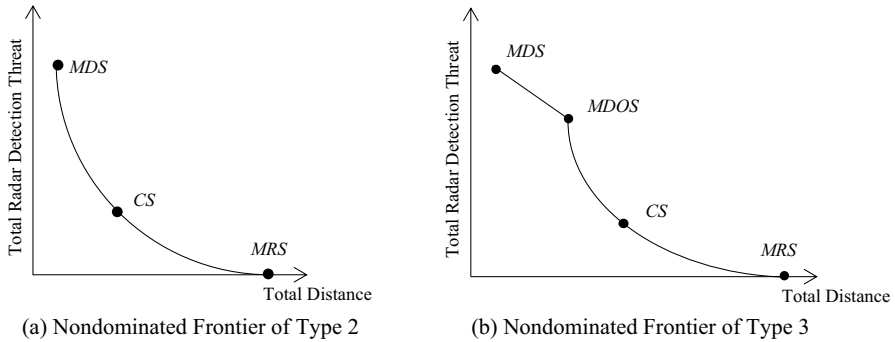


Fig. 2 Nondominated frontiers of type 2 and 3 target pairs (adapted from Dasdemir et al., 2020)

from the core radar region are the points (x_{ien}, y_{ien}) and (x_{iex}, y_{iex}) , respectively. The nondominated frontier of Type 2 target pairs is a curved frontier and is similar in structure to the curved segment of Type 3 target pairs. An example trajectory is shown in Fig. 1b. Type 1 target pairs, on the other hand, have a single efficient trajectory connecting them by a straight path.

Based on these structures, Tezcaner Öztürk and Köksalan (2023) characterize the nondominated frontiers of trajectories of Types 2 and 3 by fitting functions that represent their shapes. In this approach, depending on whether the trajectories are of Type 2 or 3, three or four actual efficient trajectories are found between each target pair, respectively. Then the objective values of these trajectories corresponding to each target pair are used to fit a function to represent the nondominated frontier of the trajectories between that target pair. We next explain Algorithm *NDF* that fits functions to represent the nondominated frontiers for Type 2 and Type 3 target pairs.

Algorithm *NDF*

Although the algorithms for Type 2 and Type 3 target pairs have many aspects in common, we write two separate algorithms. We briefly discuss how to find trajectories here and elaborate on its details in a later section. To find the trajectory type, we check the trajectory that corresponds to the minimum distance solution (MDS) connecting target pair (i, j) . Let the distance and radar detection threat values of this trajectory be z_{MDS}^1 and z_{MDS}^2 , respectively. If this trajectory passes through the core radar region, the target pair is of Type 3. If it only passes through the outer radar region, the target pair is of Type 2.

Algorithm *NDF* for Type 2 Target Pairs

Step 1. Find the shortest path trajectory between target pair (i, j) that does not pass through any radar region. Let the objective values of this trajectory (minimum radar solution, MRS) be z_{MRS}^1 and z_{MRS}^2 , respectively.

Step 2. Find a central solution with distance value that is the midpoint of z_{MRS}^1 and z_{MDS}^1 . Let the distance and radar detection threat values of the central solution be z_{CS}^1 and z_{CS}^2 , respectively.

Step 3. Fit an L_q function passing through CS, MRS, and MDS using Eq. (1).

$$\left(1 - \frac{z_{CS}^1 - z_{MDS}^1}{z_{MRS}^1 - z_{MDS}^1}\right)^q + \left(1 - \frac{z_{CS}^2 - z_{MRS}^2}{z_{MDS}^2 - z_{MRS}^2}\right)^q = 1 \tag{1}$$

Algorithm NDF for Type 3 Target Pairs

Step 1. Find the shortest path trajectory between target pair (i, j) that avoids the core radar region. Let the objective values of this trajectory (minimum distance outer solution, MDOS) be z_{MDOS}^1 and z_{MDOS}^2 , respectively.

Step 2. Find the shortest path trajectory between target pair (i, j) that does not pass through any radar region. Let the objective values of this trajectory (minimum radar solution, MRS) be z_{MRS}^1 and z_{MRS}^2 , respectively.

Step 3. Find a central solution with distance value that is the midpoint of z_{MRS}^1 and z_{MDOS}^1 . Let the distance and radar detection threat values of the central solution be z_{CS}^1 and z_{CS}^2 , respectively.

Step 4. Fit an L_q function passing through CS, MRS, and MDOS using Eq. (2).

$$\left(1 - \frac{z_{CS}^1 - z_{MDOS}^1}{z_{MRS}^1 - z_{MDOS}^1}\right)^q + \left(1 - \frac{z_{CS}^2 - z_{MRS}^2}{z_{MDOS}^2 - z_{MRS}^2}\right)^q = 1 \tag{2}$$

Step 5. Estimate the straight-line piece of the nondominated frontier using the straight-line equation that passes through the objective function values of the two solutions, MDS and MDOS.

We start the algorithm with finding the extreme efficient trajectory with the minimum distance value and maximum radar detection threat value, MDS, and classify the target pair as Type 2 or 3. If the target pair is Type 2, we additionally find two points, MRS and CS, and characterize the curved nondominated frontier. A similar approach is employed for Type 3 target pairs. We find three points, MDOS, MRS, and CS, on the curved segment. We then fit an L_q function passing through the end points and a central point on these curved segments. These functions are developed in Köksalan (1999), demonstrated on several problems in Köksalan and Lokman (2009), and used in different contexts in Köksalan and Soylu (2010) and Köksalan and Tezcaner Öztürk (2017). The unknown in Eqs. (1) and (2) is the value of q , and it can be found solving a nonlinear mathematical model, with q as its only decision variable.

The straight-line segment of Type 3 moves is found with the equation of the line passing through two points, MDS and MDOS. We next explain how we structure efficient trajectories and, more specifically, how we find efficient trajectories in Algorithm NDF.

2.3 Structuring the efficient trajectories

All efficient trajectories are structured by their entrance points to and exit points from the radar region, and the circular paths they follow in the radar region. The structures of MDS, MRS, and MDOS are common to any position of targets with respect to the radar as established in Tezcaner Öztürk and Köksalan (2023). They can be found by solving a small set of linear equations simultaneously in negligible computational time.

As demonstrated in Fig. 2, there is a continuous nondominated frontier of the trajectories between Type 2 and Type 3 target pairs and hence a continuum of efficient trajectories as each distinct point on the nondominated frontier corresponds to a distinct trajectory. To structure any of the remaining efficient trajectories between targets (i, j) we employ the search-based algorithm developed in Tezcaner Öztürk and Köksalan (2023). The algorithm finds the efficient trajectory for any value of the first objective function, d_{ij} , $z_{\text{MDS}}^1 \leq d_{ij} \leq z_{\text{MRS}}^1$.

To obtain a central solution, CS, in Algorithm *NDF*, we set d_{ij} to z_{CS}^1 , the midpoint of z_{MRS}^1 and z_{MDOS}^1 , in the search-based algorithm.

2.4 Progress of the route

When the UAV is at the base, its route should be formed by solving a generalized multiobjective TSP (MOTSP), where the target pairs are connected by multiple efficient trajectories.

To decide on which of the efficient trajectories to follow, we need to reflect the preferences of the RP. If we know or estimate the preference function of the RP, we can solve the generalized MOTSP right before the UAV starts its flight and have the UAV move toward the first target of this route. However, as the targets keep changing their locations, the initial route may not stay preferred or may not even be nondominated. We may improve by restructuring the route periodically based on new information. One extreme may be to restructure the routes continuously, accounting for all small changes in the targets' locations. The other extreme is to let the UAV follow the initial route without any modifications. In between these two extremes are the alternatives of restructuring the route at certain time intervals. Since the UAV moves toward the next target at a much faster speed than that of the targets, it is neither practical nor useful to recalculate the routes in short time intervals.

