Cooperative Mission and Path Planning for a Team of UAVs

62

Hyondong Oh, Hyo-Sang Shin, Seungkeun Kim, Antonios Tsourdos, and Brian A. White

Contents

Abstract

This chapter addresses the cooperative mission and path-planning problem of multiple UAVs in the context of the vehicle-routing problem. Since the conventional vehicle-routing algorithms approximate their path to straight lines

H. Oh (✉) • H-S. Shin • A. Tsourdos • B.A. White
Department of Engineering Physics, School of Engineering, Cranfield University, Cranfield, Bedfordshire, UK
e-mail: h.oh@cranfield.ac.uk; h.shin@cranfield.ac.uk; a.tsourdos@cranfield.ac.uk; b.a.white@cranfield.ac.uk

S. Kim
Department of Aerospace Engineering, Chungnam National University, Daejeon, South Korea
e-mail: skim78@cnu.ac.kr

K.P. Valavanis, G.J. Vachtsevanos (eds.), *Handbook of Unmanned Aerial Vehicles*,
DOI 10.1007/978-90-481-9707-1_14,

to reduce computational load, the physical constraints imposed on the vehicle are not to be taken into account. In order to mitigate this issue, this chapter describes a framework allowing integrated mission and path planning for coordinating UAVs using the Dubins theory based on the differential geometry concepts which can consider non-straight path segments. The main advantage of this approach is that the number of design parameters can be significantly reduced while providing the shortest, safe, and feasible path, which leads to a fast design process and more lightweight algorithms. In order to validate the integrated framework, cooperative mission and path-planning algorithms for two missions are developed: (1) road-network search route-planning patrolling every road segment of interest efficiently based on the optimization and approximation algorithm using nearest insertion and auction negotiation and (2) communication-relay route planning between a ground control station and the friendly fleet satisfying the constraints on the UAV speed and the avoidance of nonflying zones. Lastly, the performance of the proposed algorithms is examined via numerical simulations.

62.1 Introduction

The large scale of UAV applications has proliferated vastly within the last few years with the fielding of Global Hawk, Pioneer, Pathfinder Raven, and Dragoneye UAVs, among others (Samad et al. 2007). The operational experience of UAVs has proven that their technology can bring a dramatic impact to the battlefield. This includes, but is not limited to, obtaining real time, relevant situational awareness before making contact; helping commanders to lead appropriate decision making; and reducing risk to the mission and operation.

Groups of UAVs are of special interest due to their ability to coordinate simultaneous coverage of large areas or cooperate to achieve common goals. The intelligent and autonomous cooperation of multiple UAVs operating in a team/swarm offers revolutionary capabilities: improved situation awareness; significant reductions in manpower and risk to humans; the ability to perform in hostile, hazardous, and geometrically complex environments; and cost efficiency. Specific applications under consideration for groups of cooperating UAVs include, but are not limited to, border patrol, search and rescue, surveillance, mapping, and environmental monitoring. In these applications, a group of UAVs becomes a mobile resource/sensor and, consequently, tasks and routes of each UAV need to be efficiently and optimally planned in order to cooperatively achieve their mission. Therefore, this chapter addresses the cooperative mission and path-planning problem, which here is considered as a vehicle-routing problem of multiple UAVs for given missions.

Since general vehicle-routing algorithms approximate their paths to straight lines in order to reduce the computational load, the physical constraints imposed on the vehicle are not to be taken into account. In order to mitigate this issue, this chapter describes a framework which allows integrated mission and path planning for coordinating UAVs. One of key enablers of this approach is the path-planning scheme which was developed in a previous study (Tsourdos et al. 2010) based on the differential geometry concepts, especially Dubins paths. Path-planning algorithms using differential geometry examine the evolution of guidance geometry over time to derive curvature satisfying UAV constraints. Guidance commands defining a maneuver profile can be then computed using the derived curvature of the guidance geometry. One of the main advantages of this approach is that the number of design parameters can be significantly reduced while maintaining the guidance performance. Therefore, this approach will enable not only a fast design process and more lightweight algorithms but will also generate safe and feasible paths for multiple UAVs. This is required for the integration of operational and physical constraints of the UAVs into the cooperative mission and path-planning solution.

Road-network search and communication-relay problems are also addressed to validate the integrated framework of mission and path planning. The road-network search routing problem enables multiple airborne platforms to efficiently patrol every road segment identified in the map of interest. This problem has been mainly handled in the operational research area (Gibbons 1999; Ahr 2004; Gross and Yellen 2003; Bektas 2006) and can be generally classified into two categories: one is the traveling salesman problem (TSP) which finds a shortest circular trip through a given number of cities, and the other is the Chinese postman problem (CPP) which finds the shortest path to travel along each road in the road network. The TSP using multiple UAVs can be considered as a task assignment problem to minimize the mission time or energy by assigning each target to an UAV, for which binary linear programming (Bellingham et al. 2003), iterative network flow (Chandler et al. 2002), tabu search algorithm (Ryan et al. 1998), and receding horizon control (Ahmadzadeh et al. 2006) have been proposed. Recently, Royset and Sato (2010) proposed a route optimization algorithm for multiple searchers to detect one or more probabilistically moving targets incorporating other factors such as environmental and platform-specific effects. Meanwhile, the CPP is normally used for ground vehicle applications such as road maintenance, snow disposal (Perrier et al. 2007), boundary coverage (Easton and Burdick 2005), and graph searching and sweeping (Alspach 2006; Parsons 1976).

The communication-relay problem described in this chapter is used to extend the mission area of a main friendly fleet (FF) such as aircraft and ground convoys using relatively inexpensive subsystems, swarms of UAVs. Therefore, the focus is on cooperative route planning of multiple UAVs to maintain communication between the FF and a ground control station (GCS). UAVs in this problem are used as an effective communication-relay platform in environments characterized by poor radio frequency connectivity, which includes urban, forested, or mountainous regions where no line-of-sight exists between ground transmitters and receivers (Cerasoli and Eatontown 2007). In the late 1990s, a feasibility study which uses the UAV as a communication relay develops a Battlefield Information Transmission System (BITS) system (Pinkney et al. 1996). The main objective of this study was to provide beyond line of sight communications within an area of operations without using scarce satellite resources. The study predicted that, with the advances in miniaturization technology and improved transmitter efficiencies, the UAV could carry multifunction and multiband transponders to handle the communication relay within size, weight, and power dissipation budgets. Cerasoli (Cerasoli and Eatontown 2007) assessed the practical effectiveness of a UAV communication relay in an urban area using a ray tracing method. The paper concluded that a UAV at 2,000 m provided coverage for over 90 % of the ground receivers with 10 dB of LOS path loss by analysis of UAVs placed at various positions and heights over an approximately 500-m^2 urban area.

The remainder of this chapter is organized as follows. Section 62.2 introduces an overview of path planning using Dubins paths and explains how to allow for physical constraints of the fixed-wing UAVs in Dubins path planning. Then, Sect. 62.3 proposes road-network search route-planning algorithms for multiple UAVs based on mixed integer linear programming (MILP) optimization and the approximation algorithm using nearest insertion and auction negotiation. Section 62.4 describes route-planning and decision-making algorithms for a group of UAVs in order to guarantee communication between the GCS and the FF. Simulation results and analysis of each problem are also included in each section. Conclusions are given in Sect. 62.4.4.2.

62.2 Path Planning Using Dubins Paths

Several path-planning methodologies based on differential geometry have been proposed: Dubins curves, clothoid arcs, and Pythagorean hodograph curves (Shanmugavel et al. 2007; Tsourdos et al. 2010). This section will briefly introduce the concept of Dubins path planning based on the reference (Shanmugavel 2007; Kim et al. 2011), and the detailed algorithm will be developed to design flyable and safe Dubins path transiting between waypoints for cooperative missions of multiple UAVs. The Dubins path is considered because it is the shortest path, has simple geometry, and is computationally efficient.

Path planning is defined as the geometric evolution of curves between two desired poses in free space, C_{free}. The pose in 2D comprises the position coordinates (x, y) and the orientation θ. A simple case of producing path between two poses is first considered. This can be extended into any number of waypoints/poses:

$$P_s(x_s, y_s, \theta_s) \xrightarrow{r(q)} P_f(x_f, y_f, \theta_f), \quad r(q) \in C_{\text{free}}, \quad |\kappa(t)| \leq \kappa_{\max} \qquad (62.1)$$

where P_s and P_f denote a starting and final pose, respectively, $\kappa(t)$ represents the curvature of path $r(q)$ with parameter q, and $\kappa_{\max}$ is the maximum curvature imposed by the UAV dynamic constraints such as maximum lateral acceleration. Motion in the plane is composed of either rectilinear or turning or angular motions. A straight line provides the shortest rectilinear motion, and the circular arc provides the shortest turning or angular motion. Also, the arc provides a constant turning radius which also satisfies the maximum curvature constraint, that is, the minimum turning radius, which is a function of speed and maximum lateral acceleration. The Dubins path (Dubins 1957) is the shortest path between two vectors in a plane, and the path meets the minimum bound on turning radius. The Dubins path is a composite path formed either by two circular arcs connected by a common tangent or three consecutive tangential circular arcs or a subset of either of these two. The first path is a CLC path, and the second one is a CCC path, where "C" stands for circular segment and "L" stands for line segment. Either of these two curves will form the shortest path between two poses and so is a good approach for UAV path planning. This section focuses on Dubins paths of CLC type, and the details of CCC type paths can be found in Oh et al. (2011a).

62.2.1 Generating Dubins Path

The Dubins path can be produced by solving (Eq. 62.1). However, it is computationally efficient if it is produced by geometric principles owing to its simple geometry that it is formed by two circular arcs connected by their common tangents. Therefore, the principles of analytic geometry are used to produce the Dubins paths. There are two possible tangents between the arcs: (1) an external tangent, where the start and finish maneuvers have same turning directions, and (2) an internal tangent, where the turning maneuvers have opposite turning directions (e.g., if the start maneuver is clockwise, the finish maneuver will be anticlockwise and vice versa). Here, only the Dubins path with an external tangent is derived (the case of an internal tangent is analogous). Referring to Fig. 62.1, the input parameters for producing the Dubins path are:

i) Initial pose: $P_s(x_s, y_s, \theta_s)$
ii) Finish pose: $P_f(x_f, y_f, \theta_f)$
iii) Initial turning radius: $\rho_s (= \frac{1}{\kappa_s})$
iv) Finish turning radius: $\rho_f (= \frac{1}{\kappa_f})$

1. Find the centers of the turning circles $O_s(x_{cs}, y_{cs})$ and $O_f(x_{cf}, y_{cf})$:

$$(x_{cs}, y_{cs}) = (x_s \pm \rho_s \cos(\theta_s \pm \pi/2), y_s \pm \rho_s \sin(\theta_s \pm \pi/2)) \quad (62.2a)$$

$$(x_{cf}, y_{cf}) = (x_f \pm \rho_f \cos(\theta_f \pm \pi/2), y_f \pm \rho_f \sin(\theta_f \pm \pi/2)) \quad (62.2b)$$

   where O_s and O_f are called primary circles represented by C_s and C_f, respectively.
2. Draw a secondary circle of radius $|\rho_f - \rho_s|$ at O_f for $\rho_s \leq \rho_f$.

3. Connect the centers O_s and O_f to form a line c, called center line, where $|c| = \sqrt{(x_{cs} - x_{cf})^2 + (y_{cs} - y_{cf})^2}$.
4. Draw a line O_sT' tangent to the secondary circle C_{sec}.
5. Draw a perpendicular straight line from O_f to O_sT', which intersects the primary circle C_f at T_N, which is called as a tangent entry point on C_f.
6. Draw a line T_XT_N parallel to O_sT', where T_X is called as a tangent exit point on C_s.
7. Connect the points P_s and T_X by an arc of radius ρ_s and T_N and P_f by an arc of radius ρ_f.
8. The composite path is then formed by the starting arc P_sT_X, followed by the external tangent line T_XT_N and the finishing arc T_NP_f.

