Proceedings of the 36th Chinese Control Conference
July 26-28, 2017, Dalian, China

Digraph-based Anti-Communication-Destroying Topology Design for Multi-UAV Formation

Hao Chen, Xiangke Wang, Lincheng Shen, Jing Chen
College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, 410073, China
E-mail: {chenhao09,xkwang,lcshen}@nudt.edu.cn
chenjing001@vip.sina.com

Abstract: This paper addresses the problem of robust topology design of the multi-UAV formation. We use a digraph to model the topology of the multi-UAV formation, and propose the definition of *anti-k-communication-destroying* topology, meaning the system can still performs normally even when any arbitrary k communicating links are destroyed. By exploring the property of this kind of topology based on graph theory, we propose the algorithm *Uniform-Cost Forest Search, UCFS*, which is an extension of the classical search strategy *uniform-cost search*. The proposed algorithm would establish the *anti-k-communication-destroying* topology for the multi-UAV formation, with $k + 1$ minimal-cost edge-independent paths found for each UAV. Proof as well as examples validating the algorithm are provided in the paper. Finally, potential research directions are discussed.

Key Words: UAV Formation, Multi-UAV Systems, Digraph, Communication Network, Anti-Communication-Destroying Topology

1 Introduction

Formation flight is the main mode for multiple unmanned aerial vehicles (UAVs) to complete the task, thus has attracted widespread attentions around the world in the last 15 years [1–3]. Compared with one single UAV, the cooperation of multiple UAVs has many distinctive advantages and promising application prospect, for its distributed sensors, parallel actuators, better fault tolerance and robustness [4, 5].

Similar to other numerous networks, e.g. transport networks, road networks, electrical networks, telecommunication systems or networks of servers, a network of UAVs can be modeled by graphs (see [6] as a survey). Many attempts have been made to determine how well such a network is "connected", or how much effort is required to break down communication in the system between some vertices (see [7–12]). Clearly, it is desirable that a network stays connected as long as possible in case of faults should arise.

In general, multiple UAVs are connected by communication links to perform a task cooperatively. However, due to the particularity of environment and task, UAV formation may be influenced by many factors, including the enemy electromagnetic interference, weather, and UAVs' own problems in the process of flight, which can lead to damage of some communicating links. It is known that the loss of communicating links will deteriorate the controllability of the whole formation, resulting in the failure of the mission [13]. Besides, as indicated in [14, 15], the information flow among UAVs are usually severely restricted, due to security reasons or bandwidth limitations. Consequently, no UAVs might be able to comunicate with the entire formation. Therefore, how to guarantee the controllability of the system when some communicating links are destroyed is an urgent problem. Generally, there are two main approaches. The first one is to develop strategies that can yield robust performance in the presence of temporary loss of communicating links [16]. Another method many researchers have tried is on reconfiguration or adaptive configuration when such a failure arises, and this is also a typical method for the system to bear the loss of platforms [1, 2]. This paper, however, provides a new solution to fix this problem, namely, when UAV formation is formed, certain communicating topology with anti-destroying ability is designed at the same time, making the formation can bear the loss of communicating links to some extent. In this paper, the digraph is used to model the multi-UAV formation and its communicating topology. The failure of parts of the communicating links is corresponding to the loss of edges in the digraph. Then based on the graph theory, the anti-communication-destroying ability is studied, and correspondingly its generating algorithm is proposed. Our research can improve the topology robustness of UAVs when performing missions in the complex environment, thus providing some technical supports and theoretical foundations for the multi-UAV system in practical applications.

The paper is organized as follows: in Section 2, we briefly review the related preliminaries in graph theory and present our problem statement, defining the *anti-communication-destroying* topology and exploring the property of the topology; in Section 3, we present the algorithm to generate the *anti-communication-destroying* topology based on the properties illustrated in Section 2; then in Section 4, we use an example to validate the algorithm; and finally we give the conclusion and future research directions in Section 5.