One could choose different reasonable time intervals to reevaluate whether the current route remains a preferred route. We reevaluate every time the UAV reaches its next target. An earlier reevaluation would be beneficial only if it would require rerouting by changing the next target to be visited. This is unlikely since the UAV has been approaching to this target at a much faster speed than the speed of this target (as well as all other targets). Once the UAV reaches the next target, all other targets could have changed their locations relative to each other and relative to the

current location of the UAV. Therefore, the remaining legs of the UAV may change at this point in time and it is an opportune time to reevaluate.

The initial problem (with all targets to be visited) is a generalized MOTSP and yields a tour. Each time we complete the next leg of the trip we only need to find a route through the unvisited targets. Hence, the problem becomes a multiobjective shortest Hamiltonian path problem (SHPP) after visiting the first leg. Therefore, we solve a new multiobjective SHPP (MOSHPP) after visiting each target in real time. We refer to the UAV’s visiting order to the targets (starting from and ending at the base) as the tour in the rest of the manuscript. The visiting order to targets and the trajectories used between target pairs together constitute the route of the UAV.

Tezcaner Öztürk and Köksalan (2016) define the generalized biobjective traveling salesperson problem (GBOTSP). We provide their mathematical model below and then build upon their formulation to develop the mathematical model of the biobjective SHPP. In this model all nodes are connected with multiple efficient edges, and hence we refer to it as the generalized biobjective shortest Hamiltonian path problem (GBOSHPP). Let set M include the unvisited targets and the target the UAV is currently at, and T be the set of all targets. In TSP, $M = T$ since we aim to visit all targets when the UAV is at the base (Target 1). For both formulations, we assume that the number of efficient trajectories between targets (i, j) is finite and equal to H_{ij} . We use the binary variable x_{ij}^h to denote whether the efficient trajectory, h , connecting targets (i, j) is used or not and c_{ijh}^k denotes its coefficient in objective k .

(GBOTSP)

$$\text{Min}z_1(x) = \sum_{i \in M} \sum_{j \in M} \sum_{h=1}^{H_{ij}} c_{ijh}^1 x_{ij}^h \tag{3}$$

$$\text{Min}z_2(x) = \sum_{i \in M} \sum_{j \in M} \sum_{h=1}^{H_{ij}} c_{ijh}^2 x_{ij}^h \tag{4}$$

Subject to:

$$\sum_{j \in M} \sum_{h=1}^{H_{ij}} x_{ij}^h = 1 \quad \forall i \in M \tag{5}$$

$$\sum_{i \in M} \sum_{h=1}^{H_{ij}} x_{ij}^h = 1 \quad \forall j \in M \tag{6}$$

$$u_i - u_j + \sum_{h=1}^{H_{ij}} (|M| - 1)x_{ij}^h + \sum_{h=1}^{H_{ji}} (|M| - 3)x_{ji}^h \geq |M| - 2 \quad \forall i, j \in M - \{1\}, i \neq j \tag{7}$$

$$1 \leq u_i \leq |M| - 1 \quad \forall i \in M - \{1\} \quad (8)$$

$$x_{ij}^h \in \{0, 1\} \quad \forall i, j \in M, h = 1, \dots, H_{ij} \quad (9)$$

We minimize the two sum-type objectives in (3) and (4), respectively. Departure from and arrival to each target are satisfied with (5) and (6), respectively. Constraints (7) and (8) are the subtour elimination constraints (Desrochers and Laporte 1991) that are the strengthened version of subtour elimination constraints of Miller et al. (1960).

We solve the GBOSHPP once we arrive at a target, say ℓ , and search for the path starting from Target ℓ , reaching Target 1 (base), visiting all unvisited targets in set M .

(GBOSHPP)

(3), (4)

Subject to:

(7)–(9)

$$\sum_{j \in M - \{\ell\}} \sum_{h=1}^{H_{ij}} x_{ij}^h = 1 \quad \forall i \in M - \{1\} \quad (10)$$

$$\sum_{i \in M - \{1\}} \sum_{h=1}^{H_{ij}} x_{ij}^h = 1 \quad \forall j \in M - \{\ell\} \quad (11)$$

Constraints (10) and (11) are the modified versions of (5) and (6), respectively, such that only the targets that are not visited yet should be arrived at and departed from.

3 The solution approach

We develop a general real-time algorithm that constructs and restructures the route of the UAV in a dynamic environment as discussed above. The first phase of the algorithm constructs a route for the UAV when it is at the base by solving a GBOTSP for the initial locations of all targets. The UAV then moves toward the next target. At their meeting point, we update the route by solving a GBOSHPP that starts at the current location of the UAV, visits all unvisited targets (based on their current locations), and terminates at the base. The UAV then moves in the direction of the next target on the updated route. The process repeats until the UAV visits all targets and reaches the base. We refer to the first execution of the algorithm, where an initial tour is found, as *Phase 1*. We refer to the phase where we successively construct shortest Hamiltonian paths as *Phase 2*.

3.1 The real-time algorithm

We next present the steps of our real-time algorithm and explain how we execute each step in each phase.

Let the base be Target 1. Recall that set M includes the unvisited targets as well as the target the UAV is currently at, and T is the set of all targets. Let $Q = \{(i, j) : i, j \in M\}$, be the set of all target pairs.

Algorithm R-T

Step R.0. Find the objective function values of the efficient trajectories that can be used between target pairs $(i, j) \in Q$.

Step R.1. Find the most preferred route of the RP, and the target to visit next, say Target ℓ .

Step R.2. Find the new location of and the trajectory to follow to Target ℓ .

3.1.1 Using algorithm R-T in phase 1

We first execute Steps R.0-R.2 of Algorithm R-T when the UAV is at the base.

Step R.0.: finding efficient trajectories between target pairs $(i, j) \in Q$

Since the UAV moves in a continuous space, the efficient trajectories between target pairs are not finite. We employ Algorithm *NDF* to characterize the nondominated frontiers corresponding to these efficient trajectories.

Since the nondominated frontiers corresponding to the trajectories between target pairs are continuous, the nondominated frontier made up of tours that use these trajectories is also continuous. Furthermore, the environment is dynamic and keeps changing rapidly and the RP needs to make routing decisions quickly. In order to make decisions quickly in real time, we approximate the preferences of the RP with a weighted linear function of the objectives as given in (12), where w and $1 - w$ represent the relative importances of the first and the second objectives, respectively, to the RP. The most preferred solution of the RP is the solution that minimizes $U(x)$.

$$U(x) = wz_1(x) + (1 - w)z_2(x) \text{ where } 0 < w < 1 \quad (12)$$

We interact with the RP before the execution of the real-time algorithm to estimate the value of w^* that structures the RP's preference function. For this purpose, we use Türeci's (2017) interactive algorithm for underlying linear preference functions. This algorithm finds the most preferred route of a RP for the biobjective UAV routing problem in continuous space, with mechanisms addressing the structure of its continuous nondominated frontier. Briefly, in this algorithm, she finds the range of values, $[w_L, w_R]$, that w can take based on past preferences of the RP. This range can be narrowed down by asking for pairwise comparisons between solutions. We keep asking for pairwise comparisons from the RP and set the mid-point, w^* , as the estimated weight once the range is sufficiently narrow.