The triangle $\triangle O_sO_fT'$ is a right-angled triangle with hypotenuse O_sO_f, and the other two sides are O_fT' and O_sT', where $||O_fT'|| = |\rho_f - \rho_s|$. The included angle between O_sO_f and O_sT' is ϕ_e, where

$$\phi_e = \arcsin\left(\frac{\rho_f - \rho_s}{|c|}\right) \tag{62.3}$$

The slope of the line c is ψ, where

$$\psi = \arctan\left(\frac{y_{cf} - y_{cs}}{x_{cf} - x_{cs}}\right) \tag{62.4}$$

The angles $\phi_{ex} = \angle(X_1O_sT_X$ and $\phi_{en} = \angle(X_2O_fT_N$ are calculated from Table 62.1 where $\overrightarrow{O_sX_1}$ and $\overrightarrow{O_fX_2}$ are parallel to the positive

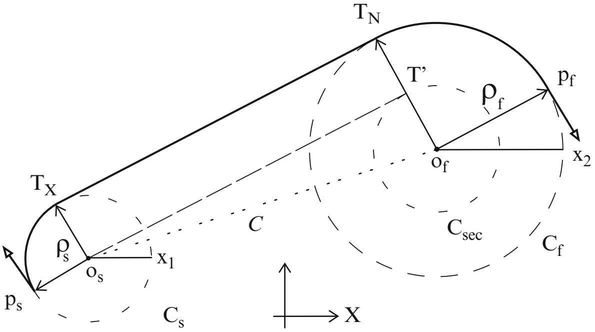


Fig. 62.1 Dubins – design of CLC path

Table 62.1 Calculation of tangent exit and entry points

Start-turn	Finish-turn	ϕ_e	ϕ_{ex}	ϕ_{en}
Clockwise	Clockwise	$\arcsin\left(\frac{\rho_f - \rho_s}{c}\right)$	$\phi_e + \frac{\pi}{2} + \psi$	$\phi_e + \frac{\pi}{2} + \psi$
Anticlockwise	Anticlockwise	$\arcsin\left(\frac{\rho_f - \rho_s}{c}\right)$	$\phi_e - \frac{\pi}{2} + \psi$	$\phi_e - \frac{\pi}{2} + \psi$

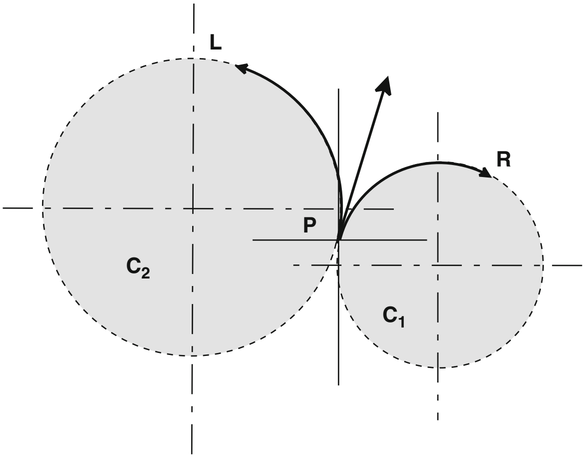


Fig. 62.2 Tangent circles

direction of x-coordinate. The values of ϕ_{ex} and ϕ_{en}, the tangent exit and entry points, are calculated as

$$T_X = \left(x_{cs} + \rho_s \cos(\phi_{ex}), y_{cs} + \rho_s \sin(\phi_{ex})\right) \tag{62.5a}$$

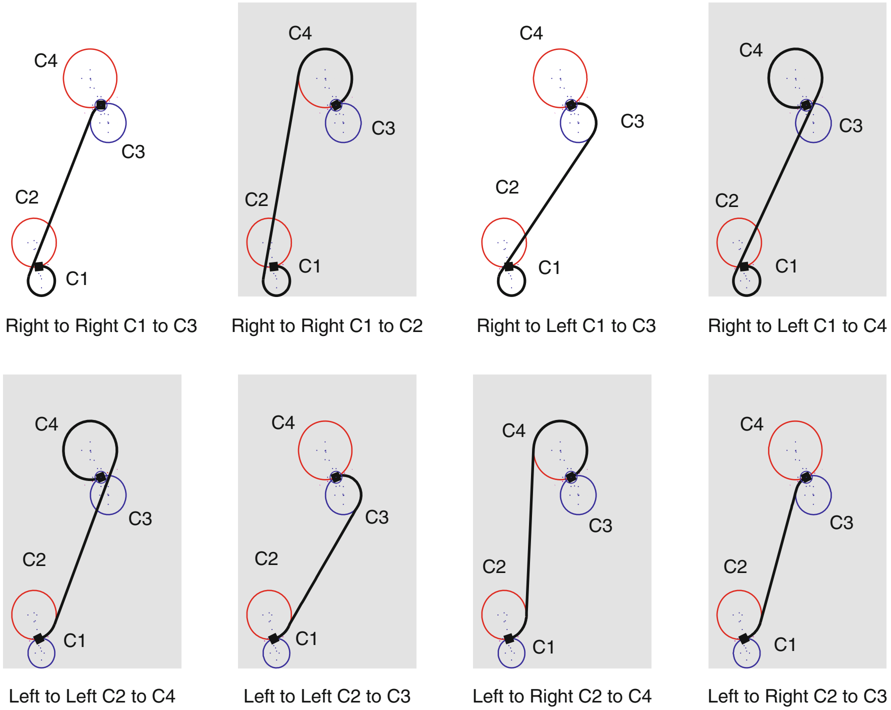
$$T_N = \left(x_{cf} + \rho_f \cos(\phi_{en}), y_{cf} + \rho_f \sin(\phi_{en})\right) \tag{62.5b}$$

From the figure, it can be seen that the turning maneuvers have clockwise rotations. Similarly, other type of Dubins path using the internal tangent can be produced by drawing the secondary circle of radius equal to $|\rho_s + \rho_f|$. It is worth pointing out that the calculation of the tangent exit and entry points T_X and T_N is central in generating the Dubins path.

For a given pose, there are two circles tangent to it. Referring to Fig. 62.2, the pose P has a right turn R on the arc C_1 and a left turn L on the arc C_2. If θ_s and θ_f are fixed, four Dubins paths are possible between P_s and P_f, which are {RSR, RSL, LSR, LSL}, where represents the tangent. However, if the final orientation is taken either $\pm\theta_f$, the number of Dubins paths between P_s and P_f will become eight. Figure 62.3 shows the eight possible Dubins paths: four paths each from the primary circle C_1 to the secondary circles C_3 and C_4 and four from C_2 to C_3 and C_4.

62.2.2 Conditions for the Existence of Dubins Paths

From Sect. 62.2.1, it can be seen that the existence of the Dubins path is determined by the common tangents between the turning arcs. These tangents vanish under two conditions. The external tangent vanishes when one of the primary circles C_s and C_f contains the other, while the internal one vanishes when the primary circles intersect. Both of these conditions together determine the existence of the Dubins



Right to Right C1 to C3 Right to Right C1 to C2 Right to Left C1 to C3 Right to Left C1 to C4

Left to Left C2 to C4 Left to Left C2 to C3 Left to Right C2 to C4 Left to Right C2 to C3

Fig. 62.3 Dubins paths with θ_f as a free variable

path between given two poses, which in turn are a function of the turning radii and hance curvature:

$$\text{External tangent} : (c + \rho_s) > \rho_f, \ \ \rho_f > \rho_s \tag{62.6a}$$

$$\text{Internal tangent} : c > (\rho_s + \rho_f), \ \ \rho_f > \rho_s \tag{62.6b}$$

62.2.3 Length of Dubins Paths

The Dubins path is a composite path of two circular arcs and a straight line. Hence, the path length s_{Dubins} is the sum of the lengths of individual path segments. Since the length of the common tangent connecting the arcs is determined by the radii of the arcs, the length is also the function of the turning radii. Hence, the length of the path can be varied by changing the radii (curvatures). Also, any two paths can be made equal in length by simply varying the curvature of the arcs:

$$s_{\text{Dubins}} = s_s + s_t + s_f \tag{62.7a}$$

$$= |\rho_s \alpha_s| + \|T_X T_N\| + |\rho_f \alpha_f| \tag{62.7b}$$

$$s_{\text{Dubins}} = f(\rho_s, \rho_f) \tag{62.7c}$$

where s_{Dubins} is the length of the Dubins path, α_s and α_f are the included angles, $\alpha_s = \phi_{ex}$, $\alpha_f = \phi_{en}$, and $\|T_X T_N\| = \sqrt{(y_N - y_X)^2 + (x_N - x_X)^2}$. Using this length of the Dubins path, a reference velocity for each UAV can be computed in order to control the time taken for the UAV to traverse each path for cooperative mission planning.

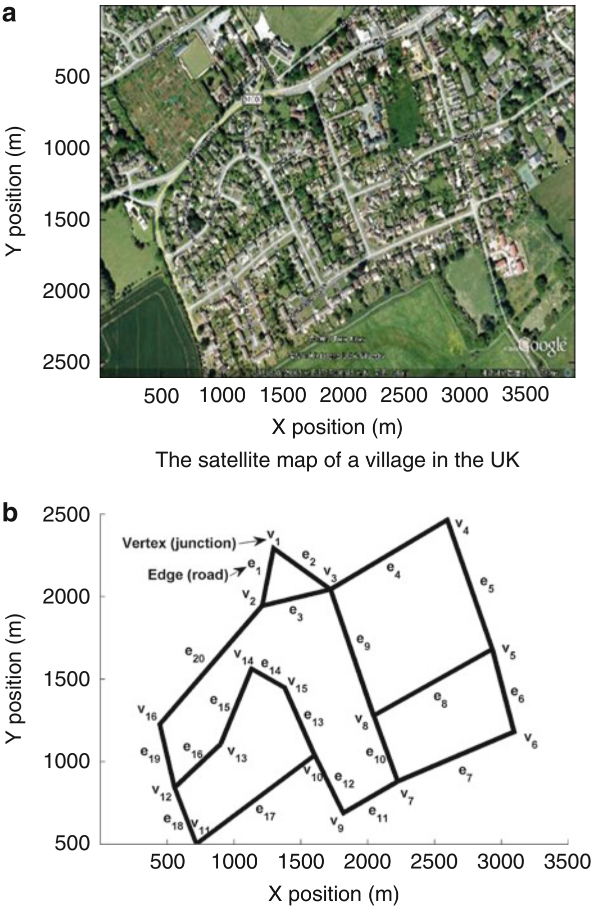
62.3 Cooperative Mission Planning I: Road-Network Search Route Planning

For a road search route-planning mission, a road network is established as a set of straight line connecting a set of waypoints. These waypoints are located either on road junctions or along the roads with sufficient separations between them to allow for accurate representation of a curved road by using a set of straight lines. A sample road network is shown in Fig. 62.4a, which is based on the Google map of a village in the UK. This road network can be translated to a graph composed of straight line segments connecting a set of vertices, as shown in Fig. 62.4b (Oh et al. 2011b). In order to search all the roads within the map of interest, there are generally two typical routing problems (Ahr 2004):

Traveling Salesman Problem (TSP): A salesman has to visit several cities (or road junctions). Starting at a certain city, the TSP finds a route with minimum travel distance on which the salesman traverses each of the destination cities exactly once (and for the closed TSP, leads him back to his starting point). *Chinese Postman Problem (CPP):* A postman has to deliver mail for a network of streets. Starting at a given point, for example, the post office, the CPP finds a route with minimum travel distance which allows the postman to stop by each street at least once (and for the closed CPP, leading him back to the post office).

Consider the CPP and its variants, which involve constructing a tour of all the roads with the shortest distance of the road network. Typically, the road network is mapped to an undirected graph $G = (V, E)$, having edge weights $w : E \rightarrow R_0^+$, where the roads are represented by the edge set $E = \{e_1, \ldots, e_n\}$ and the road junctions are represented by the vertex set $V = \{v_1, \ldots, v_m\}$ as numbered in Fig. 62.4b. Each edge $e_i = \{v_{e_{i,1}}, v_{e_{i,2}}\}$ is weighted with its length or the amount of time required to traverse it. The basic CPP algorithm involves first constructing an even graph from the road network which has a set of vertices with an even number of edges attached to them producing a pair of entry and exit points. When the road-network graph has junctions with an odd number of edges, some roads therefore must be selected as exceptions for multiple visits by the postman to make the even graph. The search pattern (tour) of the even graph is calculated by determining the Euler tour of the graph (Gross and Yellen 2003), which visits every edge of the even graph exactly once.

Fig. 62.4 The graphic representation of a road network. (**a**) The satellite map of a village in the UK. (**b**) Its graphic representation by straight line segments



The satellite map of a village in the UK

Its graphic representation by straight line segments

62.3.1 Road-Network Search Route by Multiple Unmanned Aerial Vehicles

The conventional CPP algorithm has been applied to a fully connected road network for use by ground vehicles. However, since UAVs do not have any restrictions such that they must only move along the roads, the CPP algorithm needs to be modified for the case that UAVs search a general road map having unconnected road segments. The modified CPP (mCPP) generates a tour of the road network traveling all the roads once no matter what the type area of interest map is: an even or odd graph. Even searching the area having no road somewhere in it can be tackled by the mCPP algorithm by generating a virtual road pattern with a lawnmower (Maza and Ollero 2007) or spiral-like (Nigam and Kroo 2008) algorithm.