2 Preliminaries and Problem

A *graph* is a basic tool to represent a set of objects where some pairs of objects are connected by links. Mathematically, a *graph* $G(V, E)$ is composed of a finite nonempty set of vertices V and a set $E \in V \times V$ of edges. Typically, there are two kinds of graphs. If the edges are ordered, then the graph is called *directed graph* or *digraph* for short; otherwise, it is an *undirected graph*.

A digraph $G(V, E)$ is usually used to model the communicating topology of the multi-UAV formation, where vertices V are independent UAV platforms and edges E are weighted communicating links. We consider finite digraphs without loops and multiple edges in this study, which means each

This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61403406, 61403411.

UAV is not communicating with itself, Besides, we assume there are at most two communicating links between UAV_i and UAV_j, one from UAV_i to UAV_j, and the other one from UAV_j to UAV_i.

We define the order of G by $n = n(G) = |V(G)|$. If G is a digraph, then the vertex indegree $d^-(v)$ of a vertex v is the number of edges ending at v, and the vertex outdegree $d^+(v)$ of a vertex v is the number of edges starting from v. The in-neighborhood $N^-(v)$ of vertex v is defined as $N^-(v) := \{x \in V(G)|x \to v\}$, meaning there exists an edge starting from each element in $N^-(v)$ to the vertex v. Correspondingly, the out-neighborhood $N^+(v)$ of vertex v is defined as $N^+(v) := \{x \in V(G)|v \to x\}$, meaning there exists an edge starting from the vertex v to each element in $N^+(v)$. The minimum and maximum indegree are denoted by $\delta^-(G)$ and $\Delta^-(G)$, with the minimum and maximum outdegree denoted by $\delta^+(G)$ and $\Delta^+(G)$. Obviously, in digraphs, the number of edges $|E|$ equals the sum of the indegree or outdegree of all vetices, i.e.

$$|E| = \sum_{v \in V} d^-(v) = \sum_{v \in V} d^+(v). \tag{1}$$

In our research, the edges are weighted, representing the communicating cost of the graph. Here a main difference between digraph and undirected graph is that in the undirected graph, the cost from the vertex u to v equals that from v to u, while in the digraph, this property is not necessary. If we use a matrix $M_G = [m_{ij}]$ to represent the cost of the edges, where m_{ij} is the cost on the edge from the i-th node to j-th node. Then M is a symmetrical matrix if G is an undirected graph, while A_G is not symmetrical if G is a digraph. We assume there is no cost for the UAV communicating with itself, then mathematically, $m_{ii} = 0$.

Two classical measures that indicate how reliable a graph or a digraph G is are the *edge-connectivity* $\lambda(D)$ and the *vertex-connectivity* $\kappa(G)$ of G. Before we proceed to discuss the cases for digraphs, we first give the definitions for the undirected graphs.

Definition 1 (k-edge-connected and k-vertex-connected) *For an undirected graph G, it is called* k-edge-connected *if there does not exist a set of $k-1$ edges whose removal disconnects the graph; while it is called* k-(vertex)-connected *if there does not exist a set of $k-1$ vertices whose removal disconnects the graph.*

Here we give a simple example to illustrate the concept of *k-edge-connected*. Consider two different undirected graphs as shown in Fig. 1(a) and 1(b), both with 5 vertices and 5 edges. Clearly, for the graph in Fig. 1(a), if any one edge is losing, the graph is still connected. In contrast, for the graph in Fig. 1(b), such property does not hold any longer. If the edge between A-E or D-E is losing, the connectivity is not preserved clearly. Therefore, the graph in Fig. 1(a) is *2-edge-connected*, while the graph in Fig. 1(b) is not.