Once we estimate w^* and characterize the nondominated frontier, we next identify the objective function values of the trajectory between each target pair (i, j) that optimizes (13). Here, d_{ij} and r_{ij} are the first and second objective function values of the identified most preferred trajectory between targets (i, j) , respectively.

$$U = w^*d_{ij} + (1 - w^*)r_{ij} \tag{13}$$

We next briefly explain the procedure developed in Türeci (2017) (and used in Dasdemir et al. 2020, in a similar context) that finds the objective function values of the most preferred trajectory between a target pair for an underlying linear preference function with known w .

We first check the type of each target pair $(i, j) \in Q$. We then find the point (d_{ij}, r_{ij}) on the corresponding nondominated frontier that optimizes (13) using Procedure *BT*.

Procedure BT

- If target pair (i, j) is of Type 1, there is a single efficient trajectory with objective values (d_{ij}, r_{ij}) connecting them.
- If target pair (i, j) is of Type 2, we find the point (d_{ij}, r_{ij}) minimizing (13) on L_q function (14).

$$\left(1 - \frac{d_{ij} - z_{MDP}^1}{z_{MRP}^1 - z_{MDP}^1}\right)^q + \left(1 - \frac{r_{ij} - z_{MRP}^2}{z_{MDP}^2 - z_{MRP}^2}\right)^q = 1 \tag{14}$$

- If target pair (i, j) is of Type 3, we find the point (d_{ij}, r_{ij}) minimizing (13) on L_q function (15) and on the straight line parts separately. We then find the best trajectory as the point that gives a smaller value for (13).

$$\left(1 - \frac{d_{ij} - z_{MDOP}^1}{z_{MRP}^1 - z_{MDOP}^1}\right)^q + \left(1 - \frac{r_{ij} - z_{MRP}^2}{z_{MDOP}^2 - z_{MRP}^2}\right)^q = 1 \tag{15}$$

More details on this procedure can be found in Türeci (2017).

Step R.1.: finding the most preferred route and the next target to be visited

After finding the objective values of the efficient trajectory to be used between each target pair, (d_{ij}, r_{ij}) for all $(i, j) \in Q$, we find the most preferred route of the RP. For this, we solve the single-objective version of GBOTSP (3)-(9) where $H_{ij} = 1$ and $(c_{ij1}^1, c_{ij1}^2) = (d_{ij}, r_{ij})$ for all $(i, j) \in Q$. As the objective function, we use (12), combining (3) and (4) with w^* . The resulting tour, $1 - O_1 - O_2 - \dots - O_{|T|-1} - 1$ is a symmetrical tour starting and terminating at the base, where O_i indicates the target to be visited in the i^{th} order. We have the option of going to targets O_1 or $O_{|T|-1}$ from the base. Of these, we choose the target that has the better first-connection value. That is, we choose the target to visit comparing the preference function values of the trajectories connecting 1 with O_1 and 1 with $O_{|T|-1}$. The trajectory with the smaller preference value is selected, and the chosen target is denoted as Target ℓ .

Step R.2.: find the new location of and the trajectory to follow to target ℓ

In Step R.2, we observe the direction of movement of Target ℓ and predict the point at which the UAV meets Target ℓ given its travel speed. We then execute Algorithm *NDF* to form the nondominated frontier of the trajectories between Target 1

and the new location of Target ℓ , and utilize Procedure *BT* to find the two objective function values of the most preferred trajectory on the characterized nondominated frontier. Until this point, we have worked in the objective function space. As a final step, knowing the approximate objective function values, $(d_{1\ell}, r_{1\ell})$, of the most preferred trajectory, we employ the search-based algorithm of Tezcaner Öztürk and Köksalan (2023) to find where the UAV specifically enters the radar region, how it moves in the radar region, and where it leaves the radar region. This trajectory corresponds to the efficient trajectory that has the distance value $d_{1\ell}$ and the true radar detection threat value $r'_{1\ell}$. In our experiments, $r_{1\ell}$ turned out to be very close to $r'_{1\ell}$. In the rest of the paper, we will refer to this procedure as *structuring the trajectory of the UAV*.

In practice the target may deviate from its original direction of movement. In this case, the UAV can make small adjustments to follow the target. Given the UAV operates at a much faster speed than the targets, we expect such adjustments to result in negligible changes in the overall route.

3.1.2 Using algorithm *R-T* in phase 2

When the UAV visits Target ℓ , we assume that all targets have changed their locations. We thus start executing Algorithm *R-T* for the new locations of the targets. The main difference in using Algorithm *R-T* in Phase 2 is in forming the route in Step R.1. We approximate fewer nondominated frontiers considering only unvisited targets and solve a GBOSHPP instead of a GBOTSP to form the route. We explain the steps of Phase 2 briefly, pointing out its similarities with Phase 1.

Step R.0.: finding the efficient trajectories between target pairs $(i, j) \in Q$

We characterize the nondominated frontiers between target pairs that have not been visited yet using Algorithm *NDF* and find the objective function values of the most preferred trajectory between each target pair using Procedure *BT*, as in Phase 1.

Step R.1.: finding the most preferred route and the next target to be visited

We solve the single-objective version of the GBOSHPP combining (3) and (4) using w^* as in (12). The solution gives a path that starts from Target ℓ , ends at Target 1, and visits all unvisited targets in M . We then update $M \leftarrow M - \{\ell\}$, and let the next target to be visited be denoted as ℓ .

Step R.2.: find the new location of and the trajectory to follow to target ℓ

As in Phase 1, we observe the movement of Target ℓ , and find the location where Target ℓ and the UAV meets. We then use Algorithm *NDF* and Procedure *BT* to find the objective function values of RP's best trajectory to ℓ . Using the first objective function value of this trajectory, we find its structure (the trajectory's entry point to and exit point from the radar region, and the circular path it follows in the radar region) as in Phase 1. When the UAV reaches ℓ following this trajectory, we run Algorithm *R-T* to determine the new locations of the targets in set M .

We terminate the algorithm when there is only one target left, say Target k , to visit after the UAV visits Target ℓ , before returning to the base (Target 1). We thus

structure the remaining path as $\ell - k - 1$, and find the trajectories to follow between Targets ℓ and k , and k and 1.

We next give the detailed steps of using Algorithm *R-T* in Phase 2.

R-T Algorithm – Phase 2

Step P2.0. Approximate the nondominated frontiers between all target pairs (i, j) , $i, j \in M$, using Algorithm *NDF* for the updated map. Find the objective function values of the most preferred trajectories between target pairs (i, j) , $i, j \in M$, using Procedure *BT*.

Step P2.1. Find the most preferred route solving GBOSHPP using preference function (12). Update $M \leftarrow M - \{\ell\}$. Set ℓ as the next target in the route.

Step P2.2. Find the new location of Target ℓ . and structure the trajectory to follow to Target ℓ .

- If $|M| = 2$, where $M = \{1, k\}$, structure the path to follow as $\ell - k - 1$. Structure the trajectories to follow between Targets ℓ and k , and k and 1, and terminate the algorithm.
- Otherwise, go to Step P2.0 when the UAV reaches Target ℓ .

Note that we utilize the representations of the nondominated frontiers of the trajectories throughout our algorithm. For each target pair, we find several efficient trajectories and use these trajectories to characterize their nondominated frontiers. We then work on these characterized frontiers in the objective function space to find the most preferred route. After determining the next target in the last step, we structure the specific trajectory to that target. Postponing the structuring of the actual trajectories of the UAV to the last phase reduces the computational burden substantially.

Although the algorithm finds a solution quickly, there may still be benefits to further reduce the computation time since the routes need to be updated in real time. We next develop a heuristic approach that attempts to further reduce the solution time without sacrificing much from solution quality.