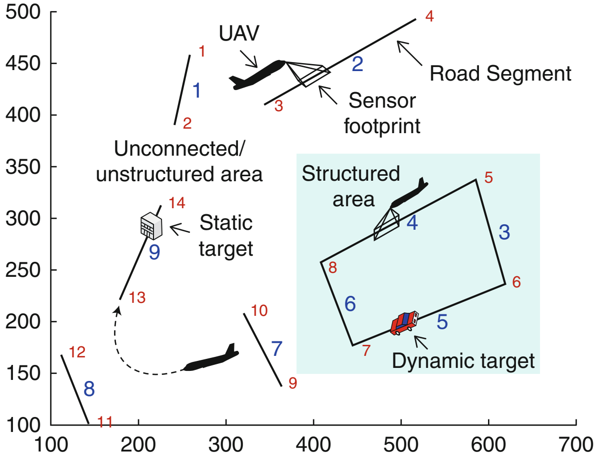


Fig. 62.5 An illustration of the modified Chinese Postman Problem (mCPP)

Figure 62.5 exemplifies a sample road-network search problem to be solved by the mCPP. As the CPP algorithms generally approximate their path to a straight line for simplicity, the mCPP algorithm using a straight line path is called the modified Euclidean CPP (mECPP) for the rest of this chapter. However, in order to produce the shortest path flyable by the UAV that connects the road segment sequence selected by the search route algorithm, the flight constraints of the UAV have to be taken into account. To accommodate this, the Dubins path is incorporated into the mCPP algorithm instead of using just a straight line to connect the roads, and this is called the modified Dubins CPP (mDCPP). A road-network search route-planning algorithm using the optimization via MILP for the mCPP problem is first developed. A new approximation algorithm is then proposed as a more practical approach to reduce the complexity of the algorithm.

62.3.2 Optimization via MILP

The mCPP is solved by use of the MILP optimization using the multidimension multi-choice knapsack problem (MMKP) formulation to find an suboptimal solution minimizing the total flight time of UAVs. The classical MMKP picks up items for the knapsacks to have maximum total value so that the total resource required does not exceed the constraints of knapsacks (Hifi et al. 2006). In order to apply the MMKP to the road-network search, the UAVs are assumed to be the knapsacks, the roads to be searched are the resources, and the limited flight time or energy for each UAV as the capacity of knapsack. The MMKP formulation allows the search problem to consider the characteristics of each UAV such as flight time capacity and minimum turning radius. The details of the proposed road-network search algorithm for multiple UAVs are explained as follows:

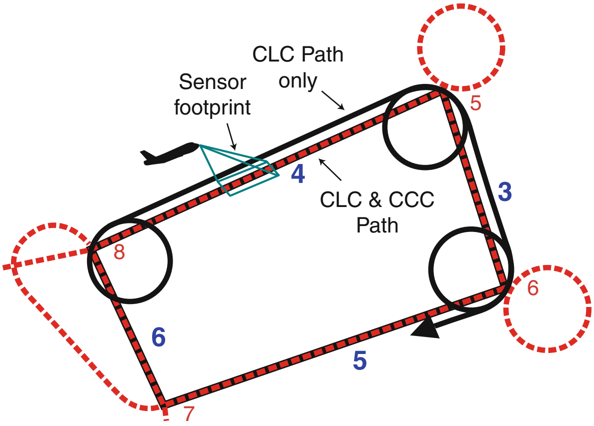


Fig. 62.6 An example of CCC and CLC paths following the road

62.3.2.1 Dubins Path Planning

Once the shortest edge permutations are determined, the next step is to compute and to store the cost (path length or flight time) and to then compute the Dubins paths to connect them. Although the CLC path is being used in general case, this study also explores the CCC path for a densely distributed road environment, because the CLC path cannot be applied in all cases (cf. Sect. 62.2.2). Moreover, to follow the road precisely taking into account the sensor footprint coverage, the path should consist of both CCC and CLC forms of Dubins path. Figure 62.6 shows an example of a road search path using CLC and CCC path for a small sensor footprint, which results in a detour at the road intersection.

62.3.2.2 Generation of the Shortest Edge Permutation

First of all, unordered feasible edge (i.e., road) permutations to be visited by the UAV are generated for all possible cases with a given petal size. The petal size means the maximum number of edges that can be visited by one UAV and is determined by the amount of resources available to it. If the end vertex of one edge and any vertex of the next edge are not connected, they are connected with an additional edge which has a shorter distance. Then, the shortest order-of-visit edge permutations considering the initial position of each UAV are computed under the assumption that a path is represented as a straight line.

62.3.2.3 MMKP Formulation and MILP Optimization

The final step of the proposed algorithm is to allocate the edge permutations to each UAV so as to cover all the roads with a minimum flying time. This can be expressed by a MMKP formulation as

$$\min\ J = \sum_{i=1}^{N_{\text{UAV}}} \sum_{j=1}^{N_{p_i}} T_{ij} x_{ij} \tag{62.8}$$

$$\text{s.t} \sum_{i=1}^{N_{\text{UAV}}} \sum_{j=1}^{N_{p_i}} E_{kj} x_{ij} \geq 1,\ \text{for}\, k = 1, \ldots, N_{\text{edge}} \tag{62.9}$$

$$\sum_{j=1}^{N_{p_i}} x_{ij} = 1,\ x_{ij} \in \{0, 1\},\ \text{for}\, i = 1, \ldots, N_{\text{UAV}},\ j = 1, \ldots, N_{p_i} \tag{62.10}$$

where N_{UAV}, N_{edge}, and N_{p_i} denote the number of UAVs, the edges to be visited, and the permutations generated by the i-th UAV, respectively. T_{ij} represents the mission cost (flight time) of the j-th permutation of the i-th UAV, and E_{kj} represents the matrix whose k-th element of the j-th permutation is 1 if edge k visited; otherwise, 0 and x_{ij} is either 0, implying permutation j of the i-th vehicle is not picked or 1 implying permutation j of the i-th UAV is picked. The first constraint states that the UAVs should visit every edge once or more, and the second constraint represents the allocation of exactly one edge permutation to the each UAV. This MMKP problem is solved using a SYMPHONY MILP solver (Ralphs et al. 2010). It should be noted that, depending on the petal size, the computational burden of the mission cost T_{ij} of all edge permutations can be significant.

62.3.3 Approximation Algorithm

62.3.3.1 Nearest Insertion-Based mDCPP

Due to the complexity of the problem, instead of using the optimization method explained above, an approximation algorithm is developed as a more practical solution to the mCPP (Oh et al. 2011a, 2012). To develop the approximation algorithm, the TSP algorithm is first studied. Although there are a lot of algorithms for the TSP (Rosenkrantz et al. 2009), one heuristic approach, a nearest insertion method, is adopted here since it is fast and easy to implement. The basic idea of the insertion method is to construct the approximate tour by a sequence of steps in which tours are constructed for progressively larger subsets of the nodes of the graph. It produces a tour no longer than twice the optimal regardless of the number of nodes in the problem and runs in a time proportional to the square of the nodes (Rosenkrantz et al. 2009). Having this in mind, the nearest insertion-based mDCPP (NI-mDCPP) algorithm is developed as illustrated in Fig. 62.7. The NI-mDCPP algorithm for the single UAV is described as follows.

Algorithm Description

1. Start from a certain point or road junction and select the nearest road to it using the Dubins path length.
2. Make and grow a tour by finding the nearest road to any of the selected tour roads.

3. Compute the cost of insertion to decide whether to insert before or stack after the closest road to the tour.
4. Insert the selected road in the decided position.
5. $2 \sim 4$ are repeated until all roads are included in the tour.

62.3.3.2 Euclidean Distance Order Approximation

To reduce the computation burden further for the dynamic environment, an additional approximation algorithm which uses the Euclidean distance order is incorporated into the NI-mDCPP algorithm. This algorithm is described as follows. First of all, make an ascending order of road list for both the Euclidean distance and the Dubins path length between all pairs of end points of the road network, and find the maximum number, $n_{\text{order,max}}$ which guarantees that road list of Euclidean distance within that number contains the shortest Dubins path. Note that although $n_{\text{order,max}}$, is determined before running the algorithm given information of the road network, a size of $n_{\text{order,max}}$ would remain almost the same against minor changes of road information for an uncertain dynamic environment. Then, when finding the nearest roads, make the ascending order list of distances between the edge of interest and all the other roads using the Euclidean distance first, and find the nearest road whose Dubins path length is the shortest among the roads in the $n_{\text{order,max}}$ Euclidean distance order. In other words, this method computes only $n_{\text{order,max}}$ Dubins path distances instead of computing all the Dubins lengths between one road the other roads.

62.3.3.3 Negotiation for Multiple UAVs

Having developed the NI-mDCPP algorithm for the single UAV, it can be extended to the case of multiple UAVs using auction-based negotiation. The algorithm is described as follows.

Algorithm Description

1. Start with N initial positions or roads of N UAVs.
2. Make and grow a tour by using single NI-mDCPP algorithm while storing the cost (path length or flight time) between selected tour roads and remaining edges (which was needed for finding the nearest vertex).
3. When conflict occurs, that is, more than one UAV wants the same road for the next tour, the auction algorithm using stored cost is used to match UAVs with the task (road) to minimize the cost.

Figure 62.8 illustrates the procedure of the algorithm. Since road 3 is not searched yet in Fig. 62.8d, each UAV sends its cost for the given task (in this case, visiting road 3), and then an auction or bipartite (linear programming) approach is used to match UAVs with the task to minimize the cost. Although overlapping road segments are avoided using the auction algorithm, collision between UAVs might occur during the transition from one road to the next. In that case, if necessary, the path can be replanned either by visiting a new road or by modifying the curvature of the arc of the Dubins path (Tsourdos et al. 2010). For simplicity, it is assumed that the collision avoidance is done by a local flight controller or by operating the UAVs

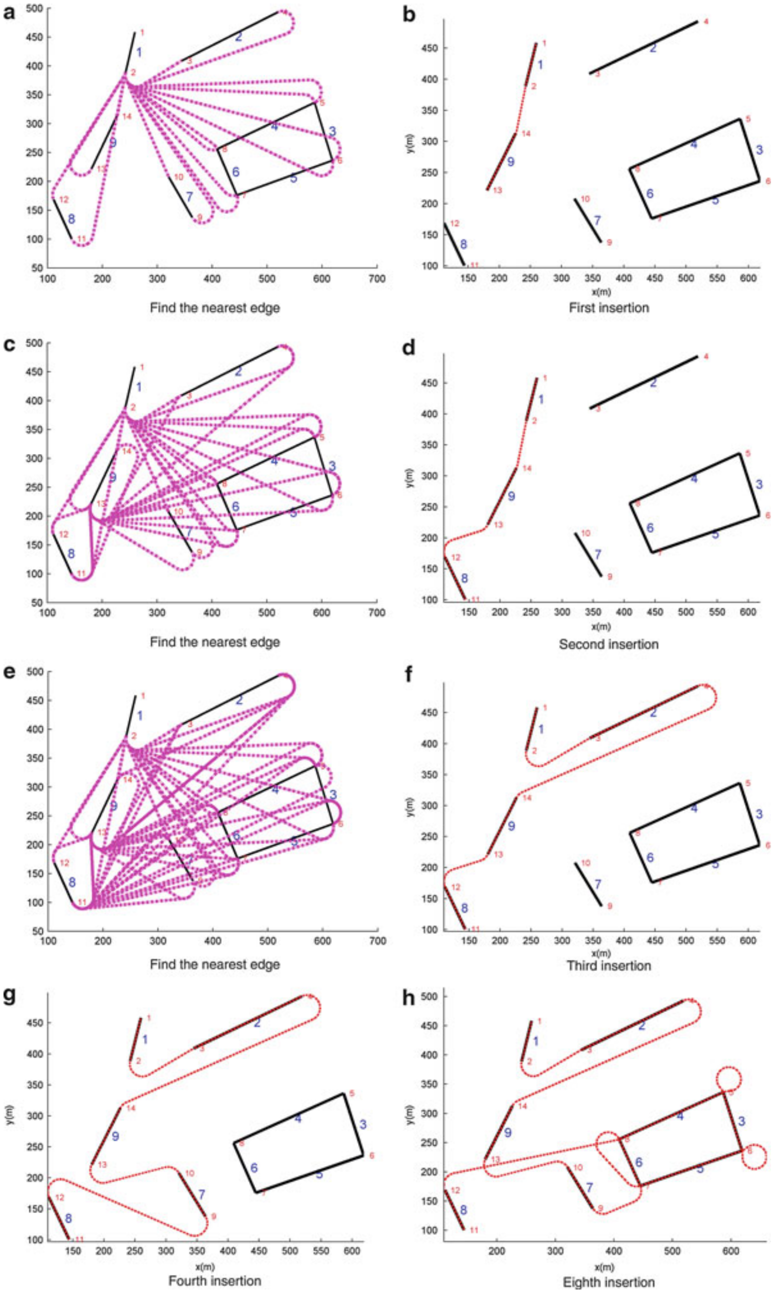
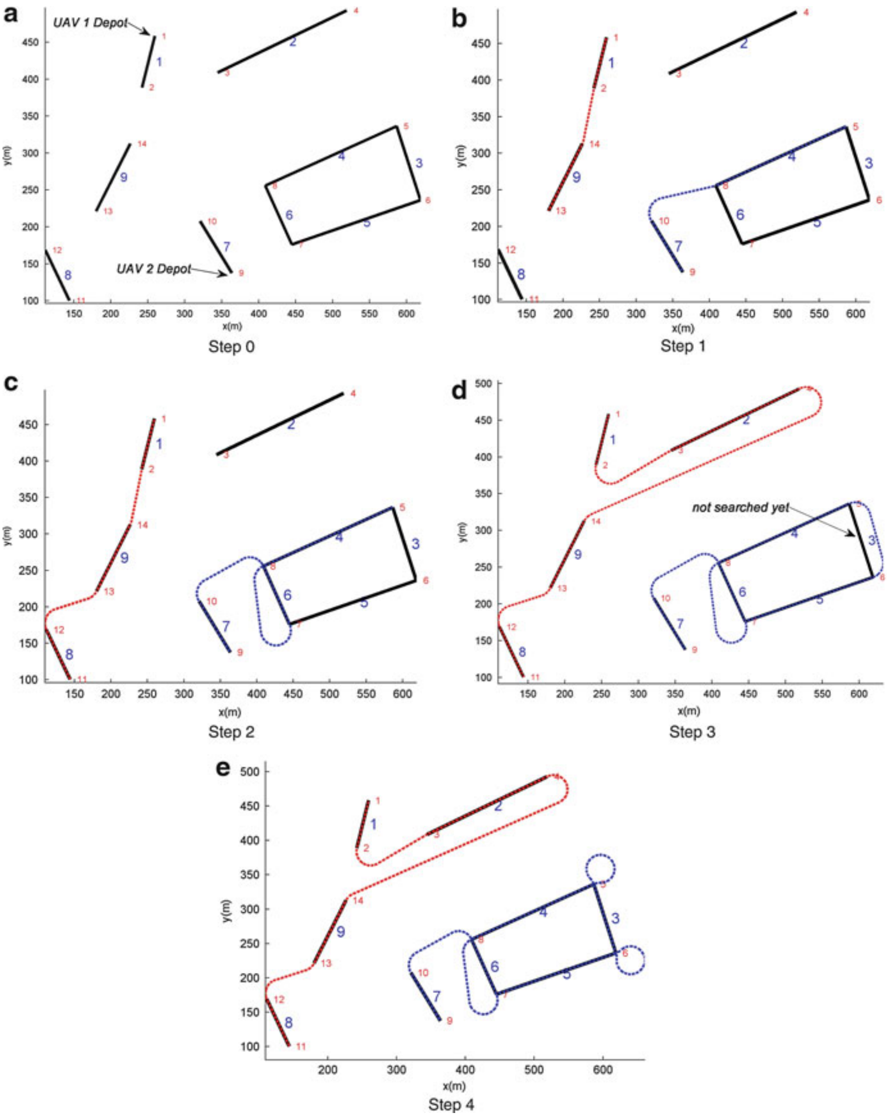