In terms of the multi-UAV formation, the topology is modeled by digraphs. Usually, some UAVs are chosen as leaders, while other UAVs act as followers. A follower UAV can also be the leader of another UAV, thus established a network. The leader-follower topology not only has the advantage to be easily understood and implemented, but also consistently manifested in biological swarms, e.g. pigeon flocks [17]. With such a topology, human operators only need to send commands to the global leaders while the other UAVs follow these global leaders. Following the definition of *k-edge-connected* of undirected graphs, we propose the definition of *anti-k-communication-destroying* for the multi-UAV formation modeled by digraphs.

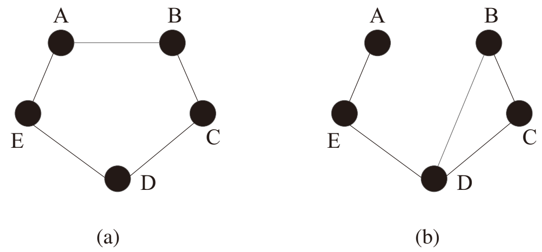


Fig. 1: Two examples of five-agent system with 5 communicating links, modeled with undirected graphs.

Definition 2 (anti-k-communication-destroying) *For the multi-UAV formation modeled by a digraph G, its topology is called* anti-k-communication-destroying *if there does not exist a set of k edges whose removal leads to no communicating path exists from the global leaders to some of the UAVs.*

We use another example to illustrate this concept. In Fig. 2, there is only one global leader *A*, with the other four UAVs as followers. For each of the follower UAV, there exists at least two distinct paths from the leader to itself. For example, the UAV B, the first path is directly from A to B, with the second one A→C→E→D→B. Therefore, the commands can still be sent to each UAV when any one communicating link destroys. Thus the topology is said to be anti-1-communication-destroying. Besides, we can found that there are eight communicating links among the UAVs in Fig. 2, more than that in Fig. 1(a), but fewer than two times of the edges in Fig. 1(a), since no edges needed pointing to the leader *A*. The property of the topology, including the minimal number of edges required, will be explored further in the following work.

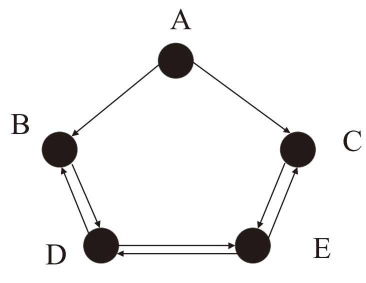


Fig. 2: Anti-1-communication-destroying topology for five UAVs with one global leader.

As the UAVs are represented as the vertices, following the definition of *k-vertex-connected*, we can propose the definition of *anti-k-agent-destroying* topology in the same way. However, we would not discuss its property in this paper, but only focus on the *anti-k-communication-destroying* topology and its generating algorithm.

For undirected graph, there is a well-known basic inequality relationship between the vertex-connectivity, edge-connectivity and minimum degree of a graph G due to Whitney in 1932 [18].

Lemma 1 ([18]) *For any undirected graph G, let $\delta(G)$, $\lambda(G)$ and $\kappa(G)$ be the minimum degree, the vertex-connectivity and the edge-connectivity, respectively. Then*

$$\kappa(G) \leq \lambda(G) \leq \delta(G). \tag{2}$$

Now we extend Lemma 1 to the topology of multi-UAV formation represented by digraphs. Denote L as the set of vertices in G representing global leaders, and $\delta^{-}(G-L)$ represents the minimum indegree of the vertices except those global leaders in G. We have the following theorem indicating the relationship between *anti-communication-destroying* and $\delta^{-}(G-L)$.

Theorem 1 *Suppose the topology of multi-UAV formation is anti-k-communication-destroying, then*

$$k \leq \delta^{-}(G-L) - 1. \tag{3}$$

Theorem 1 can be easily concluded following the Lemma 1 and the definition of *anti-communication-destroying*.

Thus, in order to construct reliable and fault-tolerant networks, as well as to minimize the communicating links to each UAV, a necessary condition satisfying $k = \delta^{-}(G-L) - 1$ would be of great interest. Besides, in terms of the number of edges in the digraph with *anti-k-communication-destroying*, we have the following theorem indicating its minimum.