3.2 The k -closest heuristic

In Algorithm *R-T*, a considerable part of the computational effort is spent in step R.0. In this step, we find the objective function values of the most preferred trajectories between all target pairs. However, some target pairs are very distant and are not likely to be visited consecutively. Based on this observation, we may be able to skip some of the calculations in Step R.0 for distant target pairs without compromising solution quality. In order to implement this idea, we find the most preferred trajectory of each target considering its k closest targets only and use rough approximate trajectories for targets farther away. We next discuss the necessary modifications to Step R.0 of Algorithm *R-T*.

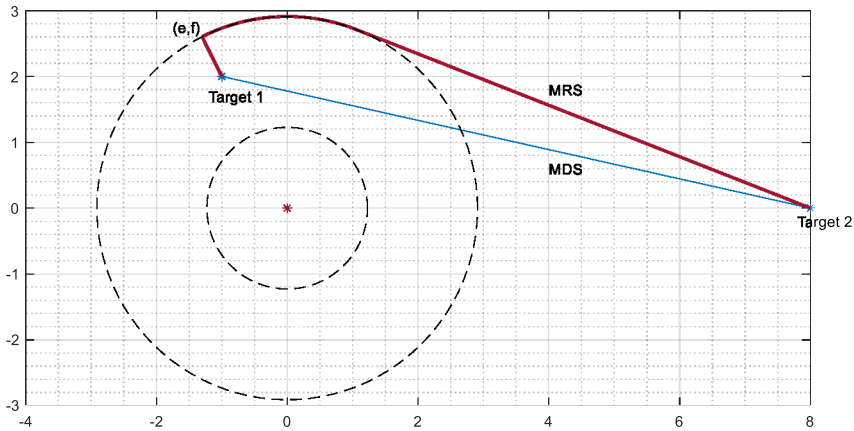


Fig. 3 Extreme trajectories when a target enters the effective radar region

Let $Q = \{(i, j) : i, j \in M\}$ be the set of all target pairs to be evaluated.

Modifications to step R.0 in the k-closest heuristic

We break step R.0 into four substeps as follows:

Step R.0.1. For each target pair $(i, j) \in Q$, find MDS and MRS having objective function values $(z_{ij,MDS}^1, z_{ij,MDS}^2)$ and $(z_{ij,MRS}^1, z_{ij,MRS}^2)$, respectively.

Step R.0.2. For each target pair $(i, j) \in Q$, compute the preference function value of the preferred one of MDS and MRS, $U_{ij} = \min_{t \in \{MDS, MRS\}} \{w_d^* z_{ij,t}^1 + w_r^* z_{ij,t}^2\}$, $t_{min} = \operatorname{argmin}_{t \in \{MDS, MRS\}} \{w_d^* z_{ij,t}^1 + w_r^* z_{ij,t}^2\}$, and $(d_{ij}, r_{ij}) = (z_{ij,t_{min}}^1, z_{ij,t_{min}}^2)$.

Step R.0.3. For each $i \in M$, let j be the order statistics such that $U_{i(1)} \leq U_{i(2)} \leq \dots \leq U_{i(|M|-1)}$. Let $Q_i = \{(j) : j = 1, \dots, k\}$.

Step R.0.4. For target pairs $(i, j), i \in M, j \in Q_i$, construct their nondominated frontiers using Algorithm *NDF*, and update the objective function values of their most preferred trajectories (d_{ij}, r_{ij}) using Procedure *BT*.

In Step R.0.1, we find the two extreme efficient solutions for all target pairs. We then assign the objective function values of the efficient solution that has a smaller preference function value as an approximation of the objective values, (d_{ij}, r_{ij}) , of the best trajectory between Targets i and j , in Step R.0.2. For each Target i , its k neighbors with the smallest U_{ij} values are identified and the nondominated frontiers between Target i and each of those k targets are characterized in Step R.0.3. The objective function values of the most preferred trajectories are selected in Step R.0.4. For targets that are not among the k closest targets, their most preferred trajectories are kept at their initially approximated values in Step R.0.2.

After finding the objective function values of the most preferred trajectories between target pairs, we continue with Steps R.1 and R.2 of Algorithm *R-T* as before.

Setting k to $|T| - 1$ corresponds to the original algorithm (i.e., finding the objective function values of the most preferred trajectories between all target pairs). As k gets smaller, there is a possibility that the solution quality deteriorates.

3.3 The 0-closest heuristic

This is the special case of the k -closest heuristic where $k = 0$. In this case, we do not construct the nondominated frontiers between target pairs. Instead, we use the extreme efficient trajectory of each target pair that has the smaller preference function value as the most preferred trajectory between that target pair.

Strictly speaking, Algorithm *NDF* structures the trajectories for targets that are in the regions radars are ineffective. If targets end up in the effective regions of the radars small adjustments may be necessary to determine the trajectories. Under such circumstances, we can employ the 0-closest heuristic using MDS and an approximation of the other extreme solution, MRS. We demonstrate an example in Fig. 3 where Target 1 falls in the effective region of a radar. Here, the light-colored (blue) trajectory that connects the two targets with the shortest path is the MDS. The dark-colored (red) trajectory is an approximation for MRS. In this trajectory, we first exit the radar region using the shortest path possible (using the straight-line path between Target 1 and point (e, f)). We then reach Target 2 by traversing the trajectory with the shortest distance that does not enter the radar region. If the UAV visits these two targets consecutively, the trajectory (MDS or MRS) that gives a better preference function value can be used. We can handle other configurations in a similar manner. We are currently working on structuring further efficient trajectories when the trajectories are directly under the regions covered by the radars.

4 Results

Although there are some differences, UAVs are restricted in the ranges they could fly and this affects the number of targets a UAV could typically visit. In order to consider a range of realistic cases, we demonstrate our algorithms on three test problems with 5, 9, and 15 targets. Similar number of targets have been used in many applications (see Venkatachalam 2018, for routing one UAV to 10 to 15 targets on average, Dasedemir et al. 2020, for routing to 15 targets at most, and Santin et al. 2021, for routing one UAV to 6 to 20 targets on average). In our implementation, we use an average speed of 220 km/h for the UAV, and a constant speed of 36 km/h for the ground vehicles carrying the targets (see Jayaweera and Hanoun 2022), which is approximately one-sixth of the speed of the UAV. We let the ground vehicles to move in random directions. For the radar detection threat measure, we set $C = 2270$, $LB_{S/N} = 15$, and $UB_{S/N} = 30$.

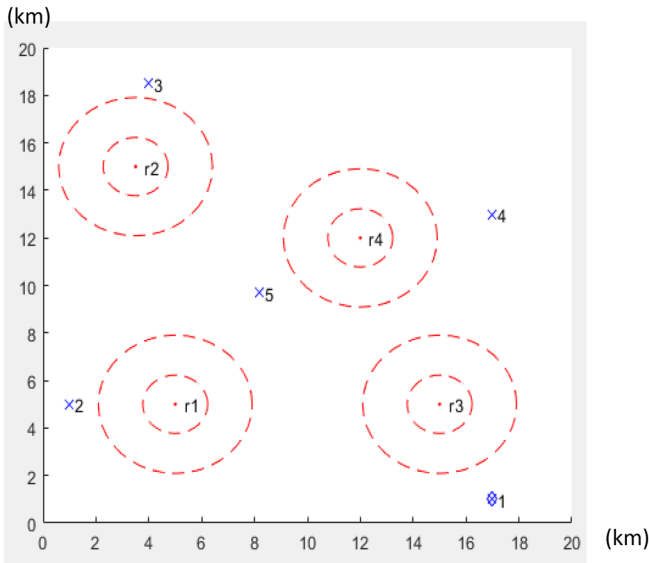


Fig. 4. 5-Target UAV route planning problem

While the UAV is traveling toward the next target, we assume that the target moves at a constant speed without changing its direction. Using the direction and speed information, we predict the point at which the UAV will reach the target. This is not a strong assumption as it encompasses a short time period during which the UAV moves from one target to the next. If the target changes its direction or speed, the UAV can make small adjustments to reach the target. Since the UAV moves at a much faster speed than the ground vehicle, we expect such small adjustments will not affect the selected route or the trajectory.