Fig. 62.7 An example simulation of the NI-mDCPP road search algorithm. (**a**) Find the nearest edge. (**b**) First insertion. (**c**) Find the nearest edge. (**d**) Second insertion. (**e**) Find the nearest edge. (**f**) Third insertion. (**g**) Fourth insertion. (**h**) Eighth insertion



Step 0

Step 1

Step 2

Step 3

Step 4

Fig. 62.8 Negotiation procedure for multiple UAVs. (**a**) Step 0. (**b**) Step 1. (**c**) Step 2. (**d**) Step 3. (**e**) Step 4

at different altitudes. The proposed algorithm is rather simple but straightforward and can be run in real time. Moreover, by including additional factors such as different minimum turning radii and total path length (or the number of roads) assigned to each UAV into the cost, the auction-based negotiation can be made very flexible and can deal with heterogeneous UAVs.

62.3.4 Numerical Simulations

62.3.4.1 Performance Comparison

To evaluate the performance of the proposed road-network search algorithms for multiple UAVs, numerical simulations are performed for a specific scenario with four UAVs, and the road network shown in Fig. 62.4a. Each UAV has different dynamic constraints given by:

- Minimum turning radius $\rho_{\min}$: [100 90 80 70] m
- Maximum cruise speed $V_{c,\max}$: [60 50 40 30] m/s

The maximum curvature $\kappa_{\max}$ of the UAVs are given by $\kappa_{\max} = 1/\rho_{\min}$. The UAVs are assumed to have maximum cruising speed during the entire mission, and the maximum petal size of the edge permutation is set to five. Figure 62.9a shows the result of the road-network search using MILP optimization. The flight path is smooth and flyable due to the Dubins path planning, and since the UAV does not need to only fly along the road, the results include additional paths connecting some of the unconnected roads. The total flight length of all UAVs is 2,798.1 m, and its flight duration is 583.1 s. In this scenario, the total computation time exceeds a reasonable limit (>5 min) using a normal PC system (Core 2 CPU, 2.0 Ghz, and 512 MB RAM). Figure 62.9b shows the search results of the NI-mDCPP algorithm. The NI-mDCPP gives a solution within less than a second having about 12 % longer flight time (653.4 s) than that of the MILP optimization. Considering both the computation time and performance, the NI-mDCPP can be regarded as a preferable approach for the given sample map or a more complex scenario.

62.3.4.2 Monte Carlo Simulation of Approximation Method

To evaluate the properties and performance of the proposed approximation algorithm, Monte Carlo simulations are performed using a random map with different parameters defining the map size and the number of UAVs. The map environment is composed of 10 by 10 vertices, numbered as shown in Fig. 62.10, and the road edges are generated by connecting two randomly selected vertices. To check the impact of the map size on the Dubins path planning, the distance d_{map} between the adjacent vertex is set to be proportional to the minimum turning radius $\rho_{\min}$ of the UAV as

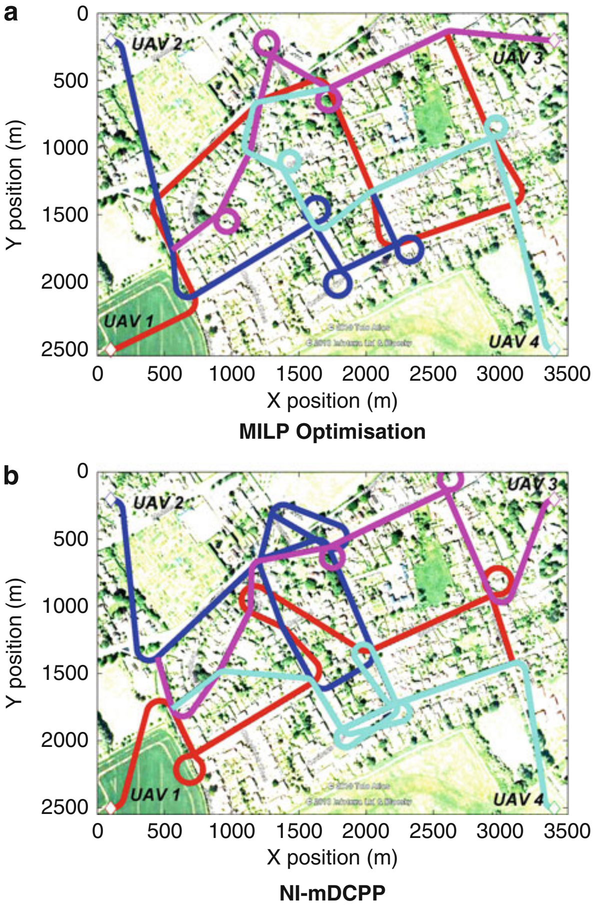
$$d_{\text{map}} = K_s \times \rho_{\min} \tag{62.11}$$

where K_s is the scale factor. Moreover, some of the selected edges whose lengths are longer than three times d_{map} are discarded to get a well-distributed road network and to distribute the roads to each UAV with a similar length. Figure 62.10 shows the sample road network with 20 randomly chosen edges. By Monte Carlo simulations, the effect of three major factors for the road-network search route planning can be investigated. These are:

The distribution density of road network: densely or sparsely distributed road relative to the minimum turning radius of UAVs

The type of path planning: straight line or Dubins path

Fig. 62.9 Road-network search route-planning results using multiple UAVs. (**a**) MILP optimization. (**b**) NI-mDCPP



The number of UAVs: computation time, path length, and the longest path length of a single UAV

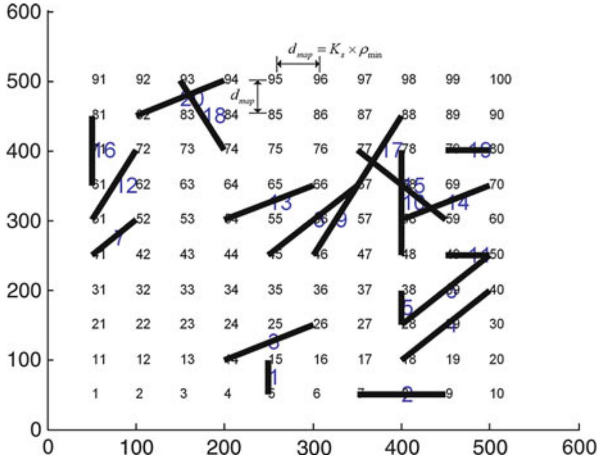
This will provide information on priorities for the efficient use of the UAV group in the planning phase of the autonomous search mission.

In the simulations, the UAVs are assumed to have a constant velocity and minimum turning radius of $\rho_{\min} = 50\,\text{m}$. The simulation results are the average of 50 runs.

Single UAV Case

The first set of simulations are performed using a single UAV with different road map scales. For the rest of this section, the terms of mDCPP and the mECPP are used for the NI-mDCPP and the NI-mECPP, respectively. One of the search route-planning results using the mDCPP for a random map is shown in Fig. 62.11, which covers all the roads satisfying turning constraints of UAVs. Figure 62.12a displays the computation time ratio between the Dubins path (mDCPP) and the

Fig. 62.10 A sample road network with 20 randomly chosen edges ($K_s = 1,\ \rho_{\min} = 50\,\text{m}$)



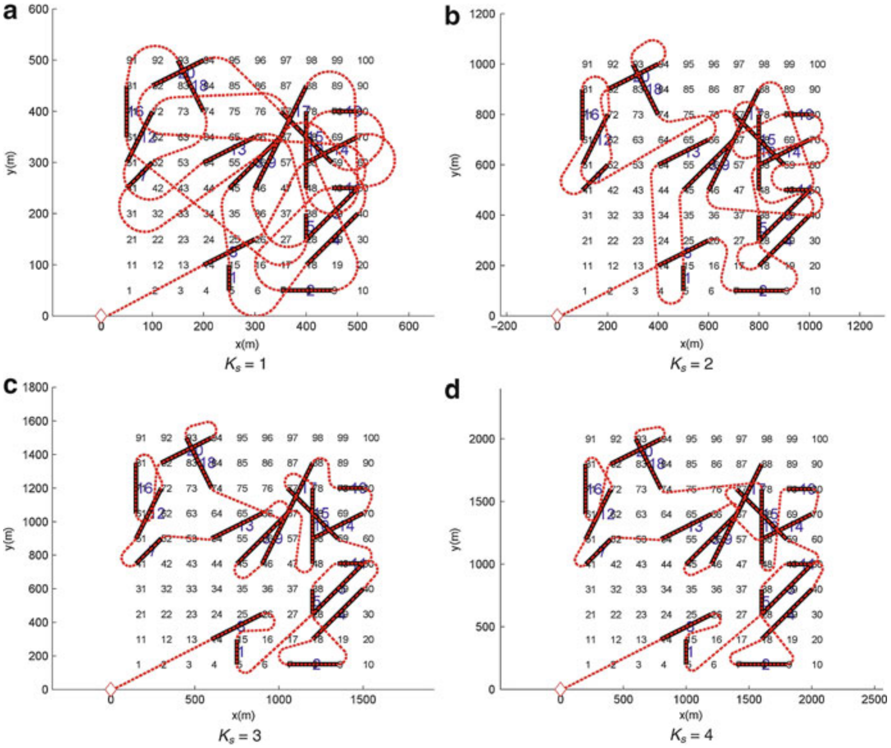


Fig. 62.11 NI-mDCPP road search path with different map scale factors. (**a**) $K_s = 1$. (**b**) $K_s = 2$. (**c**) $K_s = 3$. (**d**) $K_s = 4$