Theorem 2 *Suppose a multi-UAV formation consisting of n UAVs, with m of them as global leaders, then the minimum number of communicating links for* anti-k-communication-destroying *topology is*

$$|E|_{min} = (n-m)(k+1). \tag{4}$$

Proof Since there can be no communicating links pointing to the leaders, then according to Theorem 1, Equation (1) can be rewritten as

$$|E| = \sum_{v \in V} d^{-}(v) = \sum_{v \in V-L} d^{-}(v) \leq (n-m)(k+1). \tag{5}$$

Therefore, Theorem 2 holds. ∎

Now we present another important property, which laid the foundation of our algorithm. The existing related theory is Menger's theorem for undirected graphs.

Lemma 2 (The edge-connectivity of Menger's theorem) *Let G be a finite undirected graph, u and v any two distinct vertices. Then the edge-connectivity of the graph is equal to the minimum edge-independent paths from u to v.*

And in terms of the anti-communication-destroying topology for multi-UAV formation, we have the following claim.

Theorem 3 *Let G be the digraph representing the topology of multi-UAV formation, and L be the set of vertices in G representing global leaders. Then if G is* anti-k-communication-destroying, *then for each vertex $u \in V - L$, there are at least $k+1$ edge-independent paths from the vertices in L to u.*

According to Theorem 3, the problem to establish anti-k-communication-destroying topology is equal to find $k+1$ edge-independent paths from the leaders to each of the other UAVs.

Theorem 2 and Theorem 3 are important guidelines in our algorithm to design the anti-communication-destroying topology, which we will show in next section.

3 Generating Algorithm

Firstly, we present our algorithm for the multi-UAV formation with only one global leader. However, it can be shown later, the algorithm can be easily adapted to more leaders. Our algorithm is a typical search-based one, named as *Uniform-Cost Forest Search, UCFS*. It is an extension of a classical search strategy, *uniform-cost search* [19], which is known as *Dijkstra's Algorithm*.

Uniform-cost search is used to search for goal by establishing a search tree from the root. The vertices of the tree correspond to the states in the state space of the problem, and the branches correspond to the links between the states. By expanding the current state, a new set of states is generated, then the branches from the parent vertex representing current state, leading to new child vertices are added to the tree. Search algorithms all share this structure only vary according to how they choose the next vertex to be expanded. The strategy of uniform-cost search always expands the vertex with the lowest path cost. The algorithm terminates when the goal state is selected for expansion.

Uniform-cost search can find the optimal path to the goal, and it is appealing when there is no estimate on the cost to the goal, so that the heuristic search methods, like A* cannot be used. In our problem, we aim to find $k+1$ edge-independent paths with minimal cost from the global leader to each follower UAV. This problem is slightly different from the problems suitable for uniform-cost search. Firstly, it is not a single *goal* in our setting, but a team of UAVs for the global leader trying to reach. Secondly, we aim to find $k+1$ edge-independent paths instead of only one.

Algorithm 1 is our proposed UCFS algorithm. The input of the algorithm is a cost matrix M, where $M(i,j)$ indicating the non-negative communicating cost between UAV_i and UAV_j. During the process of the algorithm, we maintain a digraph $G(V,E)$ whose vertices representing the UAV platforms, and the edges representing the communicating links between every two UAVs. The digraph G is initialized with all the vertices but no edges. The UCFS algorithm extends uniform-cost search in a straghtforward way: instead of establishing one tree, it establishes a forest containing $n-1$ trees. The data structure of the vertices on each tree are stored as $v = (id, parent, cost)$, where id represents the corresponding UAV, $parent$ is the parent's ID of the vertex in the tree, and $cost$ is the path cost in the corresponding search tree from the global leader to the current vertex. We also maintain a set S containing $n-1$ elements, storing the unexpanded vertices with the minimum path cost on each tree.