4.1 Five-target UAV route planning problem

In this case, we consider 5 targets (a base plus 4 targets) and 4 radar regions in a square terrain of 400 km^2 . The x mark shows the initial location of each target $i=1, \dots, 5$, and concentric circular regions represent the effective ranges of the radars in Fig. 4. The first target is the base where the UAV starts and terminates its route.

We run the algorithm in MATLAB R2017a and use optimization package CPLEX 12.9 to solve the mathematical models on Intel(R) Core(TM) i7-8550U CPU PC @ 1.80 GHz (8CPUs) with 16,384 MB RAM.

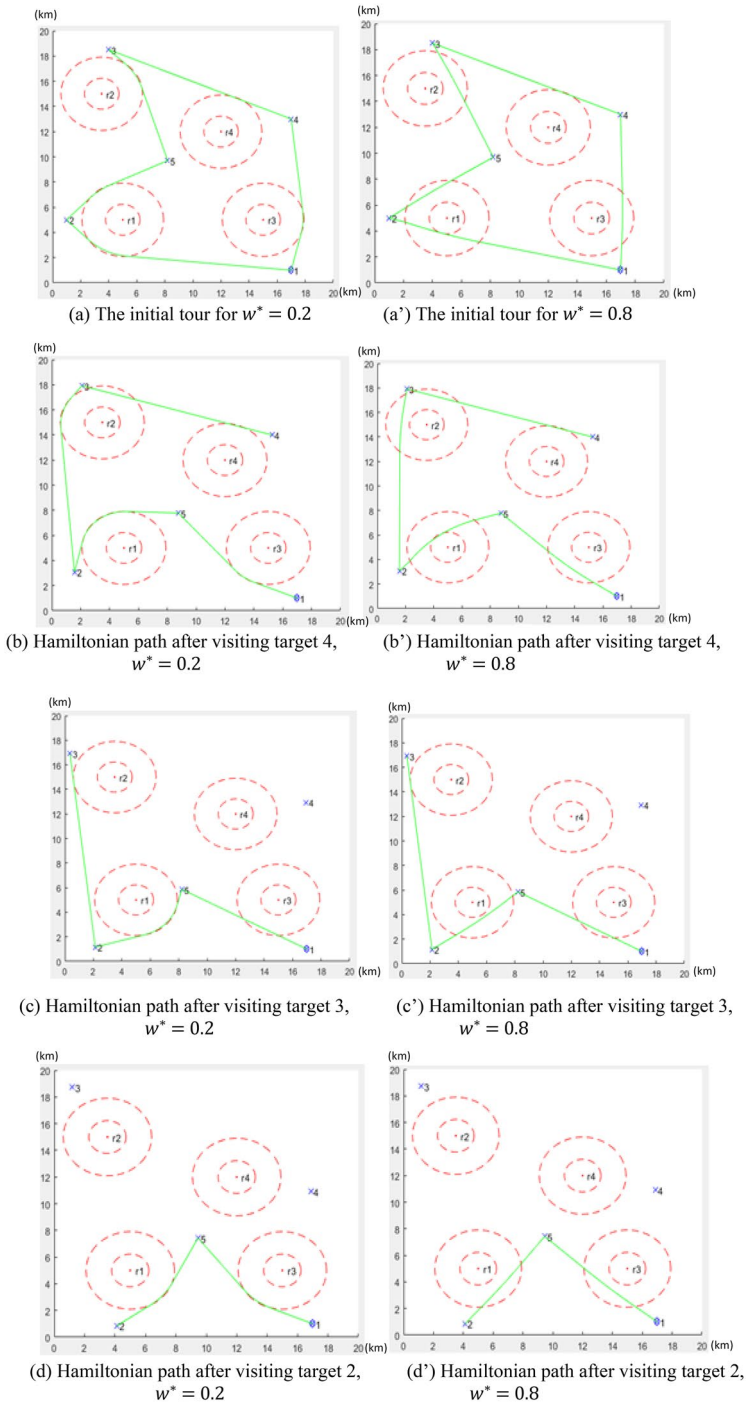


Fig. 5 Illustration of the algorithm, $k = 4$, $w^* = 0.2$ (the left column) and $w^* = 0.8$ (the right column)

Table 1 Results of 5-target problem

w	Initial route			Algorithm $R-T$ ($k=0,2,3,4$)		
	Tour	D	RDT	Tour	D	RDT
0.2	1-4-3-5-2-1	67.1532	0.2338	1-4-3-2-5-1	64.6500	0.1675
0.4	1-4-3-5-2-1	66.0302	2.7783	1-4-3-2-5-1	64.0135	1.6887
0.6	1-4-3-5-2-1	65.0673	6.8486	1-4-3-2-5-1	63.3481	4.8138
0.8	1-4-3-5-2-1	64.6674	10.1338	1-4-3-2-5-1	62.9861	8.0119

4.1.1 Illustration of the algorithm for $k = 4$, $w^* = 0.2$, and $w^* = 0.8$

We illustrate the results of the real-time algorithm by setting $k = 4$ for two different weights, $w^* = 0.2$ and $w^* = 0.8$, in Fig. 5. The UAV is at the base (Target 1) at time 0. w^* is the relative importance the RP assigns to the distance-traveled (D) objective, and $1 - w^*$ to the radar-detection-threat (RDT) objective. The RP gives a higher relative importance to the D-objective when $w^* = 0.8$, and to the RDT-objective when $w^* = 0.2$.

Both cases start with the same visiting order to the targets in Fig. 5a and a', but the trajectories followed between consecutive target pairs are different. In both cases, the initial tour, that is 1-4-3-5-2-1, is changed to 1-4-3-2-5-1 after the UAV visits Target 4. After visiting Target 3, the change in the locations of the unvisited targets does not result in a change in the visiting order. The decision of visiting Target 2 after Target 3 leaves no other path alternatives than the path 2-5-1 to reach Target 1. In general, for $w^* = 0.2$, the UAV avoids radar regions at the expense of longer D, whereas for $w^* = 0.8$ shorter trajectories are used taking the risk of larger RDT values, as would be expected.

4.1.2 Results of example problems

We next give the results of the algorithm for $k = 0, 2, 3$, and 4 for w values: 0.2, 0.4, 0.6, and 0.8. The results are summarized in Table 1. For all w values, we obtain the same results when $k = 0, 2, 3$, and 4, therefore we report their results in a single column. This is an important observation implying that the extreme solutions MDS and MRS are good indicators of the relative performances of the efficient frontiers of trajectories between a target and its different neighbors. We will further elaborate on this later.