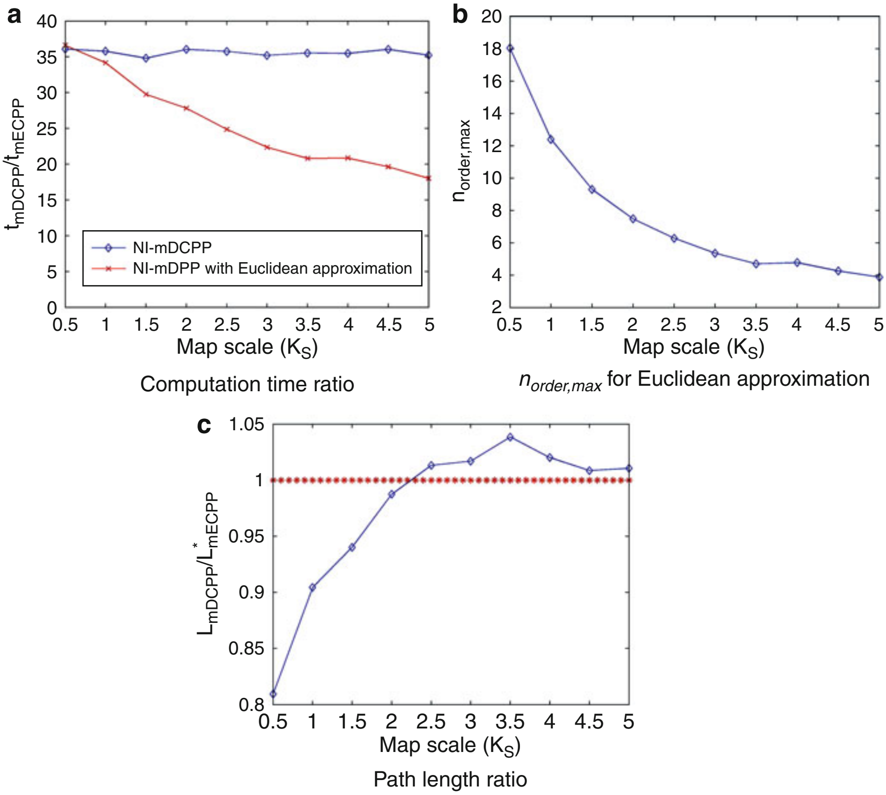
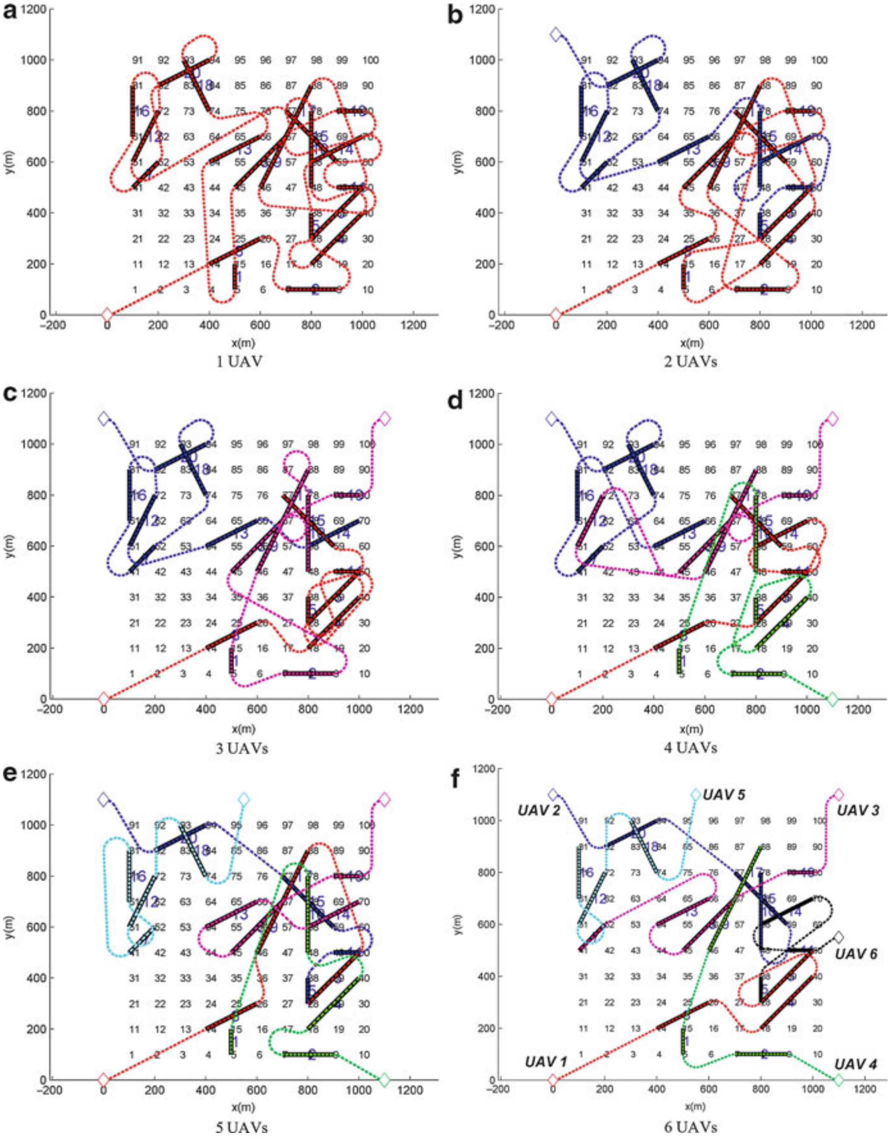


Fig. 62.12 NI-mDCPP results with different map scale factors, average for 50 simulations. (**a**) Computation time ratio. (**b**) $n_{\mathrm{order,max}}$ for Euclidean approximation. (**c**) Path length ratio

straight line (mECPP). Regardless of the map scale, the mDcPP algorithm is around consistently 35 times slower than the mECPP. Meanwhile, the computation time of the mDCPP along with the Euclidean distance order approximation, as explained in Sect. 62.3.4, decreases as the map scale increases as a result of a decrease in the maximum order $n_{\mathrm{order,max}}$ as shown in Fig. 62.12b. Figure 62.12c compares the total path length to cover the entire road map using the mDCPP and the mECPP. For fair comparison, the length of the mECPP (denoted by L^*_{mECPP}) is computed by road search route using the mECPP algorithm but connecting the roads using a Dubins path. This is because although road search route planning is performed using the mECPP, a real trajectory of the UAV connecting the roads should be of Dubins restricted by its maximum curvature. When the minimum turning radius is relatively small compared to the distance between roads, that is, when the map scale is small, the path length of the mDCPP is shorter than that of the mECPP. However, as the map scale gets bigger, the path length ratio gets closer to one (or even greater than one) since the road search order using Dubins paths is almost the same as using a straight line, as one can expect from Fig. 62.11.



1 UAV

2 UAVs

3 UAVs

4 UAVs

5 UAVs

6 UAVs

Fig. 62.13 NI-mDCPP road search path with different number of UAVs ($K_s = 2$). (**a**) 1 UAV. (**b**) 2 UAVs. (**c**) 3 UAVs. (**d**) 4 UAVs. (**e**) 5 UAVs. (**g**) 6 UAVs

Multiple UAVs Case

The second simulation is performed using multiple UAVs with different road map scales, and one of the search route-planning results using the mDCPP is shown in Fig. 62.13. The initial position of each UAV is equally distributed around the road area. Figure 62.14 shows the normalized simulation results for

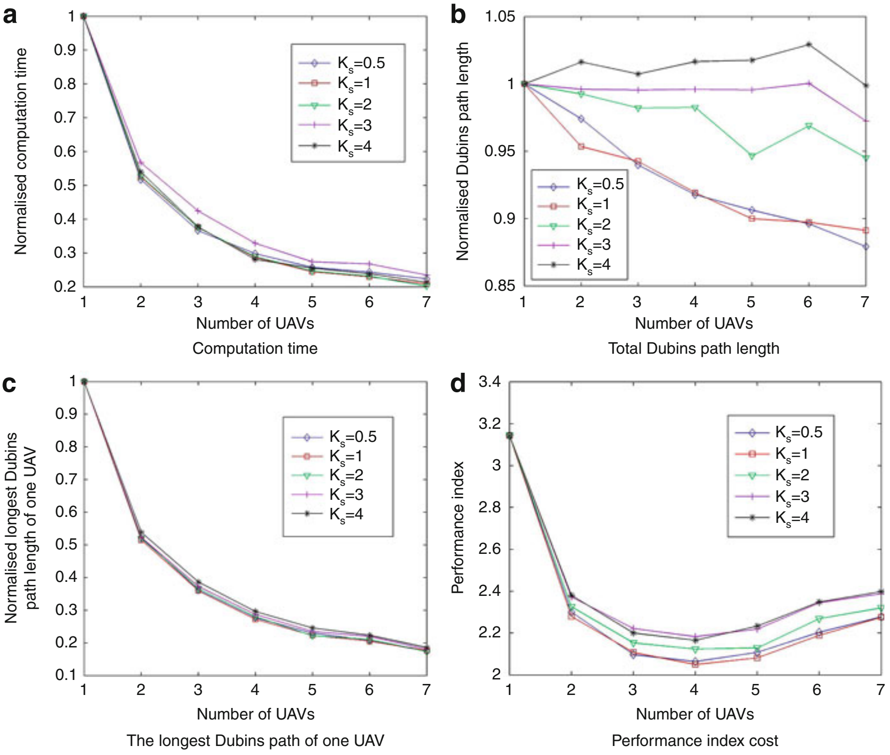


Fig. 62.14 NI-mDCPP results with different number of UAVs ($K_s = 2$), average for 50 simulations. (**a**) Computation time. (**b**) Total Dubins path length. (**c**) The longest Dubins path of one UAV. (**d**) Performance index cost

a single UAV. In particular, the longest path length (Fig. 62.14c) of the UAV is of interest since it is equivalent to the mission completion time of the entire UAV team. The normalized computation time and the longest path length of the UAV decrease as the size of the UAV team is increased, regardless of the map scale as each UAV takes partial charge of the road search mission cooperatively using the auction-based task assignment. The total path length (Fig. 62.14b) is significantly affected by the map scale. When the map scale is small, the total path length decreases in proportion to the number of UAVs. Whereas in a relatively big map environment, the normalized path length remains close to one since each UAV will fly a long distance from the initial position or from one road to another road. Apparently, the simulation results show that the bigger the UAV team size is, the better performance it shows in terms of the computation time and path length. However, using a large number of UAVs on the team require more operational cost, such as fuel and communication resources. Therefore, the performance index to determine the optimal size of the UAV team for the search mission is proposed as shown in (Eq. 62.12), which includes an additional operational cost for each UAV, normalized by the number of UAVs (by maximum seven UAVs in this case), $\bar{n}_v$, and its weighting factor, w_{n_v}:

$$J = w_t\bar{t} + w_l\bar{l} + w_L\bar{L} + w_{n_v}\bar{n_v} \tag{62.12}$$

where w_t, w_l, and w_L represent the weighting factors of computation time, the total path length and the longest path length, of one UAV, respectively. Under the assumption that all of the weighting factors are set to one, the number of UAVs to minimize the performance index J can be selected as four for all map scale factors as shown in Fig. 62.14d.

62.4 Cooperative Mission Planning II: Communication Relay

This section presents the development of trajectory planning for multiple UAVs making use of Dubins path theory in order to maintain and optimize the communication relay between a FF (friendly fleet) performing a main mission and a GCS (ground control station) centrally administrating the whole mission (Kim et al. 2011). To apply Dubins path theory to the optimization of communication between UAVs and with the GCS, various strategic or dynamic constraints have to be considered: mission planning of the FF, the positions of the GCS, any existing nonflying zones, and the limits of UAV dynamics. These constraints affect the decision making of the waypoint sequences of the UAV members. (The final, definitive version of the Section 62.4 of this chapter has been published in Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 225/1, Jan/2011 (http://pig.sagepub.com/content/225/1/12.abstract) by SAGE Publications Ltd, All rights reserved. © 2011, Institution of Mechanical Engineers.)

62.4.1 Waypoint Sequence Decision Making for Communication Relay

The decision making of the UAVs will be based on the waypoint sequence W_F of the FF which is a priory known since the friendly fleet's future movement can be broadcasted to the UAV swarm and the GCS:

$$W_F = [W_{F0}, W_{F1}, W_{F2}, \ldots, W_{Fn}] \tag{62.13}$$

where $W_{Fi} = (x_{Fi}, y_{Fi})^T$ and n is the number of waypoints to be passed by the FF. Awareness of the asset speed V_F makes it possible to get the arrival-time sequence T_F at the waypoint sequence defined in (Eq. 62.13):

$$T_F = [t_{F0}, t_{F1}, t_{F2}, \ldots, t_{Fn}] \tag{62.14}$$

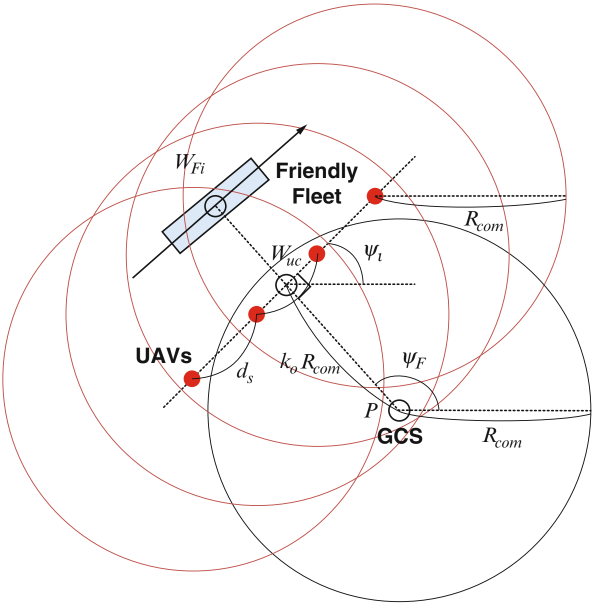


Fig. 62.15 Decision making on spreading out the UAV members: the distance from the GCS to the FF is shorter than twice of the communication range $k_o R_{\text{com}}$

where t_{Fi} is the arrival time at the i-th waypoint. This can be computed using the speed information of the FF which is assumed constant over the mission. Given these waypoint and arrival-time sequences of the FF, the waypoint sequence of N_u UAVs can be determined by considering communication relay between the GCS and the FF.