At the initial stage, all the trees are initialized with the root only (in Line 1), which are the vertices representing the UAVs except the global leader, and the costs of the roots are initialized as the communicating cost from the leader UAV to each follower. The trees would expand from the roots and try to reach all the UAVs except the global leaders. For each tree, the unexpanded vertex with the lowest path cost is stored in S, and would be compared with those in the other trees. After the comparison at each step, the tree $tree_i$ containing the vertex v_i with the lowest cost in S would be chosen (in Line 4), and the vertex v_i would be expanded in the tree, with its child vertices added to the tree (in Line 5). We would also add an edge from the parent of the vertex $v_i.parent$ to itself v_i in the digraph G (in Line 6). After that, v_i will be marked as *expanded* in $tree_i$ in Line 7. Here we should note that the mark is in $tree_i$ **only**, and not in other trees. Then we will find a new unexpanded vertex in $tree_i$, whose path cost is the minimum (in Line 8).

An important rule should be kept in mind during the process of expansion is that if an edge from the parent $v_i.parent$ to the vertex v_i itself already exists in G, the vertex v_i should not be selected for expansion. That is why we will do some updates after the expansion at each step, both the tree itself and the other influenced trees need to update the vertex for expansion (in Line 9-19). The influenced trees can be categorized into two types. The first one is that the newly expanded vertex v_i in this tree has the parent whose link to the vertex v_i has been newly added to the digraph G (in Line 10, and the first *if* case in Line 15). The other category is that the vertex v_i is also the unexpanded vertex with lowest cost in the tree, waiting for expansion, however the its indegree $d^-_G(v_i)$ has reached $k+1$ (the *or* case in Line 15).

There are three important functions in Algorithm 1, i.e. $Expansion$, $ForExpansion$, and $UpdateVertex$.

Function $Expansion$ is used to expand the corresponding tree $tree_i$ after comparing the vertices in S, and the vertex v_i is finally selected for expansion. In this function, we would try to add links from v_i to all the other unexpanded vertices in $tree_i$. There are two principles to be followed when we try to add a link from v_i to another unexpanded vertex v_j. Firstly, the edge from v_i pointing to v_j have not existed in the digraph G. Secondly, the indegree of v_j in digraph G should be less than $k+1$. Only when these two principles are both satisfied, can the link from v_i to the unexpanded vertex v_j be added. The function of $Expansion$ for the main UCFS algorithm is shown in Algorithm 2. It is the same as in the other search algorithms.

Function $ForExpansion$ would try to find the unexpanded vertex with the minimal path cost in the tree, this can be done by traversing all the unexpanded vertices in the tree.

Function $UpdateVertex$ would be used to update the parent and the path cost of the vertex in the tree, shown in Algorithm 3. This function is nearly the opposite of the function $Expansion$: instead of trying to add edges from the vertex v_i to the *unexpanded* vertices, we would add edges from the *expanded* vertices to v_i, so to find a new parent of v_i in order that the path cost to v_i in the tree is the minimum. The only principle for the parent is that there exists no edge from the

Algorithm 1 UCFS for Anti-k-Communication-Destroying Multi-UAV Formation with One Global Leader

Input: a cost matrix M, the ID of n UAVs;
Output: anti-k-communication-destroying digraph $G(V, E)$.
Initialize: a digraph $G(V, E)$ with n vertices and none edges, a forest consisting of $n-1$ empty trees, an empty set S.