In columns 2 to 4 of Table 1, we report the results for the case the UAV follows the initial route obtained in Phase 1, without any updating due to moving targets. We refer this case as IR (initial route). In columns 5 to 7, we give the results of Algorithm $R-T$. If the two algorithms result in the same visiting order to the targets, the trajectories to follow between consecutive target pairs will be found and structured using the same algorithms (Procedure BT followed with the search-based algorithm) and the resulting routes will be identical. Only if the visiting order changes, the resulting route changes. We indeed obtain results in accordance with the latter,

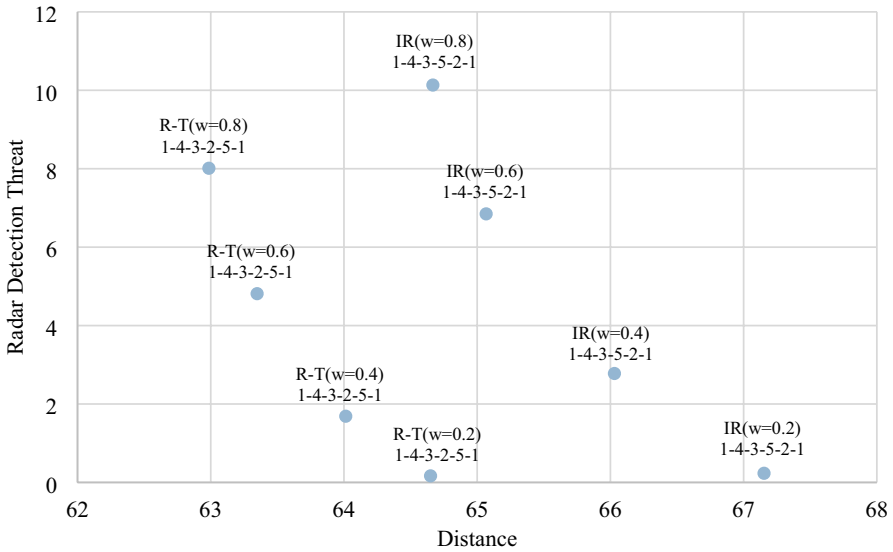


Fig. 6 Results of IR and algorithm *R-T* for the 5-target problem

for all w values. Although the visiting order is the same for the first two targets (1-4-3) in all tours, at the third iteration, the real-time algorithm changes the initial route. We plot the two objective values of the solutions obtained by both Algorithm *R-T* and IR, together with the resulting tour for each w value in Fig. 6. It can be seen that, for all w values, the routes of Algorithm *R-T* strictly dominate the routes obtained by IR (has better values in both objectives). As we increase the relative importance of D (higher w values), in general, the total D value decreases and the total RDT value increases.

We report the average solution times of four w values for $k = 0, 2, 3,$ and $4,$ after visiting each target in Table 2. The solution times generally decrease after each visit (as expected since the remaining problem is smaller in size). Also as expected, the solution times increase as the value of k increases, implying that the heuristic substantially reduces the computational burden.

4.2 Nine-target UAV route planning problem

We use the setting of Türeci (2017) with a base, eight targets, and four radar regions, slightly modifying the locations of the targets to demonstrate our approach better. We solve the problem for $k = 0, 2, 3,$ and $8.$ When $k = 8,$ we characterize at most 36 nondominated frontiers in the first iteration of the algorithm. This value decreases to 27 and 18, for $k = 3$ and $k = 2,$ respectively. Similar to the 5-target case, the results for $k = 0, 2, 3,$ and 8 are the same. For only $w = 0.8,$ the IR and Algorithm *R-T* find identical routes, with the same visiting order to the targets, and the same trajectories

Table 2 CPU times of algorithm *R-T* (seconds)—5-target problem

<i>k</i>	Iteration				Total
	1	2	3	4	
0	2.7740	2.3910	2.7880	2.5940	10.5470
2	13.2642	13.0458	6.9066	2.7128	35.9294
3	19.6258	19.2718	6.8294	2.6461	48.3730
4	28.0280	27.3350	7.7980	2.6100	65.7710

used between target pairs. For $w = 0.2, 0.4,$ and $0.6,$ updating the route progressively improves the objective values of the resulting tour and the results with Algorithm *R-T* strictly dominate those with IR (see Fig. 7). The average solution times of four w values for $k = 0, 2, 3,$ and 8 can be seen in Table 3. Again, the heuristic improves the computational burden substantially as k value gets smaller.

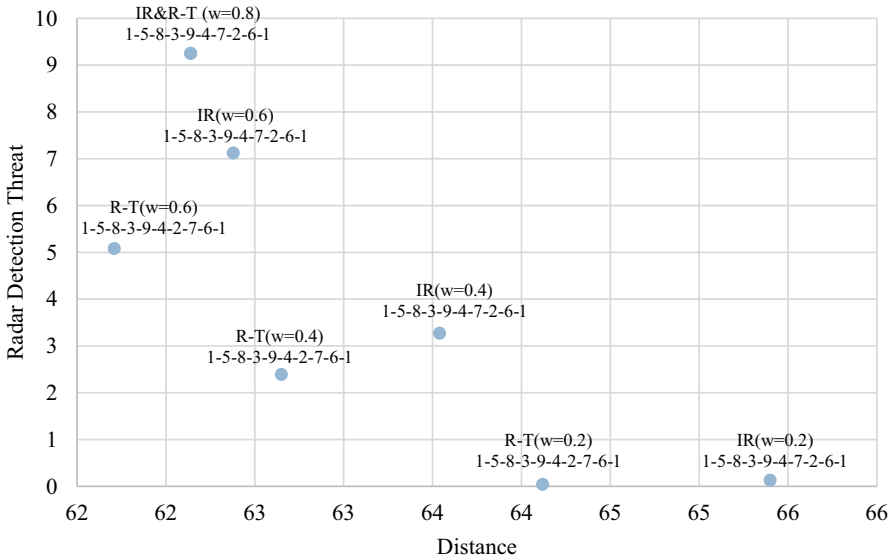


Fig. 7 Results of IR and algorithm *R-T* for the 9-target problem

Table 3 CPU times of algorithm *R-T* (seconds)—9-target problem

<i>k</i>	Iteration								Total
	1	2	3	4	5	6	7	8	
0	1.1474	4.3572	2.9009	0.5953	0.5935	1.0255	4.8634	3.3068	18.7899
2	13.8026	14.7835	9.1178	5.1229	10.5434	5.3185	8.7873	3.3140	70.7900
3	23.8795	32.8858	23.2464	15.0806	19.3717	9.5767	11.7988	4.3369	140.1764
8	60.6453	57.6695	33.4395	24.7435	17.3754	9.5500	7.3270	3.1666	213.9168

4.3 15-Target UAV route planning problem

Next problem we solve has one base, 14 targets, and 4 radar regions. We solve the problem for $k = 0, 2, 3,$ and $14,$ where we characterize at most $0, 30, 45,$ and 105 nondominated frontiers, respectively. All k values again resulted in the same tours. For all w values, the solutions we obtain with Algorithm $R-T$ have better objective values (strictly dominate) than those with IR (see Fig. 8). The solution times for $k = 0, 2, 3,$ and 14 given in Table 4 again show that the heuristic reduces the solution times substantially.

For all T -target problems, we obtain the same results when k is set to $0, 2, 3,$ or $T - 1.$ Although it might be possible to create a problem setting where this changes, we did not observe such a case in our further experiments, changing the locations of targets and radars. This is an important observation that the nondominated frontiers of target pairs have been substantially different from each other in terms of the magnitudes of the objectives in all our experiments so that even their extreme points are representative enough to develop the path to follow. Based on this observation, it may be possible to use the fastest version of the algorithm, especially when there are sizeable differences in the extreme points of the objective values of trajectories of different target pairs.