Let us consider a ratio of the distance from the GCS position $P = (x_p, y_p)^T$ to the FF over the communication range R_{com} as shown in Fig. 62.15 given as

$$k_F = \max_i \frac{|W_{Fi} - P|}{R_{\text{com}}}, \quad i \in \{0, 1, \cdots, n\} \tag{62.15}$$

It is assumed that each UAV has the same communication range as the GCS.

Firstly, consider the case where the distance from the GCS to the FF is shorter than $(1 + k_o)$-times the communication range R_{com}, that is, $k_F \leq (1 + k_o)$ as shown in Fig. 62.15 (k_o is given by (Eq. 62.16)). If the FF's next waypoint and arrival time are given in (Eqs. 62.13) and (62.14), the target center position of UAVs, W_{uc}, is assigned as the intersection between a circle of the communication range adjusted by an overlap coefficient k_o and the line-of-sight from the GCS position to the asset's next waypoint W_{Fi}, where

$$W_{uc} = P + k_o R_{\text{com}}[\cos \psi_F, \sin \psi_F]^T \tag{62.16}$$

and where

$$\psi_F = \tan^{-1} \frac{y_{Fi} - y_p}{x_{Fi} - x_p} \tag{62.17}$$

The distribution of UAVs on the boundary of the communication range aims at maximizing the coverage of communication when the FF goes out of the communication range. Note that $0 < k_o \leq 1$ is adjustable to make the boundaries between the communication circles overlap in order to prevent loss of communication. The choice of the design parameter k_o is made by a trade-off analysis.

A target waypoint of an individual UAV is next determined to distribute all of the swarm members relatively to the target center position W_{uc}. All the UAV members should spread out in a single line, as shown in Fig. 62.15, because the other UAVs enable redundant communication channels in an emergency, although in this case, a single UAV is enough to maintain the communication coverage. The distribution line needs to be perpendicular to the line-of-sight $\overrightarrow{PW_{Fi}}$ in order to maximize the lateral communication coverage. Thus, the azimuth of the distribution line relative to the GCS site, ψ_l, is obtained using the azimuth of the FF relative to the GCS site, ψ_F, computed with (Eq. 62.17) to give

$$\psi_l = \psi_F - \pi/2 \tag{62.18}$$

As a result, the position vector of the j-th UAV is decided by a rotational transformation using the number of UAVs N_u, the angle of the distribution line ψ_l, and the target center position W_{uc}:

$$W_u^j = W_{uc} + \begin{bmatrix} \cos(\psi_l) & -\sin(\psi_l) \\ \sin(\psi_l) & \cos(\psi_l) \end{bmatrix} \begin{pmatrix} (-(N_u - 1)/2 + (j-1))d_s \\ 0 \end{pmatrix} \tag{62.19}$$

where $j \in \{1, 2, \cdots, N_u\}$ and d_s is a separation distance between the UAVs at the target waypoint.

In the general case, consider the case where the distance from the GCS to the FF exceeds $(1 + k_o)$-times the communication range R_{com}, that is, $k_F > (1 + k_o)$ in (Eq. 62.15). To maintain the communication relay, the distribution of the UAVs has to be restructured since a single UAV cannot link the communication between the FF and the GCS. This study adopts a restructuring policy such that some of the UAV members can move to communication-relay positions having the shape of a chain defined by k_F. First, choose the number of UAVs N_m that should move to the new communication-relay positions in order not to lose the communication as

$$N_m = < \frac{k_F}{k_o} - 1 > \tag{62.20}$$

where $< x >$ rounds the value of x to the nearest integer toward minus infinity. Note that the closer that k_F to k_o is, the fewer UAVs are needed, but the smaller the overlapping areas are. For example, when the FF flies to the waypoint defined by $2k_o \leq k_F < 3k_o$ with the help of a four-UAV swarm, N_m becomes 1; hence, only the fourth UAV member of the swarm should move to the new relay waypoint point W_{ur}^4, as shown in Fig. 62.16. In a similar manner, generalize the relay position of W_{ur}^k by

$$W_{ur}^k = W_{Fi} + \begin{bmatrix} \cos(\psi_F) & -\sin(\psi_F) \\ \sin(\psi_F) & \cos(\psi_F) \end{bmatrix} \begin{pmatrix} -(N_u - k + 1)k_o R_{\text{com}} \\ 0 \end{pmatrix} \tag{62.21}$$

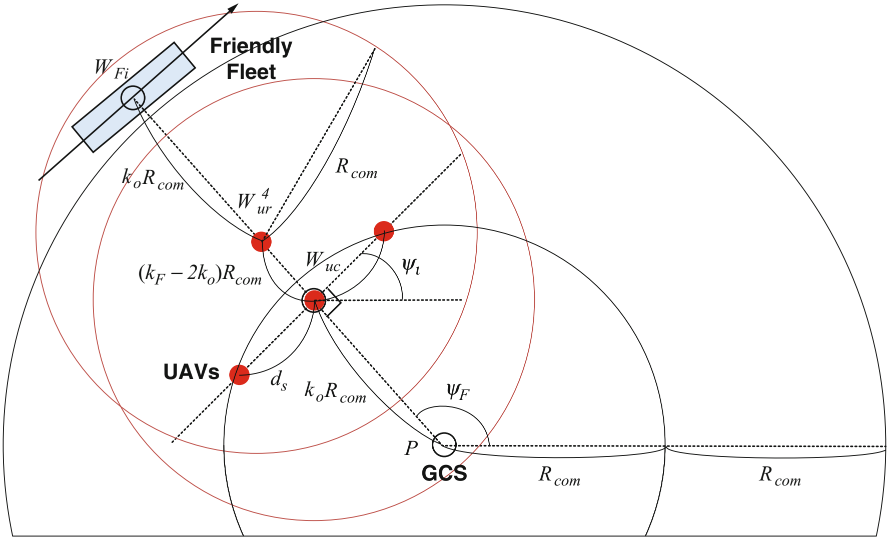


Fig. 62.16 Decision making on spreading out the UAV members: the distance from the GCS to the FF is longer than twice of the communication range $k_o R_{\text{com}}$

where $k \in \{N_u - N_m + 1, N_u - N_m + 2, \cdots, N_u\}$. In short, N_m number of UAVs move to the relay positions W_{ur}^k as defined in (Eq. 62.21), while the other $N_u - N_m$ number of UAVs are distributed on the line as defined in (Eq. 62.19). In this way, the communication between the GCS and the FF can be maintained in virtue of the reconstruction of the UAV formation no matter that the FF maneuvers beyond the communication boundary of the GCS.

Finally, the following waypoint and arrival time sequences of the j-th UAV can be obtained using (Eqs. 62.19), (62.21), and (62.14):

$$W_U^j = [W_{u0}^j, W_{u1}^j, W_{u2}^j, \ldots, W_{un}^j] \tag{62.22}$$

$$T_U^j = [t_{u0}^j, t_{u1}^j, t_{u2}^j, \ldots, t_{un}^j] \tag{62.23}$$

where $t_{ui}^j = t_{ai}$ since the UAV has to arrive at the same time as the assets to protect them effectively.

62.4.2 Nonflying Zone Constraint Against UAV

In a battlefield or an urban area, a NFZ (nonflying zone) could exist as an obstacle dangerous to a swarm of UAVs generated by an enemy, a high building/structure, or an airport space. When a threat is detected, the path planner generates an intermediate waypoint so that the threat is avoided. Consider a flyable trajectory $r(q)$ generated for a given set of poses/waypoints. The schematic of the concept

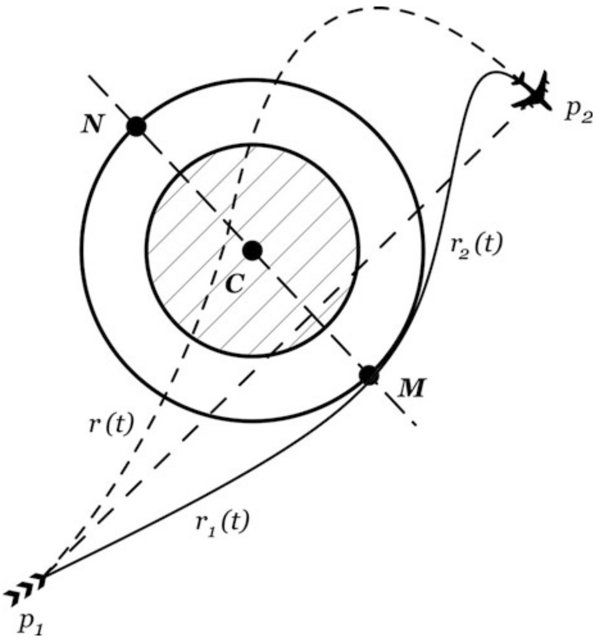


Fig. 62.17 Threat handling by intermediate pose

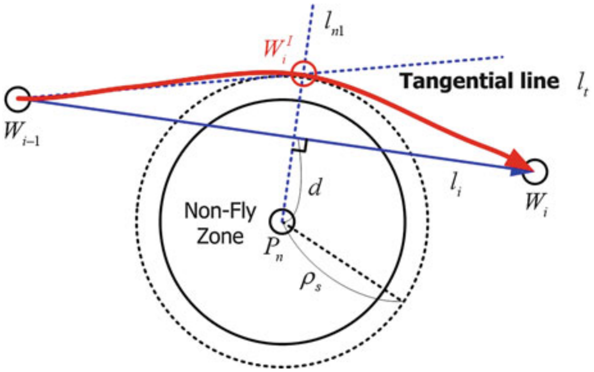


Fig. 62.18 Case 1: The line between the target waypoints crosses the NFZ

is shown in Fig. 62.17. In the figure, the central hatched circle is the threat. The intermediate waypoints are generated by first drawing a line between the current waypoint p_1 and the next waypoint p_2. If a line orthogonal to this is constructed, it will intersect the safety circle at two points N and M. These are then designated as the potential intermediate waypoints. If the center C is left to the line p_1p_2, the intermediate waypoint M is selected on the right-hand side of the threat region and vice versa. Consider two cases relating to the nonflying zone avoidance.

62.4.2.1 Case 1: The Line Between the Target Waypoints Crosses the NFZ

The first case happens when the line between the current waypoint W_{i-1} and the next waypoint W_i cuts a nonflying zone as shown in Fig. 62.18.

Define a vector of positions of target waypoints, the intermediate waypoint to be designed, and the nonflying zone center by

$$W_{i-1} = [x_{i-1}, y_{i-1}]^T \tag{62.24a}$$

$$W_i = [x_i, y_i]^T \tag{62.24b}$$

$$W_i^I = [x_i^I, y_i^I]^T \tag{62.24c}$$

$$P_n = [x_n, y_n]^T \tag{62.24d}$$

The following equation of a line l_i is obtained with a simple geometrical relation between two successive waypoints, W_{i-1}, W_i:

$$y - y_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}(x - x_i) \tag{62.25}$$

To check the violation of the NFZ, the distance from the NFZ center to the line l_i is calculated as

$$d = \frac{|(y_i - y_{i-1})(x_n - x_i) - (x_i - x_{i-1})(y_n - y_i)|}{\sqrt{(y_i - y_{i-1})^2 + (x_i - x_{i-1})^2}} \tag{62.26}$$