1: $tree_i = (i, 0, M(leader, i)),\ \ i = 1 \dots n, i \neq leader.$
2: $S = \{(i, 0, M(leader, i))\},\ \ i = 1 \dots n, i \neq leader.$
3: **while** $|E| < (n-1)(k+1)$ **do**
4: find the vertex $v_i \in S$ such that $v_i.cost$ is a minimum in S;
5: $tree_i = Expansion(tree_i, v_i, M)$;
6: $E = E \cup \{(v_i.parent, v_i)\}$;
7: mark v_i as *expanded* in $tree_i$;
8: $v_i = ForExpansion(tree_i)$, and $S(i) \leftarrow v_i$;
9: **for all** $tree_j,\ \ j \neq i,\ j = 1 \dots n, j \neq leader$ **do**
10: **if** $v_i.id == v_p.id$ where $v_p \in tree_j$, and $v_i.parent == v_p.parent$ **then**
11: $v_p = UpdateVertex(tree_j, v_p, M)$.
12: **end if**
13: **end for**
14: **for all** $v_j.id == v_i.id$ where $v_j \in S,\ j \neq i$ **do**
15: **if** $v_j.parent == v_i.parent$ **or** $d^-_G(v_i) == k+1$ **then**
16: $v_j = UpdateVertex(tree_j, v_j, M)$;
17: $v_j = For_{E}xpansion(tree_j)$, and $S(j) \leftarrow v_j$;
18: **end if**
19: **end for**
20: **end while**

Algorithm 2 The $Expansion$ Function for UCFS Algorithm.

Input: vertex v_i with its corresponding tree $tree_i$, cost matrix M;
Output: the updated tree $tree_i$.

1: **for all** unexpanded $v_j \in tree_i$ **do**
2: **if** $(v_i, v_j) \notin E$ and $d^-_G(v_j) < k+1$ and $v_j.cost < v_i.cost + M(i, j)$ **then**
3: $v_j.parent = v_i.parent$, $v_j.cost = v_i.cost + M(i, j)$
4: **end if**
5: **end for**

parent to v_i in the digraph G.

Thus we have finished the algorithm design, and we have the following claim.

Theorem 4 *The proposed UCFS algorithm would*

(1) establish the anti-k-communication-destroying topology for the multi-UAV formation;

(2) for each UAV, $k+1$ *edge-independent paths with the minimal communicating cost are found.*

Proof Firstly, we will show the UCFS algorithm would establish the anti-k-communication-destroying topology.

According to our strategy, when there exists an edge $(v_i, v_j) \in E$, then the vertex v_j would not be added as a child of v_i in any tree again. And since our algorithm starts from $n-1$ different vertices, and terminates when the indegree of all the vertices except the global leader equals to $k+1$, so for each vertex in the digraph G, we would find $k+1$ edge-independent paths from the leader to itself. Then according to Theorem 3, the topology established finally is anti-k-communication-destroying.

Algorithm 3 The $UpdateVertex$ Function for UCFS Algorithm.

Input: vertex v_i with its corresponding tree $tree_i$, cost matrix M;
Output: vertex v_i with its $parent$ and $cost$ updated.
Initialize: an empty set S.

for all expanded vertex $v_j \in S$ **do**
  if $(v_j, v_i) \notin E$ **then**
    $S = S \cup \{v_j\}$.
  end if
end for
find $v_j \in S$, such that $v_j.cost + M(j,i)$ is the minimum.
$v_i.parent = v_j$,
$v_i.cost = v_j.cost + +M(j,i)$.

And in terms of the second claim, it can be easily proven in the same way as to prove the optimality of uniform-cost search algorithm. ■

The proposed UCFS algorithm can be easily adapted for the system with multiple global leaders. Suppose there are m global leaders, while the remaining $n-m$ UAVs are followers, then we need to maintain $m(n-m)$ trees in the forest, with the root of each tree initialized by every *leader-follower* pair. And the exit condition for the algorithm would be the number of edges reaching $(n-m)(k+1)$ in Line 3, instead of $(n-1)(k+1)$.