5 Conclusions

In this paper, we consider the real-time routing problem of a UAV in a two-dimensional dynamic environment where the locations of the targets change in time. The UAV, starting from a base, visits a set of targets and returns to the base. We consider two objectives to be minimized: distance traveled and radar detection threat. The

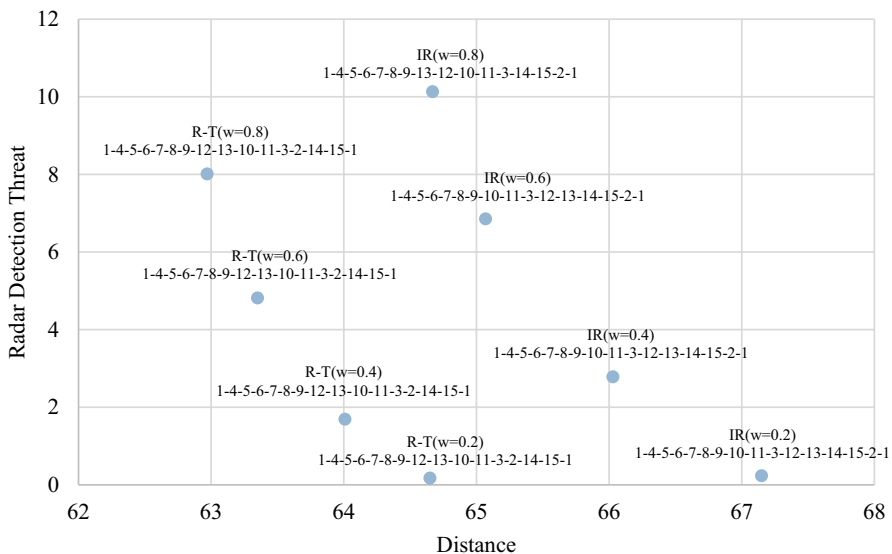


Fig. 8 Results of IR and algorithm $R-T$ for the 15-target problem

Table 4 CPU times of algorithm $R-T$ (seconds)—15-target problem

k	Iteration														Total
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
0	7.5060	1.3846	1.2349	0.9994	0.8557	0.8280	0.7513	0.8657	0.7436	0.7156	0.6961	2.8312	0.6208	0.5328	20.5657
2	12.5497	7.4940	1.2142	3.4528	3.0527	4.9016	0.9642	1.0632	0.9043	2.5654	6.7707	8.8722	2.7333	0.5511	57.0892
3	30.7941	21.4670	8.0478	7.8619	7.6289	15.1129	6.0482	5.5087	6.2992	14.3516	18.3508	17.6126	3.8512	0.6918	163.6266
14	146.5959	129.3704	112.5418	89.9264	64.8744	56.1650	49.1579	43.1888	29.3928	24.3432	19.0597	14.3084	3.3948	0.5582	782.8777

UAV moves in continuous space to visit multiple targets that change their locations to destinations unknown to us. To the best of our knowledge, this is the only study that considers such a dynamic environment in a continuous space to find and keep updating the best route of a RP under the two objectives.

We develop a real-time algorithm to find the most preferred route for a RP having an underlying linear preference function. The algorithm starts with an initial route and updates the route every time it reaches a new target, if necessary. The characterizations of the nondominated frontiers of the trajectories allow us to work in the objective function space throughout the algorithm, without the need to find the exact trajectories that require determining their entrance and exit points to and from the radar region, and the path they use within the radar region, as explained in Sect. 2.3. Only after determining the target to be visited next, Target ℓ , we find the exact trajectory between the current location of the UAV and Target ℓ . Although this keeps the computational burden low, we also develop a k -closest heuristic to further reduce the computational burden, keeping in mind that the algorithm needs to be solved in real time. In the heuristic approach, we make a detailed analysis to find the objective function values of the most preferred trajectory between relatively “close” targets and approximate the most preferred trajectories between the remaining pairs of targets with trajectories that are easier to compute. We demonstrate our algorithm on three test problems with 5, 9, and 15 targets. The algorithm finds identical tours for $k = 0, 2, 3$, and $|T| - 1$, for all problem sizes. Updating the route at each visit to the targets, rather than using the initial route, improves the objective function values in general. The results also show that we reduce the computation times considerably using the k -closest heuristic without sacrificing from the quality of solutions.

Although we allow the targets to change their locations keeping the radars static, our algorithm is applicable to dynamic radar locations as well. To implement the algorithm for changing radar locations, we need to characterize the non-dominated frontiers between target pairs for the new locations of the targets or radars at each execution of Phase 2.

Although UAV route planning has been studied well in recent years, the literature on real-time UAV route planning with multiple objectives is rather scarce. There are many possible extensions of this study. Our approach can be extended considering a three-dimensional continuous space. In this case, the altitude of the UAV and the ground structure should also be considered. Additionally, the computations of the two objectives, total distance and total radar detection threat, must be revised. Currently, we consider three different placement configurations of targets relative to the radars. If the targets reside in the effective regions, we may employ the procedure suggested in Sect. 3, and find a route using 0-closest heuristic. We are working on further enriching the efficient trajectories for targets lying directly in the effective radar regions.

If all movements of all targets during the mission of the UAV were known to the RP in advance (which is a strong assumption), finding efficient routes would still not be straightforward. In this case, the distances between all targets would change depending on the time interval at which the efficient trajectory between

them is traversed. The decisions of this problem would then include the visiting order to the targets, the trajectories to use between targets, as well as the distance between targets. This is a complex research question due to the combinatorial nature of all possible movements and could be a topic for future research. Another possible extension could be routing multiple UAVs to multiple targets in a dynamic environment. This would require making additional decisions of assigning UAVs to subsets of the targets to be visited. Considering additional dynamic components such as pop-up targets and radars is another interesting future study.

Appendix

Radar detection threat measure

We use the radar detection threat measure as in Tezcaner Öztürk and Köksalan (2023). Equation A.1 computes the detection threat of a trajectory between two targets located at Cartesian coordinates (x_a, y_a) and (x_b, y_b) . We integrate the detection probabilities along the trajectory of the UAV in order to come up with the detection threat measure.

$$RDT = \int_{(x_a, y_a)}^{(x_b, y_b)} pd_{(x,y)} ds \tag{A.1}$$

The detection probability is a function of the signal-to-noise ratio, which depends on the characteristics of the radar and the UAV and is inversely proportional to the fourth power of the distance between them. In Equation A.2, $R_{(x,y)}$ is the distance between point (x, y) and the radar and all terms except $R_{(x,y)}$ are constant. We can thus write this ratio using the term $C = \frac{P_t G_t^2 \lambda^2 \sigma}{(4\pi)^3 K T_s B_n L_s}$.