If $d < \rho_s$, a intermediate target waypoint W_i^I needs to be defined for the UAV to make a detour around the nonflying zone. For this, the tangential line l_t is required, and its gradient can be obtained using the following second-order polynomial equation which represents the relationship between the tangential line l_t and the safety circle radius ρ_s:

$$am^2 + bm + c = 0 \tag{62.27}$$

where the coefficients are defined as

$$a = (x_n - x_{i-1})^2 - \rho_s^2 \tag{62.28a}$$

$$b = 2(x_n - x_{i-1})(y_{i-1} - y_n) \tag{62.28b}$$

$$c = (y_{i-1} - y_n)^2 - \rho_s^2 \tag{62.28c}$$

This equation has two roots, and the root m_t closest to the gradient of the line l_i is chosen. The resulting equation of the line l_t is given by

$$y - y_{i-1} = m_t(x - x_{i-1}) \tag{62.29}$$

The equation of the line l_{n1} is defined using the property that it is perpendicular to the line l_i and is located at the NFZ center P_n:

$$y - y_n = m_{n1}(x - x_n) \tag{62.30}$$

where $m_{n1} = -\frac{x_i - x_{i-1}}{y_i - y_{i-1}}$. Finally, the intermediate target waypoint W_i^I is defined by the intersecting point of the lines l_t and l_{n1}. Solving (Eqs. 62.29) and (62.30) together gives

$$x_i^I = \frac{m_t x_{i-1} - y_{i-1} - m_{n1} x_n + y_n}{m_t - m_{n1}} \tag{62.31a}$$

$$y_i^I = m_t(x_i^I - x_{i-1}) + y_{i-1} \tag{62.31b}$$

$$x_i^I = \frac{m_t x_{i-1} - y_{i-1} - m_{n1} x_n + y_n}{m_t - m_{n1}} \tag{62.32a}$$

$$y_i^I = m_t(x_i^I - x_{i-1}) + y_{i-1} \tag{62.32b}$$

62.4.2.2 Case 2: The Next Target Waypoint Exists Inside the NFZ

For this case, a replacement target waypoint W_i^S needs to be defined as the next target waypoint exists inside the NFZ as shown in Fig. 62.19. In a similar manner to case 1 in Sect. 62.4.2.1, the line l_t tangential to the NFZ safety circle can be obtained using (Eqs. 62.27)–(62.29). Then, define a position vector of the substitute target waypoint as

$$W_i^S = [x_i^S, y_i^S]^T \tag{62.33}$$

The equation of the line l_{n2} is defined using the property that it is perpendicular to the line l_t and goes through the next waypoint W_i:

$$y - y_i = m_{n2}(x - x_i) \tag{62.34}$$

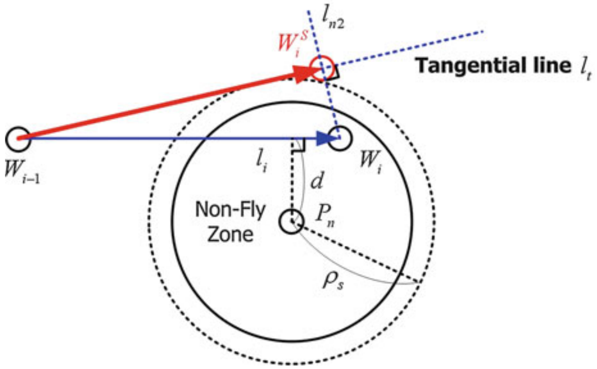


Fig. 62.19 Case 2: The next target waypoint exists inside the NFZ

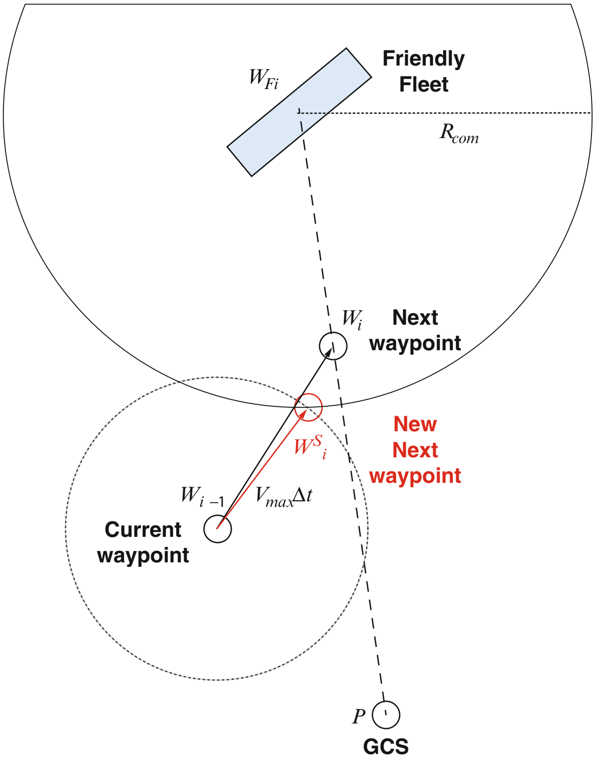


Fig. 62.20 Decision making concept on speed constraints: An intersection position exists (*Left*) and does not exist (*Right*) between the maximum arrival circle of the UAV and the protection line

where $m_{n2} = -1/m_t$. Finally, the replacement target waypoint W_i^S is defined as the intersection point of lines l_t and l_{n2}. Solving (Eqs. 62.29) and (62.34) together gives

$$x_i^S = \frac{m_t x_{i-1} - y_{i-1} - m_{n2} x_i + y_i}{m_t - m_{n2}} \tag{62.35a}$$

$$y_i^S = m_t(x_i^S - x_{i-1}) + y_{i-1} \tag{62.35b}$$

62.4.3 Speed Constraint of UAV

The UAVs have a maximum speed constraint $V_{\max}$; thus, at times they cannot arrive at the next target waypoint W_i in time from the previous waypoint W_{i-1}, as shown in Fig. 62.20. In this case, the next target waypoint has to be replaced by a new waypoint W_i^S. The decision making considers only the case in which there exists an intersection between the maximum arrival circle of the UAV and the communication range of the FF. To check existence of an intersection between the area of the maximum arrival circle and the next waypoint, the distance from the

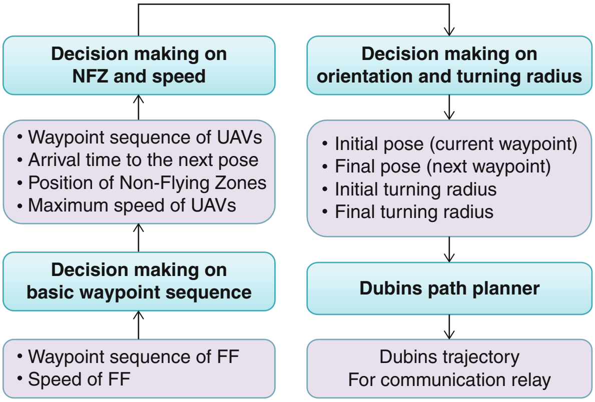

Decision making on NFZ and speed
Decision making on orientation and turning radius
• Waypoint sequence of UAVs
• Arrival time to the next pose
• Position of Non-Flying Zones
• Maximum speed of UAVs
• Initial pose (current waypoint)
• Final pose (next waypoint)
• Initial turning radius
• Final turning radius
Decision making on basic waypoint sequence
Dubins path planner
• Waypoint sequence of FF
• Speed of FF
Dubins trajectory For communication relay

Fig. 62.21 The overall block diagram of the proposed decision-making and path-planning methodology

current waypoint W_i to the next waypoint W_{i+1} is calculated as

$$d = |W_i - W_{i+1}|. \tag{62.36}$$

The radius of the maximum arrival circle is computed as $V_{\max}\Delta t$ where Δt is defined as $t^j_{ui} - t^j_{u(i-1)}$ using (Eq. 62.23). If $d > V_{\max}\Delta t$, an intersection position does not exist between the area of the maximum arrival circle of the UAV and the next waypoint, as shown in Fig. 62.20. In this case, since the UAV has to go inside the FF's communication range in order to maintain the communication relay, the new waypoint W_i should be determined by the intersection between the maximum arrival circle of the UAV and the communication range of the FF.

The starting and final positions have been defined to generate the Dubins path between the target waypoints of the UAVs. To complete the design of the Dubins path, the orientation also need be designed. The orientation of the UAVs between the i-th and $(i+1)$-th waypoints is synchronized with that of the friendly fleet in order to keep the line-of-sight stable from the GCS to the FF as

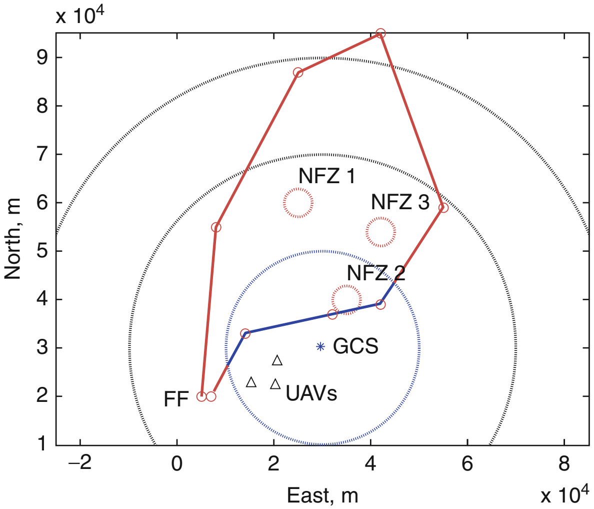
$$\theta_i = \tan^{-1}\frac{y_{Fi} - y_{F(i-1)}}{x_{Fi} - x_{F(i-1)}} \tag{62.37a}$$

$$\theta_{i+1} = \tan^{-1}\frac{y_{F(i+1)} - y_{Fi}}{x_{F(i+1)} - x_{Fi}} \tag{62.37b}$$

Figure 62.21 shows the overall block diagram of the proposed decision-making and path-planning methodology.

62.4.4 Numerical Simulations

Fig. 62.22 An illustration of basic scenario



62.4.4.1 Simulation Conditions

In order to demonstrate the performance and behavior of the path-planning system, numerical simulations are carried out on a baseline scenario as depicted in Fig. 62.22. The flight plan of the friendly fleet is shown as a bold red (outside the GCS communication range) and blue (inside the GCS communication range) line passing through a series of predefined waypoints. The circumstances are such that the flight of the FF must pass outside of the GCS communication range for a significant period. Figure 62.22 shows that the FF flies up to three times the communication ranges. Three nonflying zones (red dotted circles) are also defined around and outside the GCS communication range. The speed and the flight plan of the FF for the whole mission is known to the UAV members in advance ahead the next waypoint. The UAV group consists of three members (defined by (Eq. 62.16)) whose initial positions are illustrated as triangles. The detailed default parameters can be found in Table 62.2.

62.4.4.2 Simulation Results

To effectively compare performance, the proposed algorithm is applied to a two-UAV swarm. Figures 62.23 and 62.24 show that the movements of the UAV members fail to maintain the communication relay for about 1,200 s in the middle of the mission because the distance from the GCS to the UAV members exceeds the communication range R_{com}, which means the GCS loses the communication with the UAV. This implicates that more than two UAVs should be required to maintain the communication relay between the FF and the GCS.

The trajectories of the UAVs avoid entering the NFZ effectively, and at the same time, their speeds do not go over the maximum, as shown in Figs. 62.25 and 62.26.

Table 62.2 Simulation parameters

Variable	Description	Value	Unit
R_{com}	Communication range	20	km
P	Position of GCS	$[30, 30]^T$	km
P_n	Positions of NFZ	$\begin{bmatrix} 25 & 35 & 42 \\ 60 & 40 & 54 \end{bmatrix}$	km
ρ_s	Radius of NFZ	3	km
W_F	Waypoint sequence of FF	$\begin{bmatrix} 5 & 8 & 25 & 42 & 55 & 42 & 32 & 14 & 7 \\ 20 & 55 & 87 & 95 & 59 & 39 & 37 & 33 & 20 \end{bmatrix}$	km
V_F	Speed of FF	50	m/s
d_s	Separation distance	2.5	km
$V_{\max}$	Maximum speed of UAV	50	m/s
$n_{\max}$	Maximum lateral acceleration of UAV	1	g
k_o	Overlap coefficient	0.9	N/A

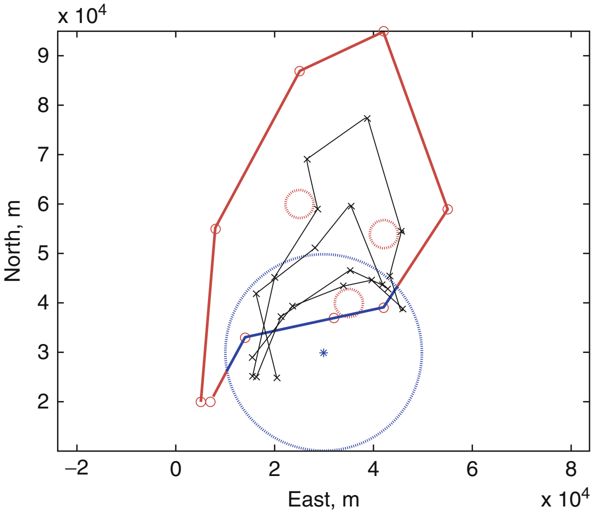


Fig. 62.23 Simulation result on basic scenario with two UAVs: position histories of UAVs (x) and friendly fleet (o) (*star*: GCS, *red dotted circle*: NFZ, *black dotted circle*: GCS coverage)

Also, this figure shows that the communication is successfully maintained. It can be shown that the minimum distance from the GCS to the UAVs, the maximum distance between the UAVs, and the minimum distance from the FF to the UAVs are always kept below the communication range R_{com}. Figure 62.27 shows snapshots of the communication boundaries of the UAV members, depicted as the red circles, where the communication between the GCS and the friendly fleet is successfully relayed by restructuring the UAV positions in a chain shape.