4 Experiment Results

In this section, we briefly present the result of an experiment with an eight-UAV system. The cost matrix M is randomly generated as

$$M = \begin{pmatrix} 0 & 736 & 260 & 708 & 637 & 498 & 996 & 668 \\ 259 & 0 & 424 & 522 & 370 & 112 & 399 & 181 \\ 697 & 359 & 0 & 921 & 221 & 907 & 368 & 173 \\ 398 & 473 & 570 & 0 & 291 & 277 & 156 & 800 \\ 909 & 518 & 402 & 771 & 0 & 184 & 368 & 915 \\ 206 & 788 & 258 & 763 & 164 & 0 & 142 & 580 \\ 990 & 836 & 288 & 606 & 318 & 510 & 0 & 198 \\ 586 & 190 & 915 & 266 & 148 & 192 & 785 & 0 \end{pmatrix}.$$

For the system with only one global leader UAV_1, the established anti-1-communication-destroying and anti-2-communication-destroying topology is shown in Fig. 3, where the blue circles represent UAVs and the arrows indicate the information flows among the systems. There are 14 edges and 21 edges for the anti-1-communication-destroying and anti-2-communication-destroying topology, respectively, thus validating Theorem 2. And for each UAV except the global leaders, the indegree of its corresponding vertex is $k+1$. However, the outdegree of these vertices varies, meaning some UAVs in the topology can be local leaders of some other UAVs, while some are just followers. Moreover, it can be seen that our UCFS algorithm does not require the cost matrix to be symmetrical, nor does require satisfying triangle inequality, making it can be easily applied to other area.

The anti-1-communication-destroying and anti-2-communication-destroying topology with both UAV_1 and UAV_2 as global leaders are shown in Fig. 4, demonstrating our algorithm can also be applied to the system with more global leaders. And we have the same claim as given in Theorem 2.

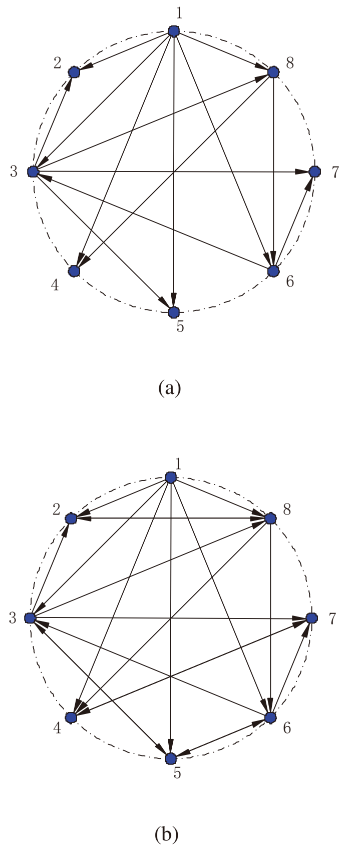


Fig. 3: Anti-1-communication-destroying and anti-2-communication-destroying topology for multi-UAV formation with one global leader.

5 Concluding Remarks

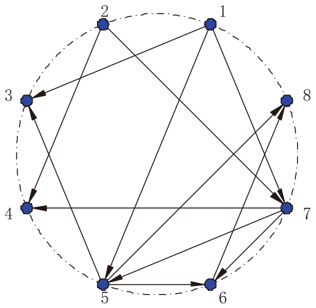
In this paper, we have investigated the anti-k-communication-destroying topology for multi-UAV formation, and put forward the UCFS algorithm to establish the topology, with $k+1$ minimal-cost edge-independent paths found for each UAV. Our algorithm is a general one, as it does not necessarily require the cost matrix to be symmetrical, nor does require satisfying triangle inequality. The algorithm can also be applied to other area involving communication and networks.

There are two potential research directions following this work. Firstly, how to select the m global leaders from n UAVs with the minimal communicating cost instead of simply running the UCFS algorithm repeatedly for $\binom{n}{m}$ times. Secondly, the algorithm to establish anti-k-UAV-destroying topology which bear the loss of any k UAVs still needs to be explored.