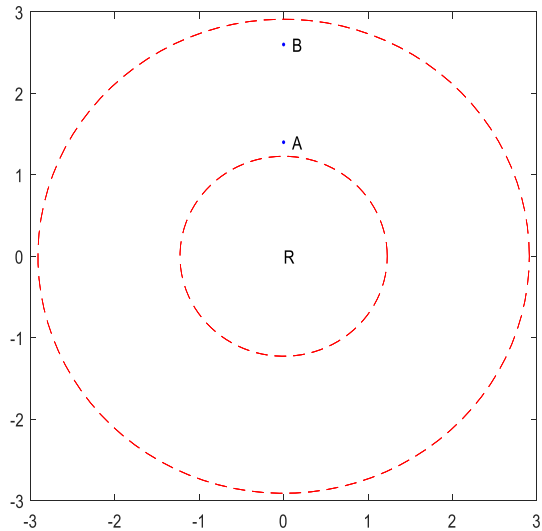
$$S/N_{(x,y)} = \frac{P_t G_t^2 \lambda^2 \sigma}{(4\pi)^3 K T_s B_n L_s R_{(x,y)}^4} = \frac{C}{R_{(x,y)}^4} \tag{A.2}$$

We then find the detection probability at a point (x, y) using Equation A.3.

$$pd_{(x,y)} = \begin{cases} 1 & \text{if } 10 \log_{10}(S/N_{(x,y)}) > UB_{S/N} \\ \frac{10 \log_{10}(S/N_{(x,y)}) - LB_{S/N}}{UB_{S/N} - LB_{S/N}} & \text{if } LB_{S/N} < 10 \log_{10}(S/N_{(x,y)}) \leq UB_{S/N} \\ 0 & \text{if } 10 \log_{10}(S/N_{(x,y)}) \leq LB_{S/N} \end{cases} \tag{A.3}$$

We present an example that shows the signal-to-noise ratios and the detection probabilities of two points, A and B, in the radar’s effective area in Fig. 9. Here, $R_A = 1.4$ and $R_B = 2.6$. For $C = 2270$, $LB_{S/N} = 15$, $UB_{S/N} = 30$, the signal-to-noise

Fig. 9 Two points (A and B) in the effective regions of the radar



ratios of the two points are $S/N_{(A)} = 590.90$ and $S/N_{(B)} = 49.67$, yielding $pd_{(A)} = 0.85$ and $pd_{(B)} = 0.13$.

The radar detection threat measure is not a probability. It is rather the integration of the detection probabilities along the trajectory of the UAV.

If there are multiple radars on the trajectory, the radar that poses the highest risk is considered as effective as in Tezcaner Öztürk and Köksalan (2023).

Acknowledgements This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-16-1-0333.

References


- Besada-Portas E, de la Torre L, de la Cruz JM, Andres-Toro B (2010) Evolutionary trajectory planner for multiple UAVs in realistic scenarios. *IEEE Trans Rob* 26(4):619–634
- Chen X, Xu G (2016) The path planning algorithm studying about UAV attacks multiple moving targets based on Voronoi diagram. *Int J Control Autom* 9(1):281–292
- Dasdemir E, Köksalan M, Tezcaner Öztürk D (2020) A flexible reference point-based multi-objective evolutionary algorithm: an application to the UAV route planning problem. *Comput Oper Res*. <https://doi.org/10.1016/j.cor.2019.104811>
- Desrochers M, Laporte G (1991) Improvements and extensions to the Miller-Tucker-Zemlin Subtour elimination constraints. *Oper Res Lett* 10:27–36
- Dolicanic E, Fetahovic I, Tuba E, Capor-Hrosik R, Tuba M (2018) Unmanned combat aerial vehicle path planning by brain storm optimization algorithm. *Stud Inform Control* 27(1):15–24
- Glade D (2000). Unmanned aerial vehicles: implications for military operations. Technical Report. DTIC Document
- Gudaitis MS (1994). *Multi criteria mission route planning using a parallel A* Search*. Master's thesis, Air Force Institute of Technology

- Jayaweera HMPC, Hanoun S (2022) Path planning of unmanned aerial vehicles (UAVs) in windy environments. *Drones* 6:101. <https://doi.org/10.3390/drones6050101>
- Köksalan M, Lokman B (2009) Approximating the nondominated frontiers of multi-objective combinatorial optimization problems. *Naval Res Logist* 56:191–198
- Köksalan M, Soylu B (2010) Bicriteria p-Hub location problems and evolutionary algorithms. *INFORMS J Comput* 22:499–654
- Köksalan M, Tezcaner Öztürk D (2017) An evolutionary approach to generalized biobjective traveling salesperson problem. *Comput Oper Res* 79:304–313
- Li S, Xiuxia S, & Xu Y (2006). Particle swarm optimization for route planning of unmanned aerial vehicles. In: *IEEE International conference on information acquisition* (pp. 1213–1218). Weihai: IEEE
- Miller CE, Tucker AW, Zemlin RA (1960) Integer programming formulation of traveling salesman problems. *J Assoc Comput Mach* 7(4):326–329
- Mufalli F, Batta R, Nagi R (2012) Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans. *Comput Oper Res* 39(11):2787–2799
- Natalizio E, Zema NR, Yanmaz E, Pugliese LDP, Guerriero F (2020) Take the field from your smartphone: leveraging UAVs for event filming. *IEEE Trans Mob Comput* 19(8):1971–1983
- Peng X, Yang S, Xu D, Gao X (2013) Evolutionary algorithms for the multiple unmanned aerial combat vehicles anti-ground attack problem in dynamic environments. In: Yang S, Yao X (eds) *Evolutionary computation for dynamic optimization problems*. Springer-Verlag, Berlin, pp 403–431
- Pfeiffer B, Batta R, Klamroth K, Nagi R (2009) Path planning for UAVS in the presence of threat zones using probabilistic modelling. In: Badiru AB, Thomas MU (eds) *Handbook of military industrial engineering*. CRC Press, USA
- Qu Z, & Xi X (2010) Cooperative UAV trajectory planning with multiple dynamic targets. In: *Guidance, navigation, and control conference*. Ontario: AIAA
- Rojas Vilorio D, Solano-Charris EL, Muñoz-Villamizar A, Montoya-Torres JR (2021) Unmanned aerial vehicles/drones in vehicle routing problems: a literature review". *Int Trans Oper Res* 28:1626–1657
- Santin R, Assis L, Vivas A, Pimenta LCA (2021) Matheuristics for multi-UAV routing and recharge station location for complete area coverage. *Sensors* 21(5):1705
- Swartzentruber L, Foo JL, & Winer EH (2009). Three-Dimensional Multi-Objective UAV Path Planner Using Terrain Information. In: *The 50th structures, structural dynamics, and materials conference*. Palm Springs: AIAA
- Tezcaner D, Köksalan M (2011) An Interactive algorithm for multi-objective route planning. *J Optim Theory Appl* 150:379–394
- Tezcaner Öztürk D, Köksalan M (2016) An interactive approach for biobjective integer programs under quasiconvex preference functions. *Ann Oper Res* 244(2):677–696
- Tezcaner Öztürk D, Köksalan M (2023) Biobjective route planning of an unmanned air vehicle in continuous space. *Transp Res Part B: Methodol* 168:151–169
- Türeci H (2017) Interactive approaches for bi-objective UAV route planning in continuous space. Middle East Technical University, Ankara
- Venkatachalam S, Sundar K, Rathinam S (2018) A two-stage approach for routing multiple unmanned aerial vehicles with stochastic fuel consumption. *Sensors* 18:3756
- Yao M, Zhao M (2015) Unmanned aerial vehicle dynamic path planning in an uncertain environment. *Robotica* 33(3):1–11
- Zheng C, Li L, Xu F, Sun F, Ding M (2005) Evolutionary route planner for unmanned air vehicles. *IEEE Trans Rob* 21(4):609–620
- Zhenhua W, Weiguoz Z, Jingping S, & Ying H (2008). UAV route planning using multiobjective ant colony system. In: *IEEE Conference on cybernetics and intelligent systems* (pp. 797-800). Chengdu: IEEE

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Nail Karabay¹ · Murat Köksalan^{1,2} · Diclehan Tezcaner Öztürk³ 

Nail Karabay
nailkarabay@gmail.com

Murat Köksalan
koksalan@umich.edu

¹ Department of Industrial Engineering, Middle East Technical University, 06800 Ankara, Turkey

² Technology and Operations, Ross School of Business, University of Michigan, Ann Arbor, MI 48109, USA

³ Department of Industrial Engineering, Hacettepe University, 06800 Ankara, Turkey