Conclusions

This chapter has developed a framework enabling cooperative mission and path planning of multiple UAVs in the context of the vehicle-routing problem. The vehicle-routing algorithms often approximate paths of UAVs to straight lines

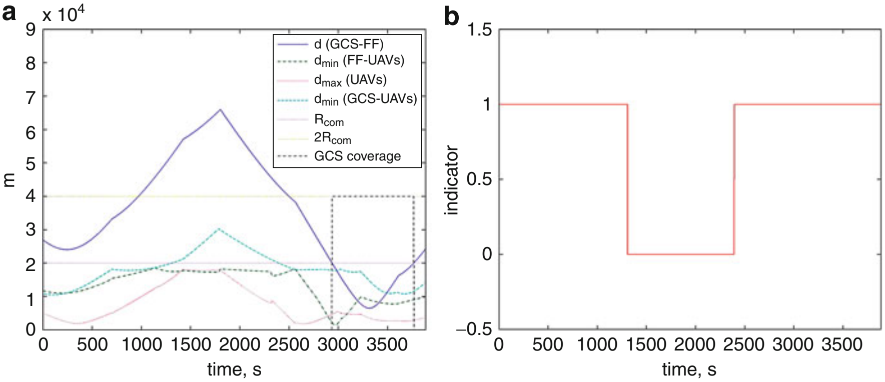
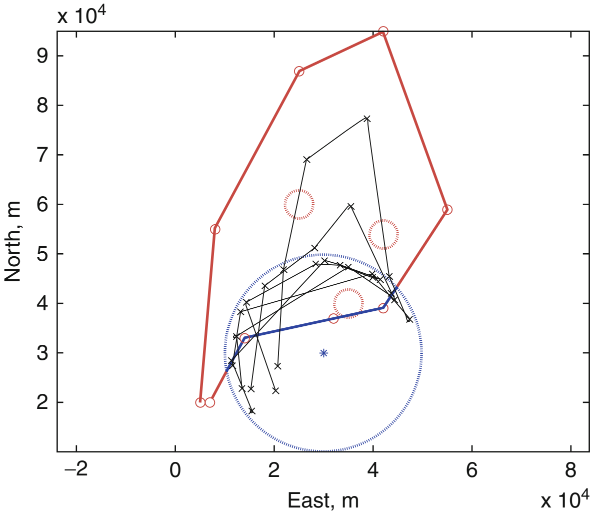


Fig. 62.24 Simulation result on basic scenario with two UAVs: (**a**) relative distance histories among GCS, UAVs, and friendly fleet and (**b**) indication on communication relay (1: success, 0: failure)

Fig. 62.25 Simulation result on basic scenario with speed and NFZ constraints: position histories of UAVs (x) and friendly fleet (o) (*star*: GCS, *red dotted circle*: NFZ, *black dotted circle*: GCS coverage)



to reduce computational load. In order to resolve this issue, this study focused on integration of the differential geometry concepts, especially Dubins paths, into the vehicle route-planning design procedure to consider non-straight path components. Note that physical and operational constraints of the UAV can be taken into account by these non-straight components of paths. In order to validate this approach, cooperative mission and path-planning algorithms for two missions are developed: road-network search and communication relay. In the road-network search algorithm, the conventional Chinese Postman Problem (CPP) algorithm was firstly explained and modified and applied to the general type of road map which includes unconnected roads. Then, MILP optimization

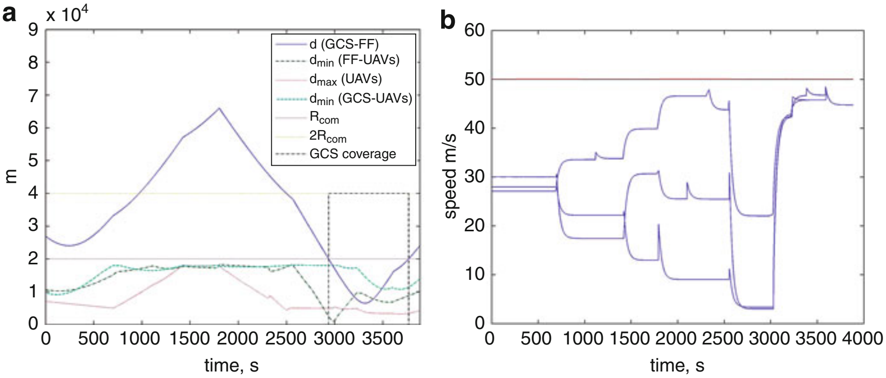


Fig. 62.26 Simulation result on basic scenario with speed and NFZ constraints: (**a**) relative distance histories among GCS, UAVs, and friendly fleet and (**b**) speed histories of UAVs (*red line*: $V_{\max}$)

and the nearest insertion algorithm along with auction negotiation were examined for multiple unmanned aerial vehicles. To realistically accommodate the maneuvering constraints of UAVs, the Dubins path planning was used to solve the modified CPP. The performance of the MILP optimization and the approximation algorithm was then compared for a specific road-network scenario. Particularly, the performance of the approximation algorithm was investigated via a Monte Carlo simulation framework by analyzing the effects of different map sizes, path-planning methods, and the number of UAVs. Based on these results, the efficient UAV team size and path-planning method were suggested for the road search route planning and hence showed that this framework can be applied to a variety of autonomous search missions in the initial phase of mission planning. In the communication-relay problem, a design methodology on vehicle route planning of multiple UAVs to maintain communication between the ground control station and the friendly fleet was proposed. Dubins theory was applied to get flyable paths, and a decision-making algorithm was developed to satisfy the constraints on the UAV speed and the avoidance of nonflying zones, as well as maintaining communication. The performance of the proposed algorithms is examined via numerical simulations.

References

A. Ahmadzadeh, G. Buchman, P. Cheng, A. Jadbabaie, J. Keller, V. Kumar, G. Pappas, Cooperative control of UAVs for search and coverage, in *Conference on Unmanned Systems* (SPIE, Bellingham, 2006)

D. Ahr, Contributions to multiple postmen problems. Ph.D. thesis, Heidelberg University, 2004

B. Alspach, Searching and sweeping graphs: a brief survey. Matematiche (Catania) **59**, 5–37 (2006)

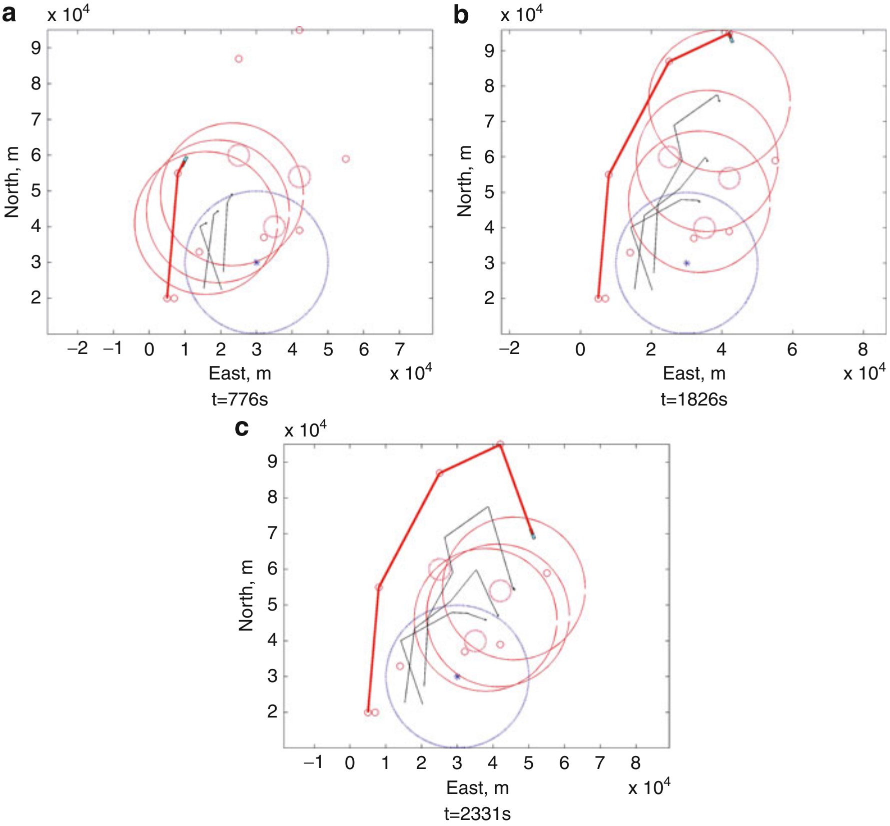


Fig. 62.27 Simulation result on basic scenario with speed and NFZ constraints: snapshots of communication ranges of UAVs. (**a**) t = 776 s. (**b**) t = 1,826 s. (**c**) t = 2,331 s

T. Bektas, The multiple traveling salesman problem: an overview of formulations and solution procedures. Int. J. Manag. Sci. **34**(3), 209–219 (2006)

J. Bellingham, M. Tillerson, A. Richards, J. How, *Multi-Task Allocation and Path Planning for Cooperating UAVs, Cooperative Control: Models, Applications and Algorithms* (Kluwer, Dordrecht, 2003)

C. Cerasoli, N.J. Eatontown, An analysis of unmanned airborne vehicle relay coverage in urban environments, in *Proceedings of MILCOM* (IEEE, Piscataway, 2007)

P. Chandler, M. Pachter, D. Swaroop, J. Hewlett, S. Rasmussen, C. Schumacher, K. Nygard, Complexity in UAV cooperation control, in *American Control Conference*, Anchorage (American Control Conference, Evanston, 2002)

L.E. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. Am. J. Math. **79**(3), 497–516 (1957)

K. Easton, J. Burdick, A coverage algorithm for multi-robot boundary inspection, in *IEEE International Conference on Robotics and Automation* (IEEE, Piscataway, 2005)

A. Gibbons, *Algorithmic Graph Theory* (Cambridge University Press, Cambridge/New York, 1999)

J. Gross, J. Yellen, *Handbook of Graph Theory* (CRC, Boca Raton, 2003)

M. Hifi, M. Michrafy, A. Sbihi, A reactive local search-based algorithm for the multiple-choice multi-dimensional knapsack problem. Comput. Optim. Appl. **33**, 271–285 (2006)

S. Kim, P. Silson, A. Tsourdos, M. Shanmugavel, Dubins path planning of multiple unmanned airborne vehicles for communication relay. Proc. IMechE G **225**, 12–25 (2011)

I. Maza, A. Ollero, Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms, in *Distributed Autonomous Robotic Systems 6*, vol. 5 (Springer, Tokyo/New York, 2007), pp. 221–230

N. Nigam, I. Kroo, Persistent surveillance using multiple unmanned air vehicles, in *Aerospace Conference, 2008 IEEE*, Big Sky (IEEE, Piscataway, 2008)

H. Oh, S. Kim, A. Tsourdos, B. White, Cooperative road-network search planning of multiple UAVs using dubins paths, in *AIAA Guidance, Navigation and Control Conference*, Portland (AIAA, Reston, 2011a)

H. Oh, H. Shin, A. Tsourdos, B. White, P. Silson, Coordinated road network search for multiple UAVs using dubins path, in *1st CEAS Specialist Conference on Guidance, Navigation and Control*, Munich (Springer, Berlin/Heidelberg, 2011b)

H. Oh, S. Kim, A. Tsourdos, B. White, Coordinated road-network search route planning by a team of UAVs. Int. J. Syst. Sci., (2012) In press

T. Parsons, Pursuit-evasion in a graph, in *Theory and Applications of Graphs* (Springer, Berlin, 1976)

N. Perrier, A. Langevin, J. Campbell, A survey of models and algorithms for winter road maintenance. Part IV: Vehicle Routing and Fleet Sizing for Plowing and Snow Disposal. Comput. Oper. Res. **34**, 258–294 (2007)

M.F.J. Pinkney, D. Hampel, S. DiPierro, Unmanned aerial vehicle (uav) communications relay, in *IEEE Military Communications Conference, 1996. MILCOM'96, Conference Proceedings* (IEEE, Piscataway, 1996)

D. Rosenkrantz, R. Stearns, P. Lewis, An analysis of several heuristics for the traveling salesman problem. Fundam. Probl. Comput. **1**, 45–69 (2009)

J. Royset, H. Sato, Route optimization for multiple searchers. Nav. Res. Logist. **57**, 701–717 (2010)

J. Ryan, T. Bailey, J. Moore, W. Carlton, Reactive tabu search in unmanned aerial reconnaissance simulations. in *30th Conference on Winter Simulation*, Washington, DC, 1998

T. Ralphs, L. Ladanyi, M. Guzelsoy, A. Mahajan, SYMPHONY 5.2.4 (2010), http://projects.coin-or.org/SYMPHONY

T. Samad, J. Bay, D. Godbole, Network-centric systems for military operations in urban terrain: the role of UAVs. Proc. IEEE **95**(1), 92–107 (2007)

M. Shanmugavel, Path planning of multiple autonomous vehicles. Ph.D. thesis, Cranfield University, 2007

M. Shanmugavel, A. Tsourdos, B.A. White, R. Zbikowski, Differential geometric path planning of multiple UAVs. J. Dyn. Syst. Meas. Control **129**, 620–632 (2007)

A. Tsourdos, B. White, M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles* (Wiley, Chichester, 2010)