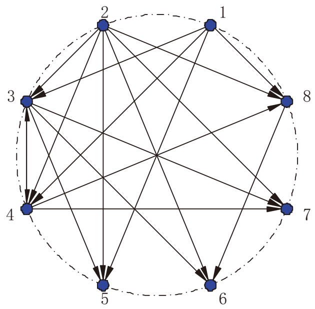
References

[1] F. Giulietti, L. Pollini, and M. Innocenti, "Autonomous formation flight," *IEEE Control Systems*, vol. 20, no. 6, pp. 34–44, 2000.

[2] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive configuration control of multiple uavs," *Control Engineering Practice*, vol. 21, no. 8, pp. 1043–1052, 2013.



(a)



(b)

Fig. 4: Anti-1-communication-destroying and anti-2-communication-destroying topology for multi-UAV formation with two global leader.

[3] V. Roldao, R. Cunha, D. Cabecinhas, C. Silvestre, and P. Oliveira, "A leader-following trajectory generator with application to quadrotor formation flight," *Robotics and Autonomous Systems*, vol. 62, no. 10, pp. 1597–1609, 2014.

[4] S. Zhu, D. Wang, and C. B. Low, "Cooperative control of multiple UAVs for moving source seeking," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 333–346, 2014.

[5] M. Zhang and H. H. T. Liu, "Cooperative tracking a moving target using multiple fixed-wing UAVs," *Journal of Intelligent & Robotic Systems*, vol. 81, no. 3-4, pp. 505–529, 2016.

[6] X. Wang, Z. Zeng, and Y. Cong, "Multi-agent distributed coordination control: Developments and directions via graph viewpoint," *Neurocomputing*, vol. 199, pp. 204–218, 2016.

[7] A. Hellwig and L. Volkmann, "Lower bounds on the vertex-connectivity of digraphs and graphs," *Information Processing Letters*, vol. 99, no. 2, pp. 41–46, 2006.

[8] L. Volkmann, "Local-edge-connectivity in digraphs and oriented graphs," *Discrete Mathematics*, vol. 307, no. 24, pp. 3207–3212, 2007.

[9] I. Stella, L. Volkmann, and S. Winzen, "Strongly 4-path-connectivity in almost regular multipartite tournaments," *Discrete Mathematics*, vol. 307, no. 24, pp. 3213–3219, 2007.

[10] L. Volkmann, "Restricted arc-connectivity of digraphs," *Information Processing Letters*, vol. 103, no. 6, pp. 234–239, 2007.

[11] A. Hellwig and L. Volkmann, "The connectivity of a graph and its complement," *Discrete Applied Mathematics*, vol. 156, no. 17, pp. 3325–3328, 2008.

[12] A. Hellwiga and L. Volkmann, "Maximally edge-connected and vertex-connected graphs and digraphs: A survey," *Discrete Mathematics*, vol. 308, pp. 3265–3296, 2008.

[13] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Transactions on Automatic Control*, vol. 50, pp. 121–127, 2005.

[14] I. Kaminer, A. M. Pascoal, E. Hallberg, and C. Silvestreo, "Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 29–38, 1998.

[15] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[16] E. Xargay, I. Kaminer, A. Pascoal, N. Hovakimyan, V. Dobrokhodov, V. Cichella, A. P. Aguiar, and R. Ghabcheloo, "Time-critical cooperative path following of multiple unmanned aerial vehicles over time-varying networks," *Journal of Guidance Control and Dynamics*, vol. 36, no. 2, pp. 499–516, 2013.

[17] M. Nagy, Z. Akos, D. Biro, and T. Vicsek, "Hierarchical group dynamics in pigeon flocks," *Nature*, vol. 464, no. 8, pp. 890–894, 2010.

[18] H. Whitney, "Congruent graphs and the connectivity of graphs," *American Journal of Mathematics*, vol. 54, pp. 150–168, 1932.

[19] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, Third Edition*. Beijing, China: Tsinghua University Press, 2011.